

Programação em blocos

TCP46 e TCP17

Versão: 1.0

TECNOLOG[>]

www.tecnolog.com.br

 Av. Pernambuco, 2623, | Conj. 101 | Porto Alegre - RS

 Telefone: (51) 3076.7800

 E-mail: vendas@tecnolog.ind.br

9. Instruções de programação (Blocos)

Os blocos disponíveis nas bibliotecas de funções possuem variações de formato e unidade, as abreviações abaixo facilitam o entendimento estrutural dos blocos.

A maioria das instruções matemáticas operam com dados inteiros ou reais e isso é indicado após o nome do bloco [Ex: MUL_I (dado inteiro) e MUL_F (dado float)].

O mesmo acontece com as instruções lógicas utilizando números inteiros, onde teremos _I, I16 (inteiro de 16 bits) e I32 (inteiro de 32 bits).

Abreviações:

V: verdadeiro (ligado)

F: falso (desligado)

NAN: erro lógico no bloco

Terminal Azul: utilizado para Bool / Bit (V ou F)

Terminal Verde: utilizado para Word (16 ou 32 bits)

Terminal Roxo: utilizado para Float (32 bits)

_B / _BOOL: binário (bool - 0 ou 1)

_I / _I16: números inteiros (16 bits)

_I32: números inteiros (32 bits)

_F: números reais (32 bits float)

_MS: milissegundos

_DEC: décimo de segundos

_S: segundos

EN: entrada

IN, IN1 ... INn: terminal ou bloco com sinal ou valor de entrada

CFG: bloco retentivo (não perde o valor quando o CLP é desligado)

OUT, OUT1...OUTn: terminal ou bloco com sinal ou valor de saída

RST: terminal de entrada para retornar as condições iniciais do bloco

REF: entrada do valor de referência

VALID: estado de validade do bloco, sendo V ou F

Cor do bloco (grupo): Vermelho (**IO**), Branco (**lógica**), Laranja (**matemática**), Amarelo claro (**numéricos**), Verde (**teste**), Cinza claro (**temporizadores**), Azul (**controle**), Amarelo escuro (**conversão**) e Cinza escuro (**sistema**)

Exemplos:

Blocos TON_MS e TON_S: **TON_MS** é um temporizador com contagem em milissegundos e **TON_S** temporiza em segundos.

Blocos Modbus escravo MDBS_IN/MDBS_CFG/MDBS_OUT: bloco com entrada não retentiva/entrada retentiva e bloco com saída não retentiva.

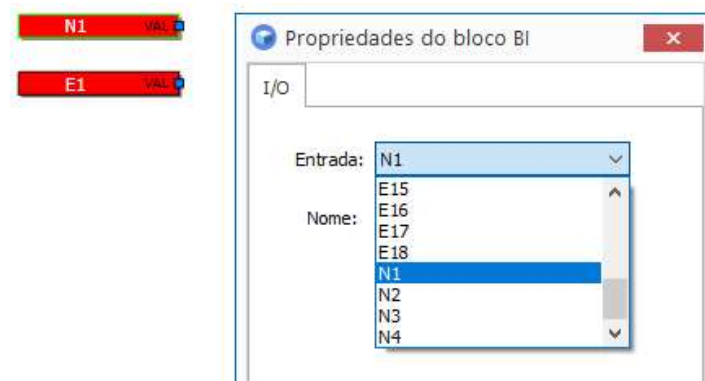
9.1. GRUPO IO – ENTRADAS E SAÍDAS DIGITAIS E ANALÓGICAS

Bloco BI – Entrada digital/Nx (contato seco)

Este bloco permite acesso a uma entrada digital do controlador. O estado da entrada é representado em uma variável booleana para a lógica, sendo **V = acionado** e **F = desacionado**.

O bloco BI também é usado para as entradas do tipo **contato seco** nas entradas N1~N4.

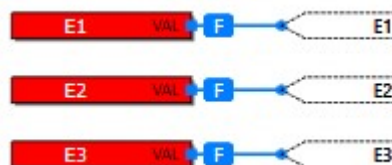
Entrada: ponto a ser monitorado / VAL: status Booleano referente a entrada (V/F).



O firmware amostra as entradas a cada 8,3ms sem sincronismo com a execução do programa.

No programa, o valor é copiado para a memória da lógica sempre que um bloco AI/BI for executado.

Obs: caso precise utilizar um bloco BI lendo a mesma entrada, o seu valor poderá ser diferente dentro de um mesmo ciclo da lógica, pois a execução do segundo bloco irá atualizar novamente o valor. Para isso, recomenda-se o uso de apenas um bloco AI/BI para cada entrada acionando um **label**, o estado do Label será compartilhado nos outros diagramas e será mantido constante durante todo o ciclo.



Outra vantagem da utilização dessa técnica é evitar atraso desnecessários, pois quando o diagrama identifica um bloco **BI**, a execução do programa é interrompida por **28us** para fazer a leitura física desta entrada.

BO – Saída digital

Permite acesso a uma saída digital do controlador.

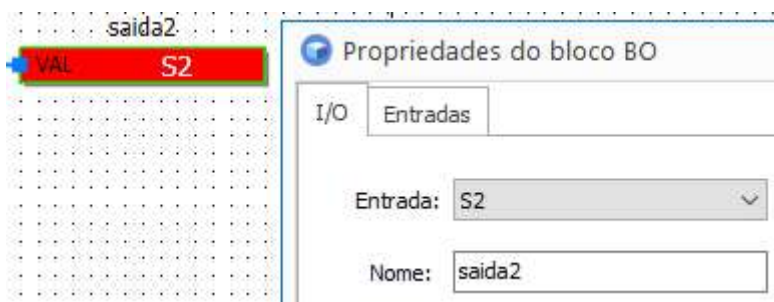
O valor das saídas é atualizado na memória do firmware (lógica -> firmware) sempre que um bloco BO/AO for executado. Esta atualização dispara uma tarefa no firmware que fará a atualização das saídas físicas. Não há sincronismo entre a execução desta tarefa e a lógica em si, mas o tempo é da ordem de poucos milissegundos.

Importante: nunca insira uma BO mais de uma vez, utilize labels com nome diferente (Ex: motor e motor1) e use o bloco OR para acionar a saída.



Entrada: saída física a ser monitorada.

VAL: valor a ser escrito na saída digital. Se **V = verdadeiro = saída acionada** e se **F = falso = saída não acionada**.



AI – Entrada analógica

Permite acesso a uma entrada analógica do controlador.

Estão disponíveis 4 canais AIx (AI1 ~ AI4) e 4 canais para NTC Nx (N1 ~ N4).

A faixa de valores da saída depende do tipo de configuração da entrada. Para entrada de tensão (0/2~10Vdc) ou corrente (0/4~20mA) o valor de saída varia de 0 ~100% e para entradas de sensores NTC (TCP46A), o valor de saída é a temperatura (°C).

Configurações individuais dos canais AI: Projeto > Configurações > Entradas analógicas.

VAL: valor de entrada (0 ~100% para V/mA ou °C para NTC).

VALID: se VALID = **V**, há um valor na entrada.



AO – Saída analógica

Permite acesso a uma saída analógica. Estão disponíveis 4 canais AOx (AO1 ~ AO4).

O valor de uma saída analógica (0/2~10Vdc) ou (0/4~20mA) varia de 0 ~100% referente a grandeza e escala selecionada.

VAL: valor da saída analógica (0 ~100%), pode-se utilizar um bloco ou um valor fixo. Para definir um valor constante, acesse as propriedades do bloco (Entradas > marcar **VAL** > definir o valor).

Para acessar as configurações individuais dos canais pelo TProg, acesse: Projeto > Configurações > Saída analógicas. Também é possível via WebServer.

Ex: VAL = 54%, configurada de 0 ~ 20mA, o valor da saída = 10,8mA (regra de três).



PULSE_FREQ – Entrada de contagem rápida

Permite acesso às 4 entradas contadoras de pulsos rápidos.

COUNT: informa o número de pulsos ocorridos.

RST: zera o número de pulso de contagem.

FREQ: frequência atual de pulsos, atualizada a cada segundo (Hz).

Entradas rápidas: DI1 (E1) e DI4 (E4) até 6,8kHz



Entradas rápidas: DI2 (E2) e DI3 (E3) até 400Hz



PULSE INPUT – Contador de pulso rápido

Permite acesso às entradas contadoras de pulso rápido.

VAL: número atual de pulsos contados.

RST: zera o número de pulso de contagem.



EXP – Blocos para as expansões

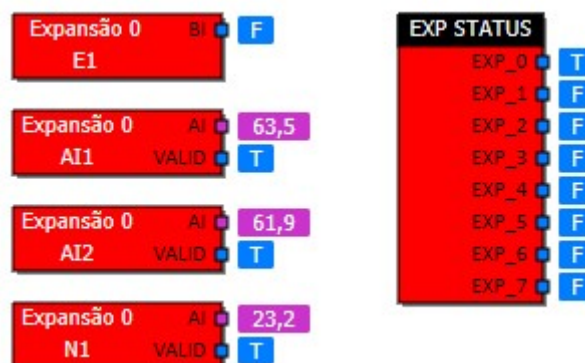
Utilize os blocos de função **EXP** na biblioteca IO, selecione o número da expansão e o ponto a ser monitorado.

EXP_BI (entrada digital/NTC (contato seco)) e EXP_BO (saída digital)

EXP_AI (entrada analógica/NTC) e EXP_AO (saída analógica)

EXP_STATUS (validade em cada expansão)

Obs: não é possível usar as entradas rápidas nas expansões



9.2. MODBUS ESCRAVO - OPERANDO DE COMUNICAÇÃO

Esse grupo permite que um dispositivo mestre da rede leia ou escreva dados do tipo: Bool, Float, I16 e I32 na memória do TCP46.

O grupo MDBS é dividido em 3 blocos:

- MDBS_IN** = leitura **não retentiva**
- MDBS_CFG** = leitura **retentiva**
- MDBS_OUT** = escrita

MDBS_CFG_BOOL / MDBS_CFG_FLOAT / MDBS_CFG_I16 / MDBS_CFG_I32

Disponibiliza uma variável de entrada de um bloco para ser escrita pelo mestre da rede Modbus. Os blocos CFG são **retentivos** e na falta de energia elétrica mantém o último valor informado.



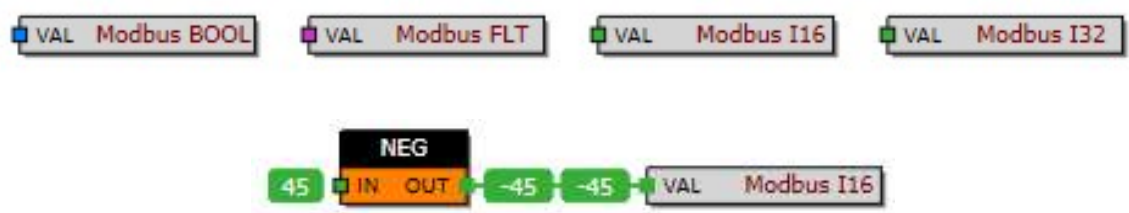
MDBS_IN_BOOL / MDBS_IN_FLOAT / MDBS_IN_I16 / MDBS_IN_I32

Disponibiliza uma variável de entrada de um bloco para ser escrita em um mestre da rede Modbus. Os blocos IN **não são retentivos** e na falta de energia elétrica retorna com o valor 0.



MDBS_OUT_BOOL / MDBS_OUT_FLOAT / MDBS_OUT_I16 / MDBS_OUT_I32

Disponibiliza uma variável de saída de um bloco para ser lida pelo mestre Modbus.

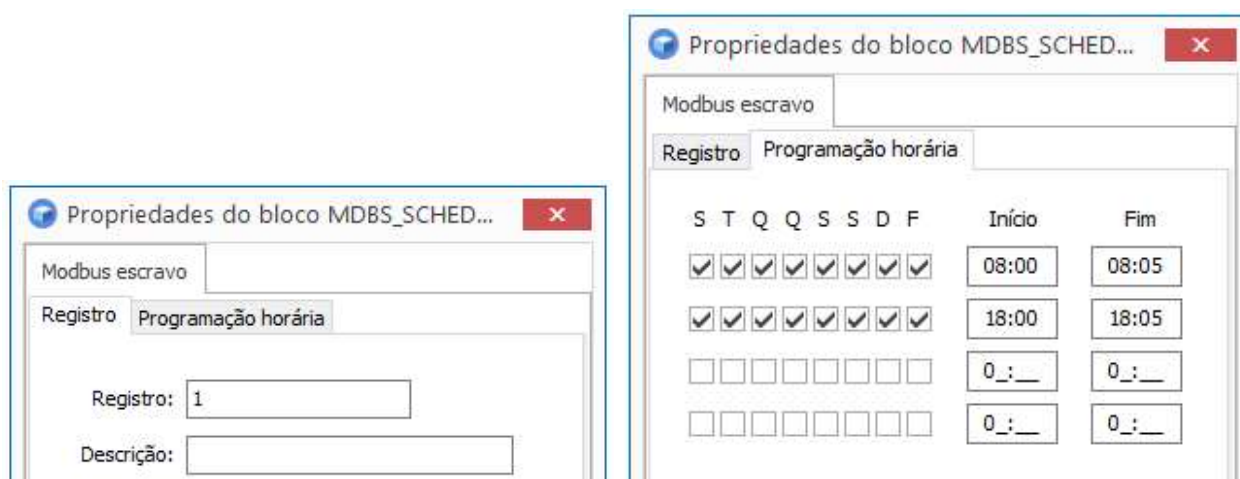


Obs: os blocos MDBS estão mais detalhados no capítulo 8.16.2.

MDBS SCHEDULE – Programação horária

Essa função permite utilizar uma **agenda inclusa** no bloco ou uma **agenda a partir da rede Modbus** para ativar e desativar uma saída em até 4 horários para cada dia da semana.

Esta programação é composta de 4 períodos semanais, com horário de início e fim de cada período. Devem ser configurados os dias da semana para qual o período é válido, indicando o horário de início (liga) e de fim (desliga). A configuração "F" indica se o período é válido em feriados (conforme a tabela de feriados em: Biblioteca > Configurações > Diversos).



No Modbus, a configuração da programação horária é feita em 4 períodos que ocupam 12 registros consecutivos, sendo 3 para cada período, com o seguinte formato:

- **REGISTRO 1:** Dias da semana (Bit 7 = segunda, Bit 6 = terça ... bit 0 = feriados)
Em binário: **11111000** (seg ~ sex)
Convertendo para decimal: **248**
- **REGISTRO 2:** Horário de início
- **REGISTRO 3:** Horário de fim

Nos registros de **horário** (registro 2 e 3), a codificação usada é: **(hora x 100) + minuto**.

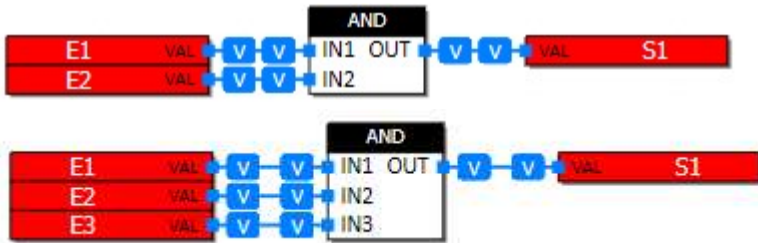
Exemplo: ativar a programação de segunda a sexta, das 08:00 às 18:00 pela IHM.
Registro 1 = **248** (seg ~ sex), registro 2 = **800** (8 h) e Registro 3 = **1800** (18 h).

No registro 1 na IHM o **numeric input** pode ser configurado para **16bit Binary** para que seja possível colocar o valor dos dias em binário (11111000).



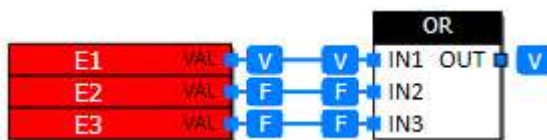
9.3. GRUPO LÓGICA - OPERAÇÕES COM AS ENTRADAS

AND / AND3 – Operação AND (E) de duas ou três entradas



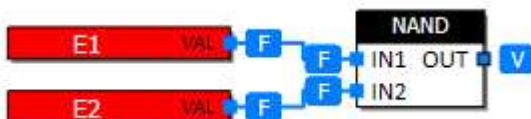
A	B	AND	AND3
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	0
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

OR / OR3 - Operação OR (ou) de duas ou três entradas Se uma ou mais entradas estiver acionada a saída OUT = V.



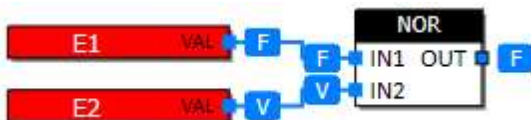
IN3	IN2	IN1	OUT
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

NAND – Operação NAND (E negado) de duas entradas



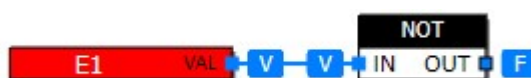
IN 2	IN 1	NAND
0	0	1
0	1	1
1	0	1
1	1	0

NOR – Operação NOR (ou negado) em duas entradas



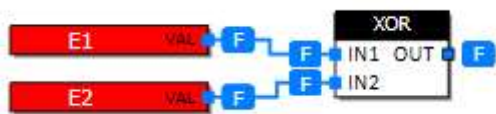
IN 2	IN 1	NOR
0	0	1
0	1	0
1	0	0
1	1	0

NOT – Operação NOT (negado)



IN 1	NOT
0	1
1	0

XOR / XOR3 - Operação lógica (OU exclusivo) de duas ou três entradas

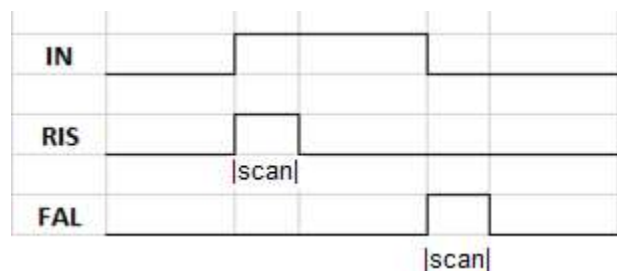


IN 2	IN 1	XOR
0	0	0
0	1	1
1	0	1
1	1	0

EDGE – Detecta a borda de subida/descida do sinal de entrada e copia para a saída

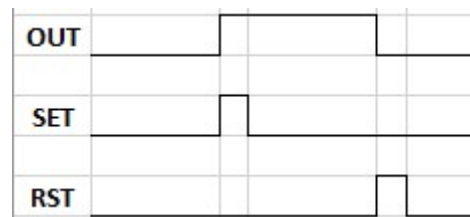
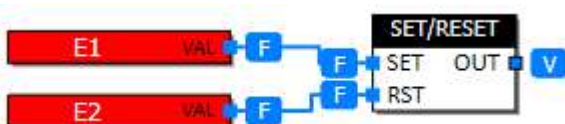
A borda de subida em IN aciona a saída RIS e a borda de descida em IN aciona a saída FAL.

Obs: a saída é ativada durante apenas 1 ciclo do Scan.



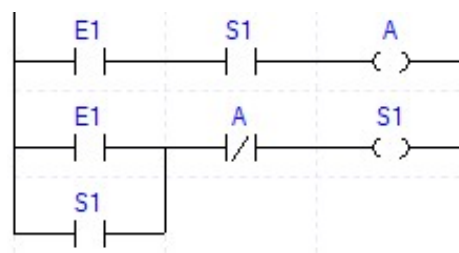
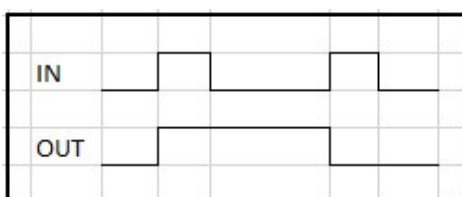
SET RESET – Controla o estado da saída por um pulso na entrada SET ou RST

Conhecido como FLIP-FLOP, é um bloco que memoriza um estado de um bit. A saída permanece verdadeira com um pulso em SET e falsa com um pulso em RST. A entrada **RST tem prioridade** sobre a entrada SET.



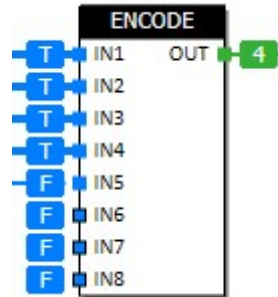
TOGGLE - Inverte o status da saída OUT a cada borda de subida em IN

Inicialmente OUT = F, com um pulso em IN (OUT = V) e um novo pulso em IN (OUT = F).



ENCODE – Identifica a entrada ativa mais significativa

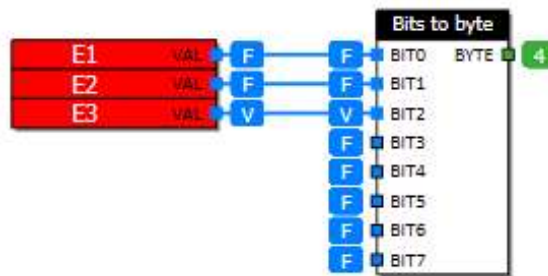
Esse bloco verifica a entrada ativa (1~8) mais significativa, sendo IN1 (menos) e IN8 (mais) e envia o número dessa entrada para OUT (INx mais significativa = OUT). Caso nenhuma entrada esteja ativa, OUT = 0.



Clique [aqui](#) para visualizar esse bloco em uma aplicação prática.

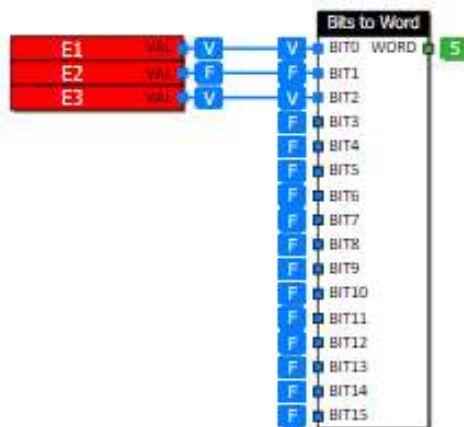
BITS TO BYTE – Conversor de Bits para Byte

Monta uma palavra a partir do nível lógico dos 8 bits individuais das entradas (Bit0 ~ Bit7) e gera um valor decimal correspondente na saída BYTE (0 ~ 256). Esse bloco costuma ser utilizado quando precisamos realizar uma **chave seletora**.



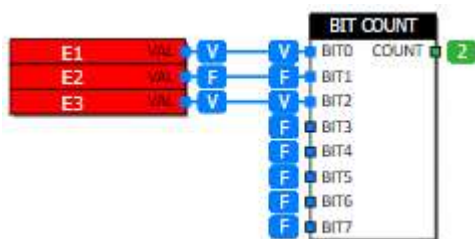
BITS TO WORD – Conversor de Bits para Word

Monta uma palavra a partir do nível lógico dos 16 bits individuais das entradas (BIT0 ~ BIT15) e envia um valor decimal correspondente na saída WORD (0 ~ 65536).



BIT COUNT – Contabiliza até 8 entradas ativas

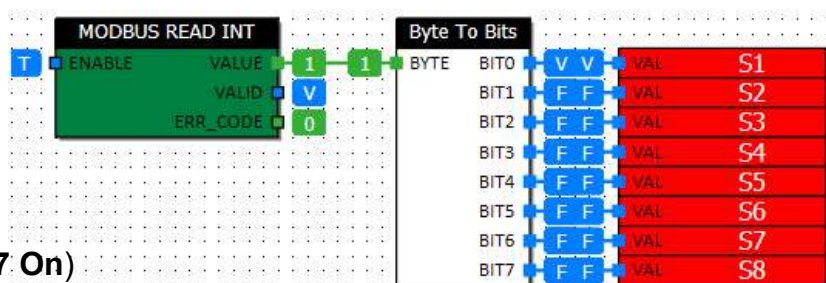
Verifica até 8 entradas ativas (BIT0 ~ BIT7) e gera o somatório em COUNT.



BYTE TO BITS – Conversor de Byte para Bits

Separa e ativa até 8 bits individuais a partir de uma palavra de entrada.

- Valor 1: 0000 0001 (Bit 0 On)
- Valor 2: 0000 0010 (Bit 1 On)
- Valor 4: 0000 0100 (Bit 2 On)
- Valor 8: 0000 1000 (Bit 3 On)
- Valor 16: 0001 0000 (Bit 4 On)
- Valor 32: 0010 0000 (Bit 5 On)
- Valor 64: 0100 0000 (Bit 6 On)
- Valor 75: 0100 1011 (Bits 1, 2, 4 e 7 On)
- Valor 128: 1000 0000 (Bit 7 On)

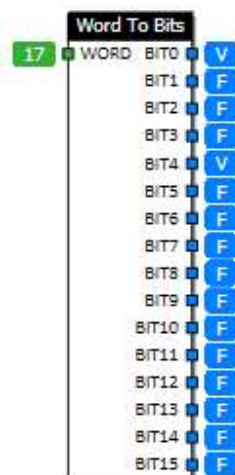


WORD TO BITS – Conversor de Word para Bit

Separa e aciona os bits individuais da palavra de entrada (0 ~ 65536) com até 16 bits.

Ex: entrada WORD = 17

Soma	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	0	0	0	1	0	0	0	1
	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
17	0	0	0	16	0	0	0	1



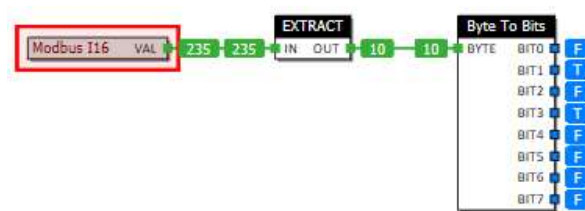
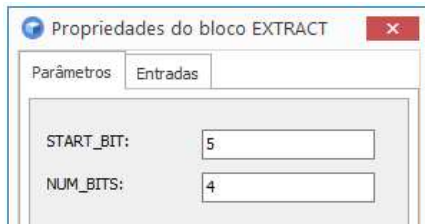
EXTRACT – Extrai bits de uma palavra

Extrai uma parte especificada de uma palavra de 16 ou 32 bits.

O parâmetro START_BIT define o bit inicial (0 a 31) e NUM_BITS define o número de bits a serem extraídos na direção do **mais significativo** para o **menos significativo** (esquerda para direita).

O resultado é a **soma binária** dos bits da palavra resultante (número de bits).

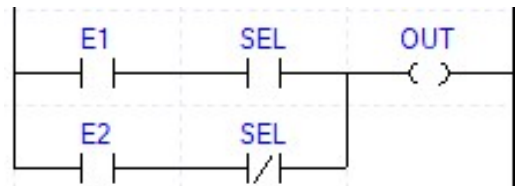
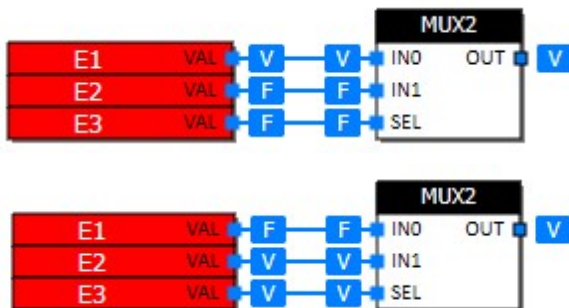
Ex: 235 = 11101011 (extrai 4 Bits a partir do Bit 5) = 10



MUX2 B - Multiplexador binário de 2 entradas

SEL = F, OUT recebe a cópia da entrada IN0.

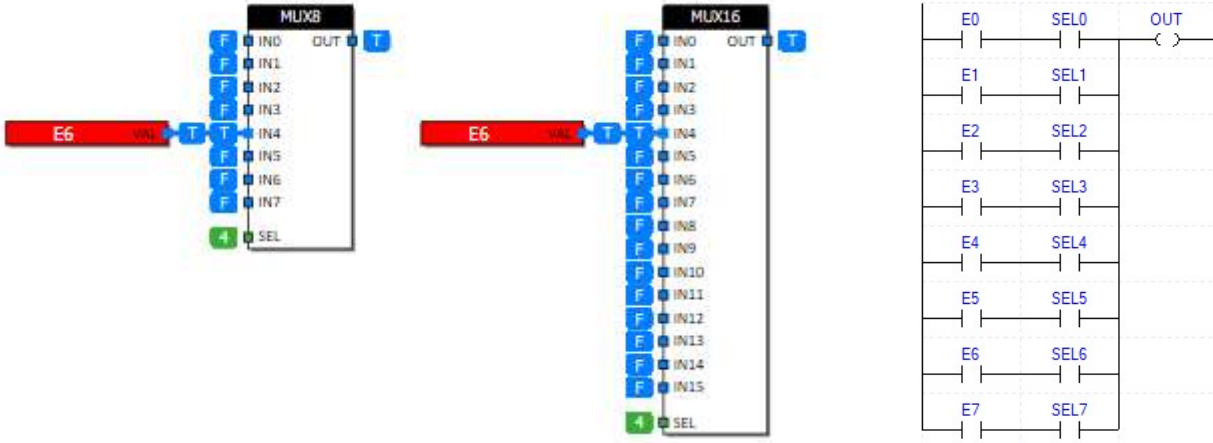
SEL = V, OUT recebe a cópia da entrada IN1.



MUX8 B/MUL16 B - Multiplexador binário de 8 ou 16 entradas digitais

SEL: entrada **Int** que seleciona a entrada **bool** que será copiada para a saída.
 Caso o valor de SEL esteja fora da faixa (**MUX8: 0 ~ 7**) ou (**MUX16: 0 ~ 15**), a saída (OUT) não é alterada.

Ex: SEL = 4, copia a entrada IN4 para OUT.



DEMUX2 – Demultiplexador binário de 2 saídas

Conecta a entrada IN as saídas **OUT0** ou **OUT1** conforme a entrada de seleção **SEL**.

SEL: variável de seleção que define qual informação de entrada será ligada à saída.
 SEL = **F** (OUT0 = IN e OUT1 = F) ou SEL = **V** (OUT1 = IN e OUT0 = F).



SEL	IN	.Out0	.Out1
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

9.4. GRUPO TESTE – COMPARAÇÕES

Os blocos com a terminação **_F** comparam números **reais** e os **_I** comparam números **inteiros**.

EQUAL F / EQUAL I - Verifica se o valor de entrada é igual ao valor de referência
Se $IN = REF$ (OUT = V).



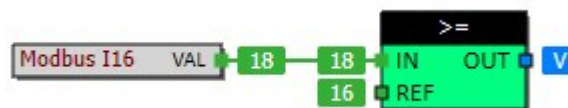
NOTEQUAL F / NOTEQUAL I – Verifica se o valor de entrada é diferente do valor de referência.

Se $IN \neq REF$ (OUT = V).



GREATEREQUAL F / GREATEREQUAL I - Verifica se o valor de entrada é maior ou igual ao valor de referência

Se $IN \geq REF$ (OUT = V)



GREATER F / GREATER I - Verifica se o valor de entrada é maior que o valor de referência

Se $IN > REF$ (OUT = V)



LOWEREQUAL F / LOWEREQUAL I – Verifica se o valor de entrada é menor ou igual ao valor de referência

Se $IN \leq REF$ (OUT = V).

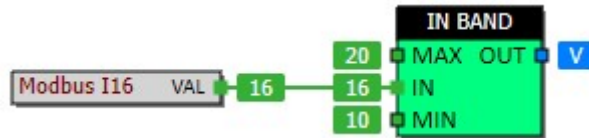


LOWER F / LOWER I – Verifica se o valor de entrada é menor que o valor de referência
Se $IN < REF$ ($OUT = V$).



IN BAND F / IN BAND I – Verifica se o valor de entrada (IN) está entre o valor mínimo (MIN) e máximo (MAX)

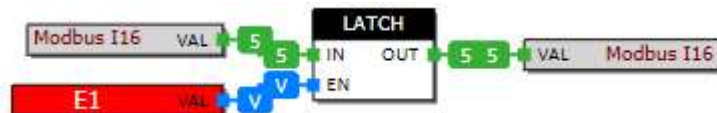
Se $IN \geq MIN$ e $IN \leq MAX$ ($OUT = V$)



9.5. GRUPO MATEMÁTICAS

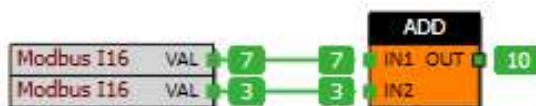
LATCH F / LATCH I – Movimenta valores entre registros

Copia o valor de IN para OUT se EN = V (Função Move).
Se EN = F, o último valor de OUT é mantido.



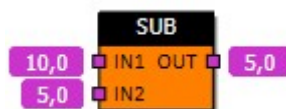
ADD2 F / ADD2 I / ADD3 F / ADD3 I / ADD4 F / ADD4 I – Soma os valores na entrada

Soma os valores de 2, 3 ou 4 entradas.
Obs: **todas** as entradas INx devem ser utilizadas.

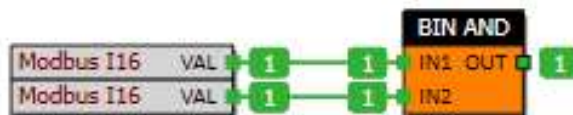


SUB2 F / SUB2 I – Subtrai dois valores de entrada

OUT = IN1 – IN2



AND2 - Operação binária AND (E) em duas entradas

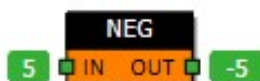


MUL2 F / MUL2 I – Multiplica dois valores de entrada

O bloco MUL2_I permite valores de entrada de -32767 a 32767 (I16 bits) e o resultado em OUT é I32 bits.



NEG F / NEG I - Inverte o sinal do valor de entrada



SIN - Calcula o seno do valor da entrada (em radianos)

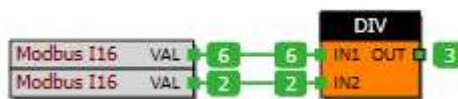
OUT = SIN (IN)

Para que o valor de entrada seja em graus, utilize o bloco “Deg to Rad” antes do SIN.



DIV2 F / DIV2 I – Divide os valores de duas entradas

OUT = IN1 / IN2, sendo IN2 diferente de 0.



FRAC - Mostra a parte fracionada do valor de entrada



INTEG – Mostra a parte inteira do valor de entrada

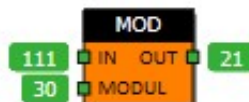


MOD I - Calcula o resto da divisão do número inteiro IN por MOD

Representa a parte indivisível da divisão do número inteiro IN por MODUL.

OUT = IN MOD MODUL

Ex: 111/30 = 21 (resto)



COS - Calcula o cosseno do ângulo de entrada (em radianos)

OUT = COS (IN)

Para que o valor de entrada seja em graus, utilize o bloco “Deg to Rad” antes do COS.



TAN – Calcula a tangente do ângulo do valor da entrada (em radianos)

$$OUT = TAN (IN)$$

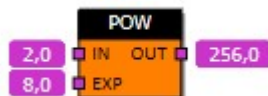
Para que o valor de entrada seja em graus, utilize o bloco “Deg to Rad” antes do TAN.



POW F - Calcula IN elevado a potência EXP

$$OUT = IN ^ EXP$$

$$OUT = 2,0 ^ 8,0 = 256,0$$



ABS F / ABS I – Calcula o valor absoluto da entrada

O valor absoluto (módulo) de um número real é igual a este número se for positivo, e esse número com o sinal oposto caso seja negativo.



SQRT F – Calcula a raiz quadrada do valor de entrada



OR2 - Operação (OR) entre duas palavras de entradas

O resultado é uma outra palavra.



XOR2 - Operação XOR binária entre 2 números inteiros



IN1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	1	1	1	1	1	1	1	1
255	128	64	32	16	8	4	2	1
XOR								
IN2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	1	1	1	0	1	1	1	1
239	128	64	32	0	8	4	2	1
=								
OUT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	0	0	0	1	0	0	0	0
16	0	0	0	16	0	0	0	0

9.6. GRUPO NUMÉRICO

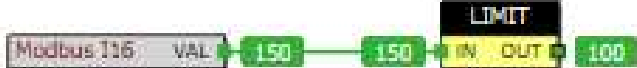
LIMIT F / LIMIT I - Limita o valor de saída

Os valores dos parâmetros MIN e MAX (propriedades do bloco) limitam o valor da saída caso seja inserido um valor em IN fora dos limites especificados.

IN: valor de entrada
OUT: valor de saída limitado.

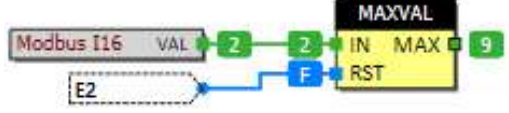
Exemplos:

MIN = 0 e MAX = 100, e IN = 150, OUT = 100 pois o valor de entrada é maior que o máximo permitido.
MIN = 10, MAX = 100 e IN = 5, OUT = 10.



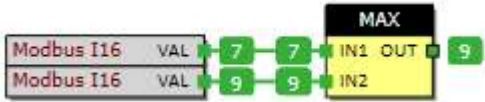
MAXVAL F / MAXVAL I - Registra o maior valor de entrada

O maior valor atribuído a entrada IN é copiado para a saída MAX.
O valor em MAX é resetado quando RST for acionado ou na falta de energia elétrica.



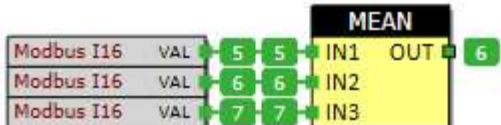
MAX F / MAX I - Move o maior valor atual entre 2 entradas

Copia para OUT o maior valor **presente** nas entradas IN1 ou IN2.



MEAN2 F / MEAN2 I / MEAN3 F / MEAN3 I - Calcula a média aritmética (2 ou 3 IN)

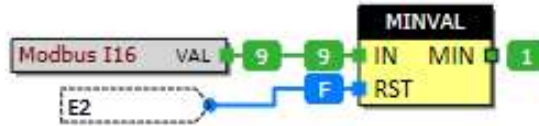
OUT gera o valor médio aritmético das entradas INx.
Obs: O valor de uma entrada falsa (F) não é incluso na média.



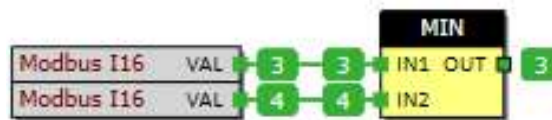
MINVAL F / MINVAL I - Registra o menor valor da entrada IN

O menor valor atribuído a entrada IN é copiado para a saída MIN. O bloco é volátil, perdendo o valor da saída na falta de energia.

Quando RST é acionado, o valor atual de IN é copiado para MIN.



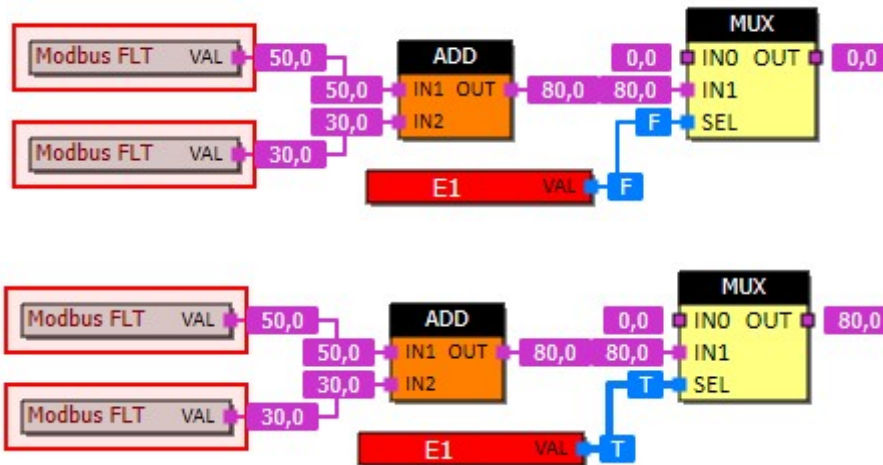
MIN F / MIN I – Copia para OUT a entrada de menor valor atual



MUX2 F / MUX2 I - Multiplexador de 2 entradas

Move para OUT um dos valores das entradas (IN0 e IN1) conforme a chave SEL. Se SEL = F (IN0 = OUT) e se SEL = T (IN1 = OUT).

O bloco **MUX2** permite **condicionar blocos** que não possuem o terminal Enable, assim, o resultado das operações é enviado para a saída OUT somente se SEL=T.

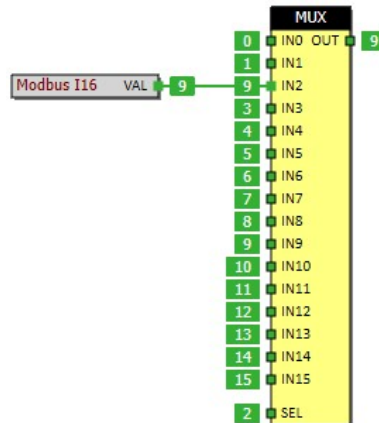


MUX3 F / MUX3 I / MUX4 F / MUX4 I / MUX8 F / MUX8 I / MUX16 F / MUX16 I

Multiplexador de 3, 4, 8 ou 16 entradas (ponteiro)

Os blocos **_F** são usados para dados do tipo *Float* e **_I** usado para dados inteiros.

Copia o valor da entrada apontada por SEL em OUT (OUT = IN_x = Sel_x)

**SCALE2 / SCALE2 I - Aplica uma escala + offset à entrada**

IN: Valor de entrada.

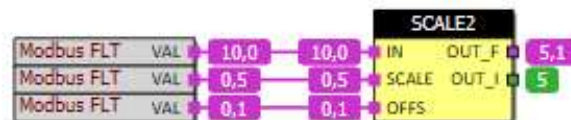
SCALE: Multiplicador a ser aplicado.

OFFS: Offset a ser aplicado.

OUT_F: Saída *Float* com escala aplicada.

OUT_I: Saída *Int* com escala aplicada.

Fórmula do bloco: **OUT = (IN x SCALE) + OFFS** (equação da reta Y=A.X+B)

**SCALE3 F / SCALE3 I / SCALE F / SCALE I**

Aplica uma escala sobre o valor de entrada. Utilizado para escalonar o sinal de sensores.

IN: Valor de entrada sem escala.

IN_MIN: Valor mínimo do sinal de entrada.

IN_MAX: Valor máximo do sinal de entrada.

OUT: Valor de saída com escala aplicada.

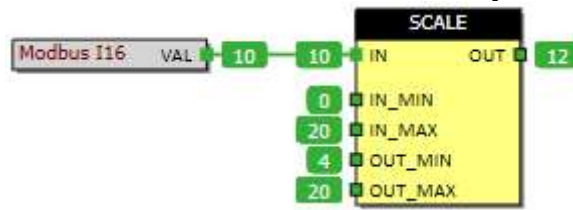
OUT_MIN: Valor de saída para o valor mínimo de entrada.

OUT_MAX: Valor de saída para o valor máximo de entrada.

Fórmula do bloco SCALE3_F:

$$OUT = \frac{IN - IN_MIN}{IN_MAX - IN_MIN} \times (OUT_MAX - OUT_MIN) + OUT_MIN$$

Ex: escala em um sensor com saída de 0 ~ 20mA para mostrar valores de 4 ~ 20 mA. Se a entrada é 0mA, a saída será 4mA e caso a entrada seja 10mA, a saída será 12mA.



FILTER F / FILTER I - Calcula a média do sinal de entrada

Este bloco filtra o sinal de entrada através de uma média aritmética simples a partir da quantidade de amostras e tempo de amostragem.

IN: sinal de entrada a ser filtrado.

OUT: sinal de saída filtrado.

SAMPLE_TIME: tempo de amostragem do sinal de entrada (décimos de segundo).

NUM_SAMPLES: número de amostras médias.

A saída é atualizada conforme: NUM_SAMPLES x SAMPLE_TIME x 100ms.

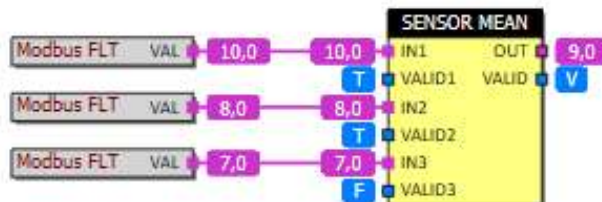


SENSOR MEAN - Calcula a média aritmética em até 3 sensores

Cada entrada possui um sinal de validade, caso a entrada não esteja válida (F) ela é automaticamente removida da média.

VALID: é verdadeiro se ao menos uma entrada INx estiver validada pela entrada VALIDx

OUT: média aritmética dos valores das entradas INx válidas



DELTA HIST - Calcula a diferença entre o valor da entrada e um valor de referência

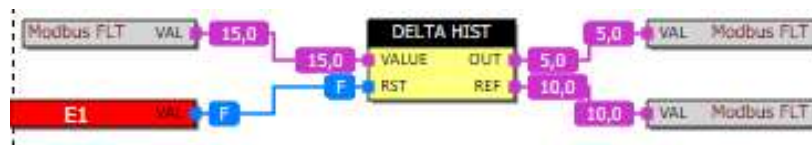
VALUE: valor para comparação (valor atual de um acumulador, por exemplo)

REF: valor atual de referência (retentivo)

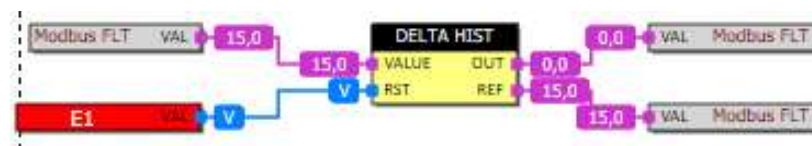
OUT: valor de saída (**OUT = VALUE - REF**)

RST: sinal para reiniciar a comparação. Quando ativo, copia a entrada VALUE para a variável de referência REF

Diferença em relação a um valor antigo: RST não acionado



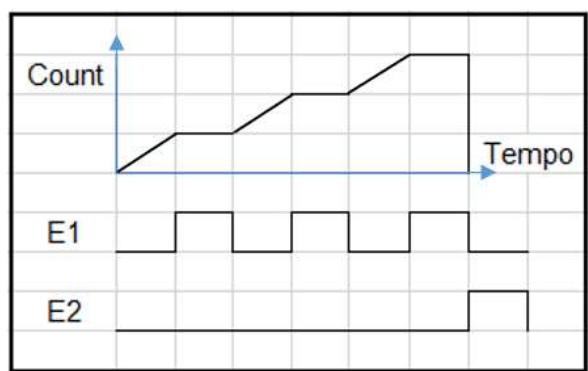
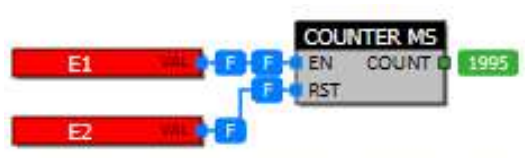
Diferença em relação a um valor atual: RST acionado



9.7. GRUPO TEMPORIZADORES

COUNTER DEC / COUNTER MS / COUNTER SEC - Contabiliza o tempo que a entrada EN permaneceu ativa

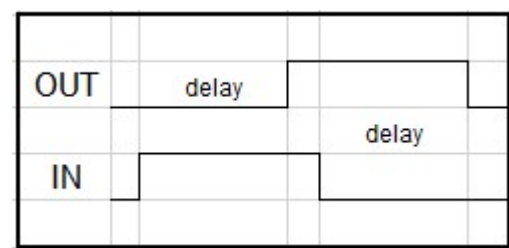
O tempo é acumulativo e armazenado na saída COUNT. Esse valor é zerado por falta de energia ou através de RST.



DELAY / DELAY MIN - Gera um atraso na saída a partir da entrada

ATRASO: define o tempo de atraso (em décimos de segundo ou minutos).

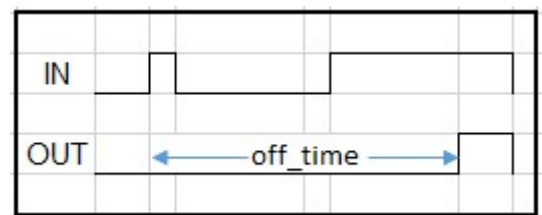
OUT: saída temporizada para ligar e desligar.



MIN OFF: Saída desligada por um tempo

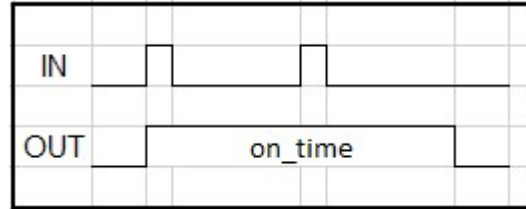
Após um pulso na entrada IN, OUT = F durante o tempo indicado em OFF-TIME (minutos), caso IN = V logo após o tempo OFF_TIME, OUT = V enquanto IN = V.

Aplicação: utilizado em bloqueios por demasiadas tentativas de inserir senhas.



MIN ON: Saída ligada por um tempo

A partir de um pulso na entrada IN, OUT = V durante o tempo definido em ON TIME (minutos), após o tempo decorrido, OUT = F. Pulsos ocorridos durante o período ON_TIME são ignorados.

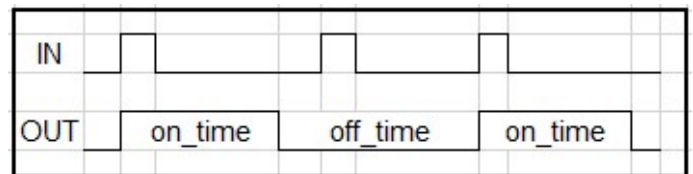


MIN ONOFF - Controla os tempos mínimos para cada estado da saída

Controla OUT para garantir que os tempos mínimos desligado e ligado sejam respeitados. ON_TIME e OFF_TIME controlam os tempos em minutos.

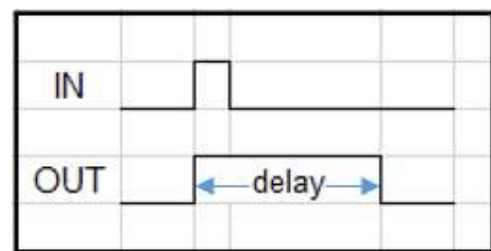
Ex: tempo ON_TIME = 1 e OFF_TIME = 2.

Um pulso em IN, garante que a saída OUT = V (durante 1 minuto). Após o tempo ON_TIME, OUT = F durante OFF_TIME (2 minutos) mesmo com um novo pulso em IN.

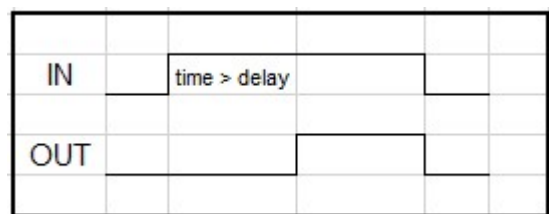


OFF DELAY / OFF DELAY2 - Gera um atraso (décimo de segundo) para desligar a saída

Ex: com a borda de subida de pulso em IN, OUT = V e após o tempo definido em DELAY, OUT = F.



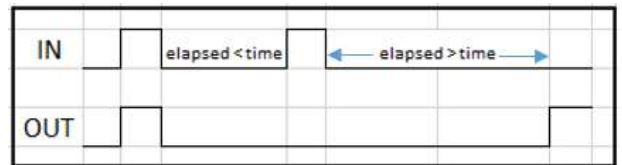
ON DELAY / ON DELAY2 - OUT liga após IN permanecer acionado durante o tempo definido em DELAY



TOFF MS / TOFF S –Temporizador para desligar

Uma borda de descida em IN, OUT = F e o temporizador conta até o tempo “TIME” (milissegundos ou segundos). Após o valor da temporização, OUT = V até receber um novo pulso em IN.
Um novo acionamento em IN reinicia o tempo acumulado.

ELAPSED: tempo decorrido após a borda de descida em IN.



TON MS / TON S - Temporizador para ligar

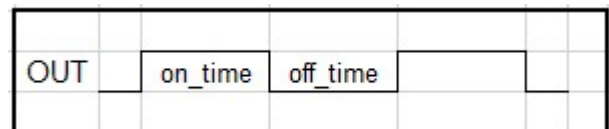
Inicia a temporização se IN = V e após o tempo TIME, OUT = V.
Caso IN seja desativada, o tempo é reiniciado.
TIME: milissegundos ou segundos.
ELAPSED: tempo decorrido.



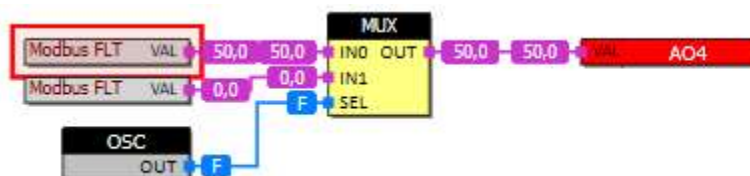
OSCILLATOR - Oscilador com tempo liga/desliga configurável (décimos de segundo)

Este bloco gera um sinal de saída controlado pelos tempos ON_TIME e OFF_TIME.

Ex: pisca led a cada 10 décimos de segundos (1 segundo). ON_TIME e OFF_TIME = 10.



Utilizando em conjunto com o bloco um MUX2 é possível oscilar dois valores com tempos on e off definidos no OSC.



PULSE – Saída com pulso de largura fixa

Gera um pulso na saída de **largura fixa** a cada borda de subida em IN.

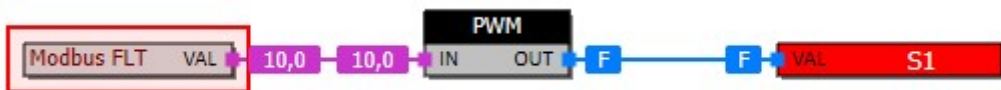
A largura do pulso é definida em PULSE_LENGTH (décimos de segundo).
Caso ocorra uma nova borda de subida em IN durante o pulso de saída, o tempo é reiniciado e o pulso é estendido.



PWM – Saída digital PWM

O bloco PWM modula uma **saída digital** por largura de pulso, conforme o valor de Duty cycle na entrada IN (0 a 100%).

PERIOD: parâmetro que define o tempo de ciclo em segundos (1 ~ 3600).

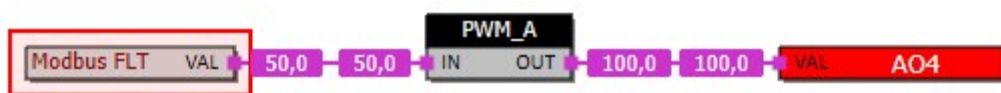


PWM_A – Saída analógica PWM

PWM_A modula uma **saída analógica** por largura de pulso, conforme o valor de Duty cycle na entrada IN (0 a 100%). Normalmente é utilizado em controle de potência com relé de estado sólido como saída de um controlador PID.

PERIOD: parâmetro que define o tempo de ciclo em segundos (1 ~ 3600).

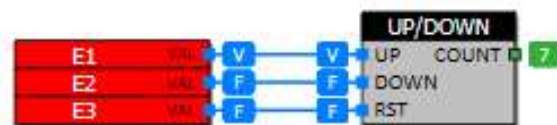
O menor pulso possível é 10ms, que é 1% de 1000ms (1s).



UP_DOWN – Contador Up/Down com reset

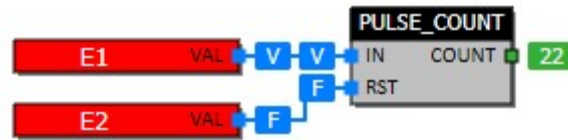
Este bloco conta as transições (bordas de subida) nas entradas UP e DOWN.

A entrada **UP incrementa** o valor de COUNT e **DOWN decrementa** o valor de COUNT. Caso ambas sejam acionadas simultaneamente, a entrada **UP tem prioridade**.



PULSE_COUNT – Contador de pulsos com saída retentiva

COUNT: a cada borda de subida em IN o valor é incrementado.
 Utilize esse bloco para contagens normais (não rápidas).
 Esse bloco possui o valor de saída (count) retentivo.



HOUR_METER1 / HOUR_METER2 – Horímetro

Este bloco implementa um Horímetro retentivo.
 A saída HOURS representa o tempo (em horas) que a entrada ENABLE permaneceu ativa.

HOUR_METER1: valor de **saída Float**.

HOUR_METER2: valor de **saída inteiro**.

Ex: Hours = 0,1 corresponde a 10% de 1 hora, ou seja, 6 min.



9.8. GRUPO CONVERSÕES

TOFLOAT – Converte um número inteiro para real

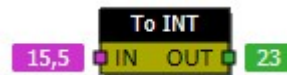


TOINT – Conversor de real para inteiro com escala

É possível aplicar um multiplicador **antes** da conversão do valor. A parte fracionária do número é eliminada, tornando o formato inteiro.

Ex: IN = 15, 5

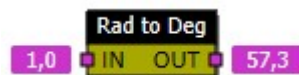
OUT = $15,5 * 1,5$ (escala) = 23,25 > eliminando a parte fracionária OUT = 23.



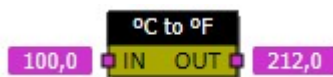
DEG TO RAD – Conversor angular (Graus para Radiano)



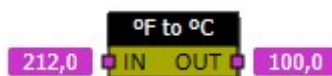
RAD TO DEG – Conversor angular (Radiano para Graus)



C TO F – Conversor de temperatura (Celsius para Fahrenheit)



F TO C – Conversor de temperatura (Fahrenheit para Celsius)

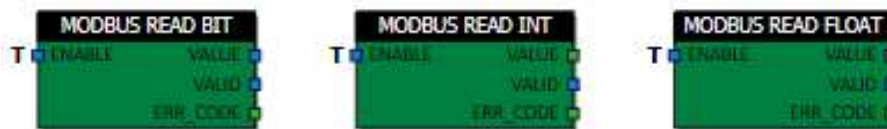


9.9. MDBM – BLOCOS DE COMUNICAÇÃO MODBUS MESTRE

Obs: os blocos MDBM estão melhores detalhados no capítulo 7.18.1.

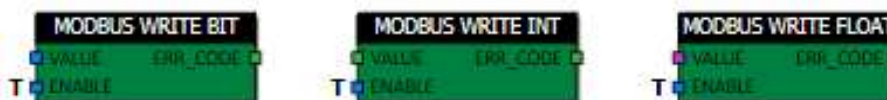
MDBM READ BIT / MDBM READ INT / MDBM READ FLOAT

Faz a leitura de um registro (Bit, Int ou Float) em um equipamento escravo Modbus.



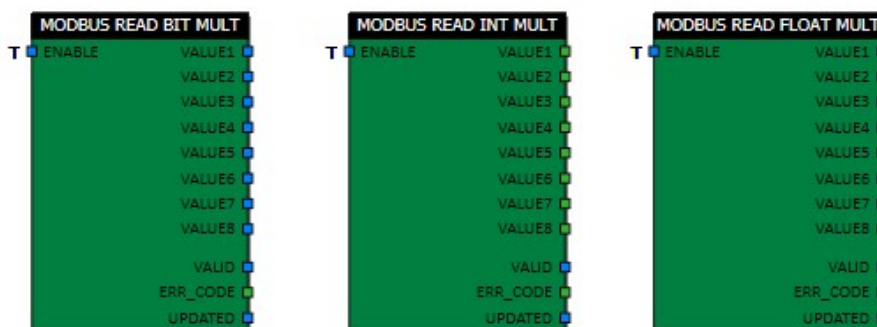
MDBM WRITE BIT / MDBM WRITE INT / MDBM WRITE FLOAT

Faz a escrita de um registro (Bit, Int ou Float) em um equipamento escravo Modbus.



MDBM READ BIT MULTIPLE/ MDBM READ INT MULTIPLE/ MDBM READ FLOAT MULTIPLE

Faz a leitura de até 8 registros em sequência (Bit, Int ou Float) a partir de um endereço inicial em um equipamento escravo na rede Modbus.



MDBM WRITE BIT MULTIPLE/ MDBM WRITE INT MULTIPLE/ MDBM WRITE FLOAT MULTIPLE

Faz a escrita de até 8 registros em sequência (Bit, Int ou Float) a partir de um endereço inicial em um equipamento escravo na rede Modbus.

MODBUS WR BIT MULT			
T	ENABLE	ERR_CODE	<input type="checkbox"/>
		UPDATED	<input type="checkbox"/>
F	VALUE1		
F	VALUE2		
F	VALUE3		
F	VALUE4		
F	VALUE5		
F	VALUE6		
F	VALUE7		
F	VALUE8		

MODBUS WR INT MULT			
T	ENABLE	ERR_CODE	<input type="checkbox"/>
		UPDATED	<input type="checkbox"/>
0	VALUE1		
0	VALUE2		
0	VALUE3		
0	VALUE4		
0	VALUE5		
0	VALUE6		
0	VALUE7		
0	VALUE8		

MODBUS WR FLOAT MULT			
T	ENABLE	ERR_CODE	<input type="checkbox"/>
		UPDATED	<input type="checkbox"/>
0	VALUE1		
0	VALUE2		
0	VALUE3		
0	VALUE4		
0	VALUE5		
0	VALUE6		
0	VALUE7		
0	VALUE8		

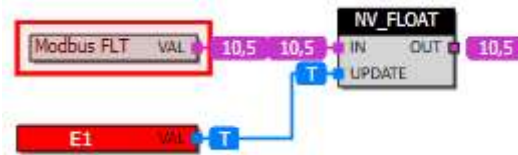
9.10. GRUPO DIVERSOS

NV FLOAT - Cria uma variável *Float* retentiva

Esse bloco cria uma variável *Float* não volátil (retentiva mesmo com o CLP desligado), o valor é armazenado na memória RAM.

O valor de saída é atualizado a cada borda de subida na entrada UPDATE.

Para armazenar um valor inteiro (tipo word), use o bloco TOFLOAT em série com IN.



SCHEDULE OFFSET – Programação horária interna

Essa função permite utilizar uma agenda a partir dos terminais do bloco, para ativar e desativar uma saída em até 4 horários para cada dia da semana.

Esta programação é composta de 4 períodos semanais, com horário de início e fim de cada período. Devem ser configurados os dias da semana para qual o período é válido, indicando o horário de início (liga) e de fim (desliga). Validade do período em feriados (tabela de feriados em: Biblioteca > Configurações > Diversos).

A programação é feita pelos terminais, sendo 3 (Px_DAYS, Px_INI e Px_END) para cada programação de horários.

- **Px_Days:** Dias da semana (Bit 7 = segunda, Bit 6 = terça ... bit 0 = feriados)
Em binário: **11111000** (seg ~ sex)
Convertendo para decimal: **248**
- **Px_INI:** Horário de início
- **Px_END:** Horário de fim

OFFS_INI: offset (minutos) para ligar a saída **antes** da programação configurada.

OFFS_END: offset (minutos) para ligar a saída **após** a programação configurada.

Exemplo: ativar a programação de segunda a sexta, das 15:45 às 15:50.

P1_DAYS = **248** (seg ~ sex), P1_INI = **1545** (15:45h) e P1_END = **1550** (15:50h).



9.11. GRUPO HVAC

ABS HUMITY – Calcula a umidade absoluta do ar (g/kg)

Fórmula do bloco:

$$OUT = \frac{621.9907 \times PV}{ATM_PRESS - PV}$$

PV representa a pressão de vapor, definida por:

$$PV = \frac{PS \times RH}{100}$$

PS é a pressão de saturação, definida por:

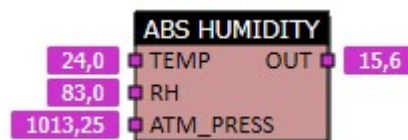
$$PS = 6.116441 \times 10^{\frac{7.591386 \times TEMP}{TEMP + 240.7263}}$$

TEMP: Temperatura, em °C (medida em campo).

RH: Umidade relativa do ar, em %RH (medida em campo).

ATM_PRESS: Pressão atmosférica, em hPa (padrão: 1013,25 hPa ou mbar).

OUT: Umidade absoluta, em g/kg.



ENTHALPY – Cálculo da entalpia do ar

Calcula a entalpia do ar baseado na temperatura e umidade relativa.

A fórmula utilizada para cálculo é:

$$OUT = 4.18 \times \left(6.7 + 0.243 \times TEMP + \left(\frac{RH}{100} \times 10^{\frac{7.5 \times TEMP}{237.3 + TEMP}} \right) \right)$$

TEMP: Temperatura do ar, em °C (medida em campo).

RH: Umidade relativa do ar, em %RH (medida em campo).

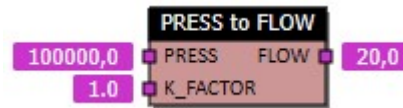
OUT: Entalpia do ar, em kJ/kg.



PRESS TO FLOW – Conversor de Pressão (PA) para vazão (m³/h)

Converte uma entrada de pressão diferencial (em Pa) para uma saída de vazão (m³/h), aplicando o fator K da caixa de VAV.

$$FLOW = K_FACTOR * \sqrt{\frac{PRESS}{248.84}}$$



PRESS: Pressão, em Pa.

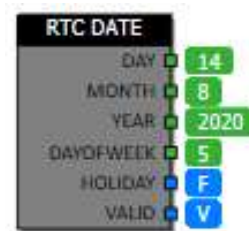
K_FACTOR: Fator K, sem unidade.

FLOW: Vazão, em m³/h.

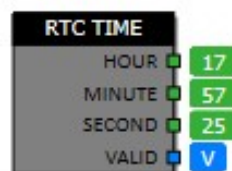
9.12. GRUPO SISTEMA

Obs: a data e hora podem ser alterada pelo WebServer.

RTC DATE – Informa a data atual do TCP46



RTC TIME – Informa a hora atual do TCP46



9.13. GRUPO CONTROLE

STATE MACHINE 4/8/16 – Máquina de estados de Moore/Sequência de operações

Este bloco cria uma máquina de estados (tipo Moore) para sequenciamento de operações, onde somente um estado E_x pode estar ativo de cada vez e a passagem de um estado para outro é feito pelo estado lógico da transição TR_x .

ENABLE: O estado normal da entrada Enable é sempre T (verdadeiro), se for conectado um condicionante, o estado normal da entrada passa a ser F, aguardando a condição ficar T para habilitar o funcionamento do bloco, se Enable = F, a máquina congela no estado atual.

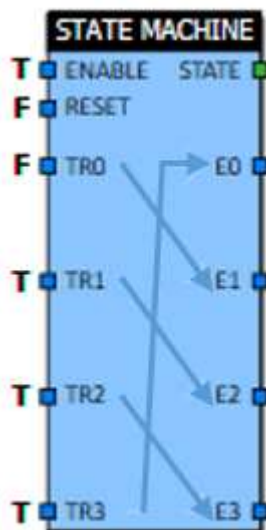
RESET: se ativada, todas as saídas E_x serão desligadas e o estado atual retorna ao $E0$.

STATE: número do estado atual (diagnóstico intrínseco do processo)

TR0 ~ TR7: transições (condição lógica para a troca de estado)

E0 ~ E7: saídas de cada estado que condicionam as ações

RETENTIVO: Parâmetro para habilitar a retenção do Estado atual do bloco (de fábrica o parâmetro vem desabilitado). Habilitar a retenção (propriedades do bloco > retentivo > true) implica em retornar ao último estado de operação após desligar o CLP ou em uma queda de energia elétrica.



- A operação do bloco inicia sempre com o $E0$ ativado.

- Os estados e transições não utilizados são ignorados.

- Após o último estado a máquina retorna automaticamente para o estado 0.

- Cada máquina pode ter até 8 estados, caso sejam necessários mais estados basta ligar outros blocos em série.

- Apenas uma troca de estado ocorre a cada ciclo do CLP.

Mesmo com todas as transições válidas, apenas uma troca de estado ocorrerá a cada ciclo, ou seja, são necessárias no mínimo 7 varreduras para completar um ciclo do bloco.

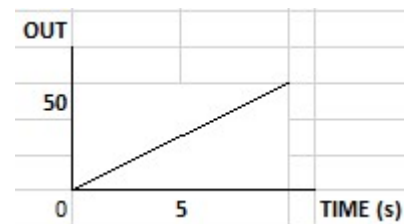
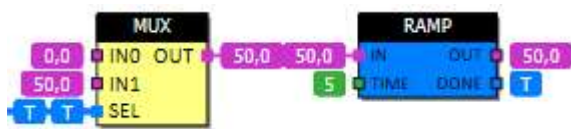
Clique [aqui](#) para ver mais exemplos do uso desse bloco no capítulo 11.

RAMP – Rampa com tempo de transferência

Esse bloco transfere gradualmente o valor da entrada para a saída no tempo definido em TIME, após detectar uma variação no valor da entrada IN.

O parâmetro “TIMEBASE_MS” define o tempo base do bloco em múltiplos de milissegundos.

O bloco percebe uma variação no valor em “IN” e inicia uma rampa na saída para igualar os dois valores, com tempo definido em “TIME”. Quando OUT=IN, a saída “DONE” fica verdadeira, confirmando que a saída chegou ao novo patamar e permanecerá verdadeira até o valor “IN” ser novamente alterado.



Clique [aqui](#) para visualizar um exemplo prático.

ANALOG STAGING4 - Controle o rodízio de até 4 saídas analógicas com sinal individual de habilitação

Cada saída possui um sinal de falha associado (FAULTx) que quando ativado, desativa a saída correspondente e causa um rodízio.

A entrada deve ser linear, variando de 0 a NUM_OUT x 100%. Assim que um estágio atingir 100%, um novo é ligado parcialmente.

Entradas

ENABLE: Habilita o bloco para operação. Quando desabilitado, os estágios são desligados em sequência, respeitando os tempos programados.

IN: Entrada de sinal. Este sinal deve variar de 0 a 100%, multiplicado pelo número de saídas. Para 3 saídas, de 0 a 300%, por exemplo.

FAULTx: Sinal de falha para a saída x. Quando ativado, desliga a saída e força um rodízio.

ROTATE: Quando ativada força um rodízio das saídas.

Saídas

HORAS_OPER: Número de horas em operação do bloco. Quando atinge o tempo programado (ROTATE_HOURS), um rodízio é forçado e este valor é reiniciado em zero. O número de horas de operação é retentivo, ou seja, se mantém mesmo na falta de energia.

DOx: Sinal de habilitação para a saída x. Permite a habilitação do inversor, se necessário.

AOx: Valor analógico da saída x. Varia de 0 a 100%.

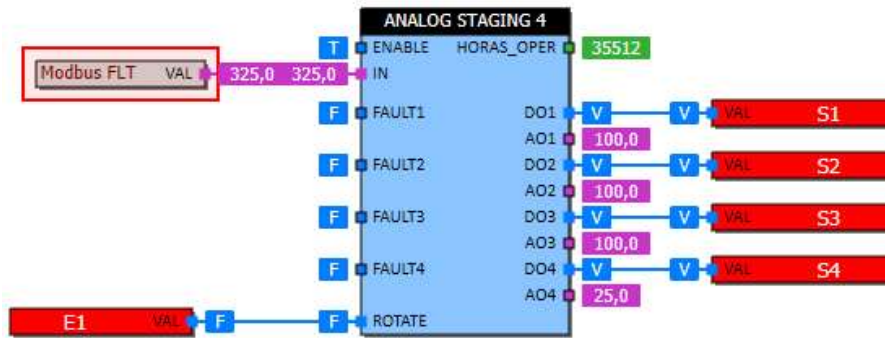
Configurações

STAGE_DELAY: Define o atraso entre estágios (ligar e desligar), em segundos. Variações parciais na saída também respeitam este atraso.

NUM_OUT: Define o número útil de saídas. O bloco usa as saídas em sequência, de 1 ao número útil.

DEADBAND: Banda morta para alteração da saída. Diferenças entre entrada e saída menores que a banda morta são ignoradas.

ROTATE_HOURS: Número de horas para rodízio.



FLOAT VALVE - Controla a posição de uma válvula *Float* utilizando duas saídas digitais

Este bloco permite o controle de uma válvula *Floating* utilizando duas saídas digitais para controlar a abertura e o fechamento de uma válvula.

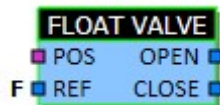
POS: Posição desejada da válvula.

REF: Na borda de subida do sinal, executa o referenciamento da válvula. Este referenciamento é feito automaticamente na inicialização do bloco.

OPEN: Saída digital para controle de abertura da válvula.

CLOSE: Saída digital para controle de fechamento da válvula.

TEMPO_ABERTURA: Parâmetro do tempo total de execução da válvula (segundos).



INTERLOCK - Intertravamento de saídas

Permite que somente uma saída seja acionada por vez.

A entrada habilita a saída correspondente (Ex: IN1 liga OUT1).

Depois que a primeira saída é ligada, nenhuma outra pode ser ligada sem que a primeira saída seja desligada anteriormente.

Obs: No modo LATCH, a saída é desligada somente se a entrada RESET for acionada. Para ativar essa função: propriedades do bloco > alterar LATCH para verdadeiro.

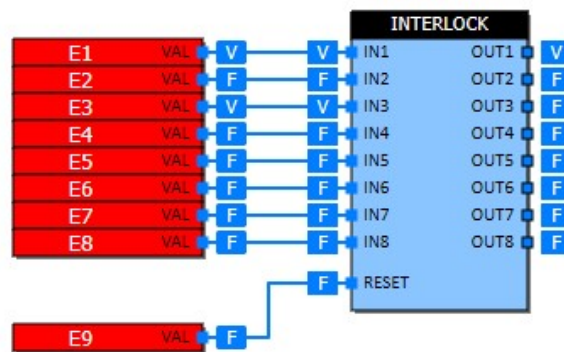
INx: Cada entrada controla o estado da saída correspondente.

RESET: No modo LATCH, o desligamento das saídas é feito pelo acionamento desta entrada.

LATCH: Se verdadeiro, a saída acionada só é desligada pelo acionamento da entrada Reset, independente do estado da entrada correspondente.

OUTx: Comanda das saídas intertravadas.

No exemplo, IN1 e IN3 estão acionadas, porém IN1 foi acionada primeiro, então a saída OUT1 foi acionada. Para que a OUT3 possa ser acionada, a entrada IN1 deve ser desacionada.



ONOFF C - Controle ON/OFF para refrigeração

A **saída ativa** com o valor de entrada **acima** do **setpoint** e **desativada** com o valor da entrada **abaixo** de Setpoint - Histerese.

IN: Variável de controle.

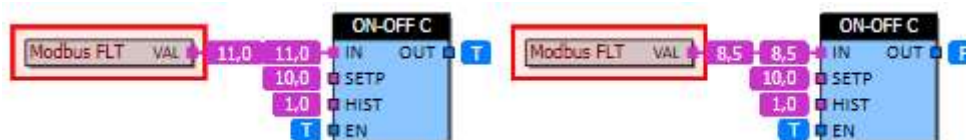
SETP: *Set point* para a variável de controle.

EN: Entrada de habilitação do bloco. Se desativada, a saída permanece desligada.

HIST: Entrada de histerese para desligamento da saída. É desligada quando a entrada IN fica abaixo de SETP - Histerese.

OUT: Saída do controlador.

Ex: A temperatura na entrada é 11°C e está acima do *Set Point* (10°C), a saída permanece verdadeira enquanto a entrada for maior que 9°C (SETP – Histerese = 10° – 1° = 9°).



ONOFF H - Controle ON/OFF para aquecimento

A **saída ativa** com a entrada **abaixo do setpoint** e **desligada** com a entrada **acima de setpoint + histerese**.

IN: Entrada da variável de controle.

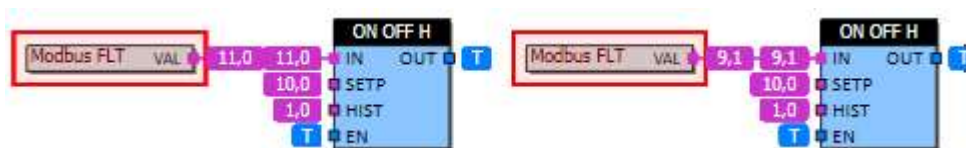
SETP: *Set point* para a variável de controle.

EN: Entrada de habilitação do bloco. Se desativada, a saída permanece desligada.

OUT: Saída do controlador.

HIST: Entrada de histerese para desligamento do controlador. A saída é desativada caso a entrada fique acima de SETP + HISTERESE.

Ex: A temperatura de entrada (9°C) está abaixo do *Set Point* (10°C), a saída permanece verdadeira enquanto a entrada for menor que 11°C (SETP + Histerese = 10° + 1° = 11°).



ONOFF MULT4 - Controle ON/OFF Multiestágio (até 4) para refrigeração e aquecimento

Este bloco implementa um controle ON-OFF Multiestágio, permite o escalonamento de estágios de refrigeração/aquecimento a partir da diferença entre a entrada e o setpoint.

IN: Variável a ser controlada.

SETP: *Setpoint* para a variável controlada.

HISTERESE1: SETP + HISTERESE1 e desativado quando menor que SETP.

HISTERESE2: Ponto de acionamento do segundo estágio. É ativado quando a diferença da entrada for maior que SETP + HISTERESE [1, 2] e desativado quando menor que SETP + HISTERESE 1

HISTERESE3: Ponto de acionamento do terceiro estágio. É ativado quando a diferença da entrada for maior que SETP + HISTERESE [1, 2, 3] e desativado quando menor que SETP + HISTERESE [1, 2]

HISTERESE4: Ponto de acionamento do terceiro estágio. É ativado quando a diferença da entrada for maior que SETP + HISTERESE [1, 2, 3, 4] e desativado quando menor que SETP + HISTERESE [1, 2, 3]

MODE: Define o modo de operação (0 = refrigeração e 1 = aquecimento). No modo refrigeração, a diferença na entrada é calculada como IN - SETP. No modo aquecimento como SETP - IN.

STAGES: Número de estágios que devem ser acionados (saída).

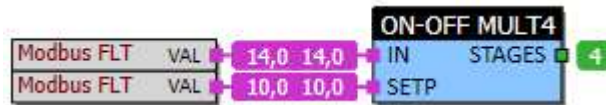
Ex: Com os 4 valores de histerese = 1 e MODE = 0 (refrigeração), os valores de entrada acima de SP temos os seguintes estágios acionados:

Valor de entrada = 10, nenhum estágio acionado;

Valor de entrada = 11, estágio 1 acionado;

Valor de entrada = 12, estágio 2 acionado;

Valor de entrada = 13, estágio 3 acionado;
 Valor de entrada = 14, estágio 4 acionado.



Clique [aqui](#) para ver um exemplo prático.

ONOFF MULT C - Controle ON-OFF Multiestágio para refrigeração

Este bloco implementa um controle ON-OFF Multiestágio, permitindo o escalonamento de estágios de refrigeração a partir da diferença entre a entrada e o setpoint. Este bloco permite o controle de refrigeração até 3 estágios.

Entradas:

IN: Variável a ser controlada.

SETP: Setpoint para a variável controlada.

Saídas:

STAGES: Número de estágios que devem ser acionados.

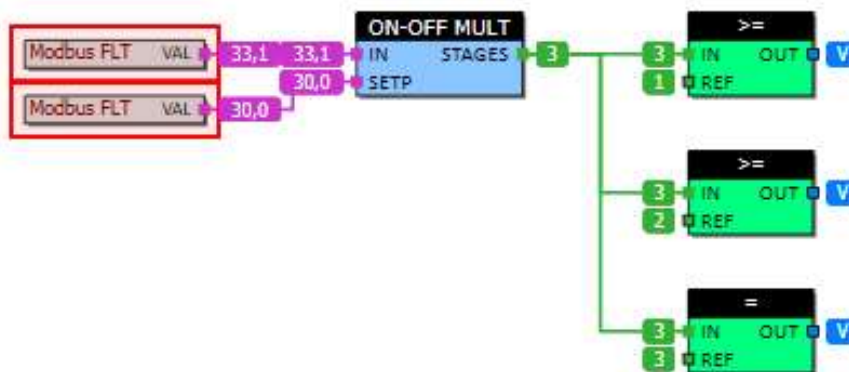
Configurações:

HISTERESE1: Ponto de acionamento do primeiro estágio. É ativado acima de $SETP + HISTERESE1$ e desativado abaixo de SETP.

HISTERESE2: Ponto de acionamento do segundo estágio. É acionado acima de $SETP + HISTERESE1 + HISTERESE2$ e desativado abaixo de $SETP + HISTERESE1$.

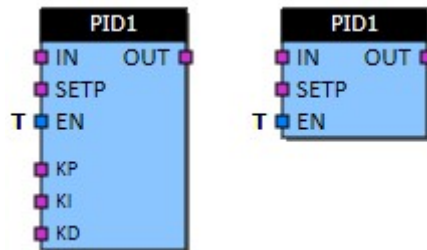
HISTERESE3: Ponto de acionamento do terceiro estágio. É acionado acima de $SETP + HISTERESE1 + HISTERESE2 + HISTERESE3$ e desativado abaixo de $SETP + HISTERESE1 + HISTERESE2$.

Ex: Com o $SETP = 30$, $Histerese1 = Histerese2 = Histerese3 = 1$ e $IN = 33,1$, os 3 estágios serão acionados (3, 1º de diferença) pois a cada 1º de erro faz acionar uma saída.



PID1B_F / PID1_F - Controle PID clássico

Os blocos PID1B_F e PID1_F são funções de controle PID de resfriamento que podem ser utilizados individualmente ou combinado (controle P, controle PI, controle PD ou PID). O bloco PID1B_F possui terminais para ajustar online os ganhos individualmente e no bloco PID1_F os ganhos são especificados na programação das propriedades do bloco.



Neste tipo de controle combinamos a **ação proporcional ao erro de MV (proporcional)** com a capacidade de **eliminação do offset da ação MV (integral)** e ainda adicionamos o **efeito estabilizador da ação antecipativa da MV (derivativa)**, o valor final resultante **MV** será o valor **transferido para o atuador do processo**.

Fórmula do bloco:

$$OUT = KP \cdot \varepsilon + KI \cdot \int \varepsilon + KD \cdot \Delta \varepsilon$$

Entradas:

IN: Entrada da variável de controle (PV).

SETP: Setpoint para a variável de controle (SV)

EN: Habilitação do bloco. Caso desativada, a saída permanece em zero.

Saídas:

OUT: Saída do controle PID limitada entre os valores OUT_MIN e OUT_MAX (MV)

Configurações:

OUT_MAX: Valor máximo da saída do PID.

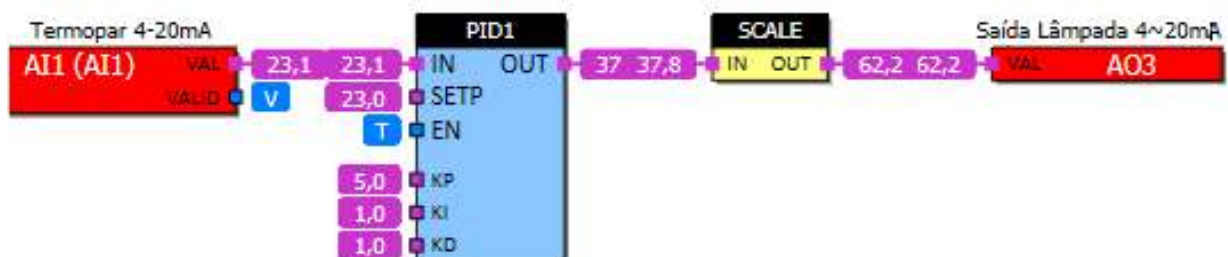
OUT_MIN: Valor mínimo da saída do PID.

EXEC_TIME: Tempo de execução (x 100ms) do PID. Um novo valor para a saída é calculado a cada EXEC_TIME. Esta configuração permite fazer o PID executar mais lentamente em sistemas com resposta lenta.

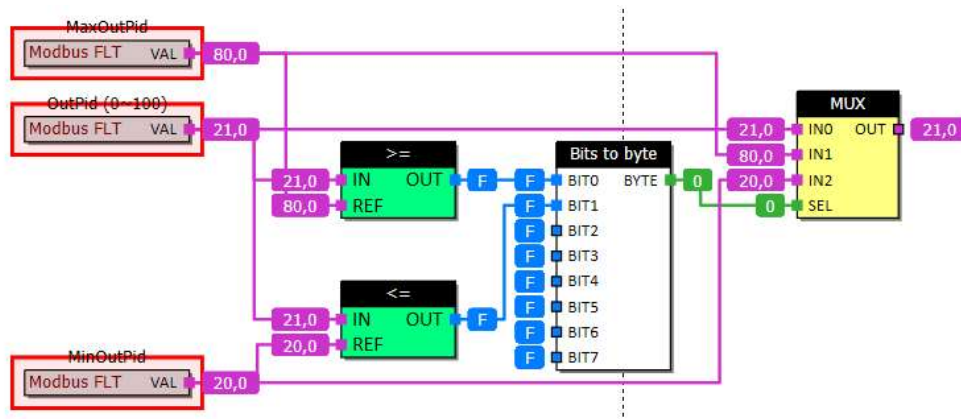
KP: Ganho proporcional do controlador PID.

KI: Ganho integrativo do controlador PID.

KD: Ganho derivativo do controlador PID.



PID com limites de saída ajustável (0~100%) - OutPid = Out do bloco PID1



PID2B F / PID2 F – Controle PI ISA

Os blocos PID2B_F e PID2_F implementam um controle PI no padrão ISA. O bloco PID2B_F possui terminais para alteração nos ganhos KP e KI, enquanto o bloco PID2_F os ganhos são fixos e determinados na programação do bloco.

Entradas:

IN: Entrada da variável de controle.

SETP: Setpoint para a variável de controle.

EN: Habilitação do bloco. Caso desativada, a saída permanece em zero.

Saídas:

OUT: Saída do controle PID. Esta saída é limitada entre os valores OutMin e OutMax.

Configurações:

PB: Banda proporcional (-1000 a 1000) do PID. Este parâmetro define o valor de erro para deixa a saída máxima (OutMax).

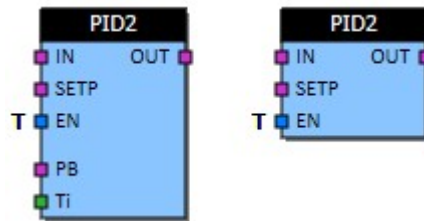
Ti: Tempo de integração (x 100ms). Este parâmetro define o tempo que o controlador repete a parcela proporcional na saída do PID. Tempos maiores tornam a contribuição integral mais lenta.

Deadband: Banda morta do controlador. Valores de erro menores que a banda morta são ignorados e a saída permanece constante. Evita atuações desnecessárias pela variação de saída do controlador.

OutMax: Valor máximo da saída do PID.

OutMin: Valor mínimo da saída do PID.

ExecTime: Tempo de execução (x100ms) do PID. Um novo valor para a saída é calculado a cada ExecTime. Esta configuração permite fazer o PID executar mais lentamente em sistemas com resposta lenta.



PRIORITY LOCK - Intertravamento de saídas (até 32) com prioridade

As entradas INx controlam as saídas OUTx, mas apenas uma saída será acionada de cada vez.

A entrada com maior prioridade (menor valor de PRIOx) define qual a saída que será acionada.

As entradas PRIOx definem a ordem de prioridade e devem ter valor entre 1 (maior prioridade) e 32 (menor prioridade).

Se várias entradas estiverem acionadas simultaneamente, a entrada com maior prioridade manterá sua respectiva saída acionada enquanto estiver verdadeira.

A ligação de uma nova saída só ocorrerá quando a entrada anteriormente ligada for desligada.

Ex: as entradas IN1 e IN2 foram acionadas simultaneamente, a saída OUT1 ligou pois IN1 tem maior prioridade em relação a IN2.



PRIORITY LOCK2 - Intertravamento de saídas (até 32) com prioridade e desligamento forçado

As entradas INx controlam as saídas OUTx, mas apenas uma saída será acionada de cada vez.

A entrada com maior prioridade (menor valor de PRIOx) define qual a saída que será acionada.

As entradas PRIOx definem a ordem de prioridade e devem ter valor entre 1 (maior prioridade) e 32 (menor prioridade).

Se várias entradas estiverem acionadas simultaneamente, a entrada com maior prioridade manterá sua respectiva saída acionada enquanto estiver verdadeira.

A ligação de uma nova saída ocorrerá sempre que uma entrada de maior prioridade for acionada.

Ex: a entrada IN2 (PRIO2) está acionada (OUT2=ON), ao acionar IN1 (PRIO1) automaticamente a saída OUT1=ON e OUT2=OFF pois IN1 tem maior prioridade.

SEQUENCE – Controle de saídas com sequência fixa e quantidade especificada

Este bloco controla até 8 saídas com sequência fixa, a quantidade de saídas controladas é estabelecida pela entrada NUM.

A sequência é iniciada a partir da borda de subida em ENABLE, as saídas ligam com sequência crescente (OUT0 -> OUT7). Desabilitar ENABLE (borda de descida) desliga as saídas com sequência decrescente (OUT7 -> OUT0).

Entradas:

ENABLE: Habilita a operação do bloco.

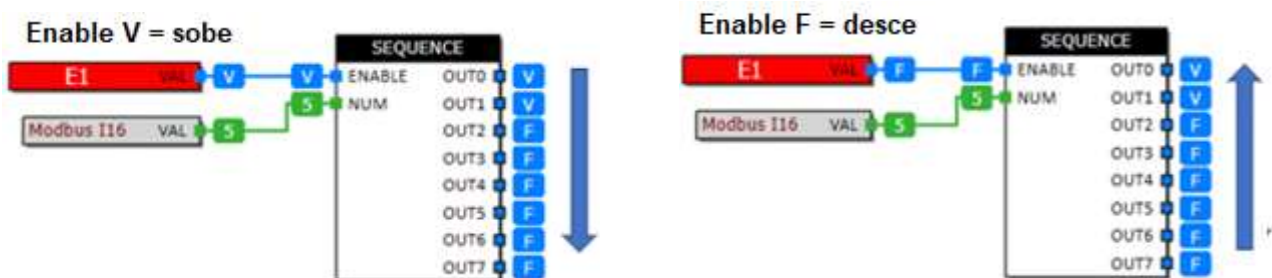
NUM: Define o número de saídas que devem ser ligadas.

Saídas: OUTx / Saída x.

Parâmetros:

DELAY_ON: Define o atraso (em décimos de segundo) entre o acionamento de cada saída.

DELAY_OFF: Define o atraso (em décimos de segundo) entre o desligamento de cada saída.



STAGING8 - Rodízio para até 8 saídas digitais por falha ou por tempo de operação

Entradas:

ENABLE: Habilita o bloco para operação. Se desabilitado, os estágios serão desligados respeitando os tempos configurados.

NUM: Define o número de estágios que devem estar ativos.

FAULTx: Se ativo, indica falha na saída x. A saída correspondente será desligada e outra acionada no lugar.

ROTATE: Quando ativado, força um rodízio de estágios.

Saídas:

OUTx: Saídas (OUT1 ~ OUT8)

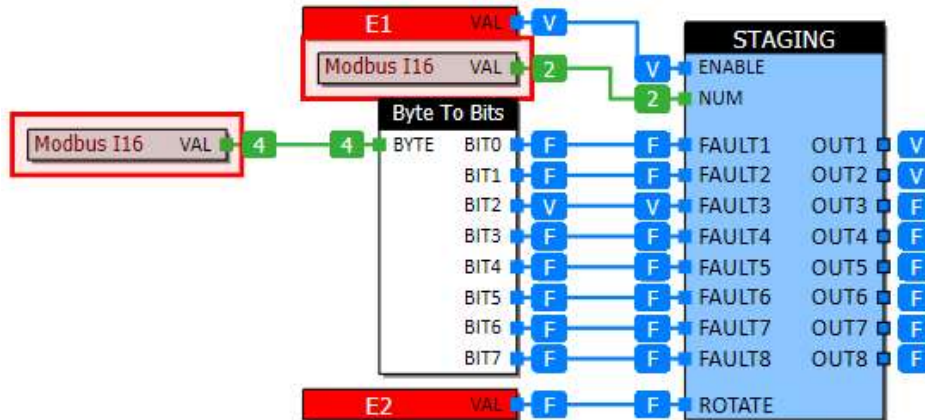
Configurações:

STAGE_DELAY: Define o delay (segundos) entre o acionamento/desligamento de cada estágio.

ROTATE_TIME: Define o tempo (em horas) de operação para forçar um rodízio das saídas. O valor mínimo desse parâmetro é 1 hora.

NUM_OUT: Define o número máximo de saídas que serão utilizadas no bloco. As saídas usadas sempre serão as da sequência 1 a N.

Ex: E1 fará o acionamento (ENABLE) de duas saídas (NUM), respeitando o tempo (STAGE_DELAY), no rodízio de 5 saídas (NUM_OUT=5). A entrada 3 está em falha (FAULT3), o rodízio será entre as entras OUT1, OUT2, OUT4 e OUT5. O acionamento do rodízio pode ser através de E2 (ROTATE) ou pelo tempo de operação (ROTATE_TIME). Sequência de acionamentos: OUT1 e OUT2 / OUT2 e OUT4 / OUT4 e OUT5 / OUT5 e OUT1.



STATUS – Monitora o estado de uma saída (feedback) por uma entrada digital

Este bloco permite utilizar uma entrada digital para monitorar uma saída digital e indicar falha caso a entrada de status não confirme que a saída foi ativada, após o tempo especificado.

Entradas:

OUT: Estado da saída a ser monitorada.

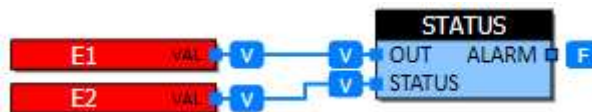
STATUS: Estado da entrada de monitoração.

Saídas:

ALARM: Se verdadeira (V), indica falha da saída monitorada.

Configurações:

DELAY: Atraso (x 100ms) para ativação da saída de alarme caso OUT e STATUS sejam diferentes.



10. Exemplos de aplicações

10.1. Função Modbus - Utilizando IHM Weintek e TCP46

Com base no mapa de endereçamentos dos drivers Modbus TCP/IP e Modbus RTU (ambos Zero-Based) da Weintek, as seguintes funções Modbus devem ser utilizadas:

3x ou **4x**: leitura/escrita de uma variável **Word (Int16)**

3x_Bit ou **4x_Bit**: leitura/escrita de uma variável do tipo Bit (**Booleana**)

5x: leitura/escrita de uma variável **Float** ou **I32 (32 bits)**.

Bit/Word	Device type	Format	Range	Memo
B	0x	DDDDD	1 ~ 65535	Output bit
B	1x	DDDDD	1 ~ 65535	Input bit (read only)
B	3x_Bit	DDDDDdd	100 ~ 6553515	Input Register bit (read only)
B	4x_Bit	DDDDDdd	100 ~ 6553515	Output Register bit
B	6x_Bit	DDDDDdd	100 ~ 6553515	Output Register bit
B	0x_multi_coils	DDDDD	1 ~ 65535	Write multiple coils
W	3x	DDDDD	1 ~ 65535	Input Register (read only)
W	4x	DDDDD	1 ~ 65535	Output Register
DW	5x	DDDDD	1 ~ 65535	4x double word swap
W	6x	DDDDD	1 ~ 65535	4x single word write
W	4x string central europe rev	DDDDD	1 ~ 65535	

Obs: Internamente no CLP a execução da lógica utiliza os formatos **INT32 para os valores inteiros** e **FLOAT IEEE32 (normalizado) para os dados reais**.

No Easy Builder Pro:

Para adicionar um driver de comunicação Modbus:

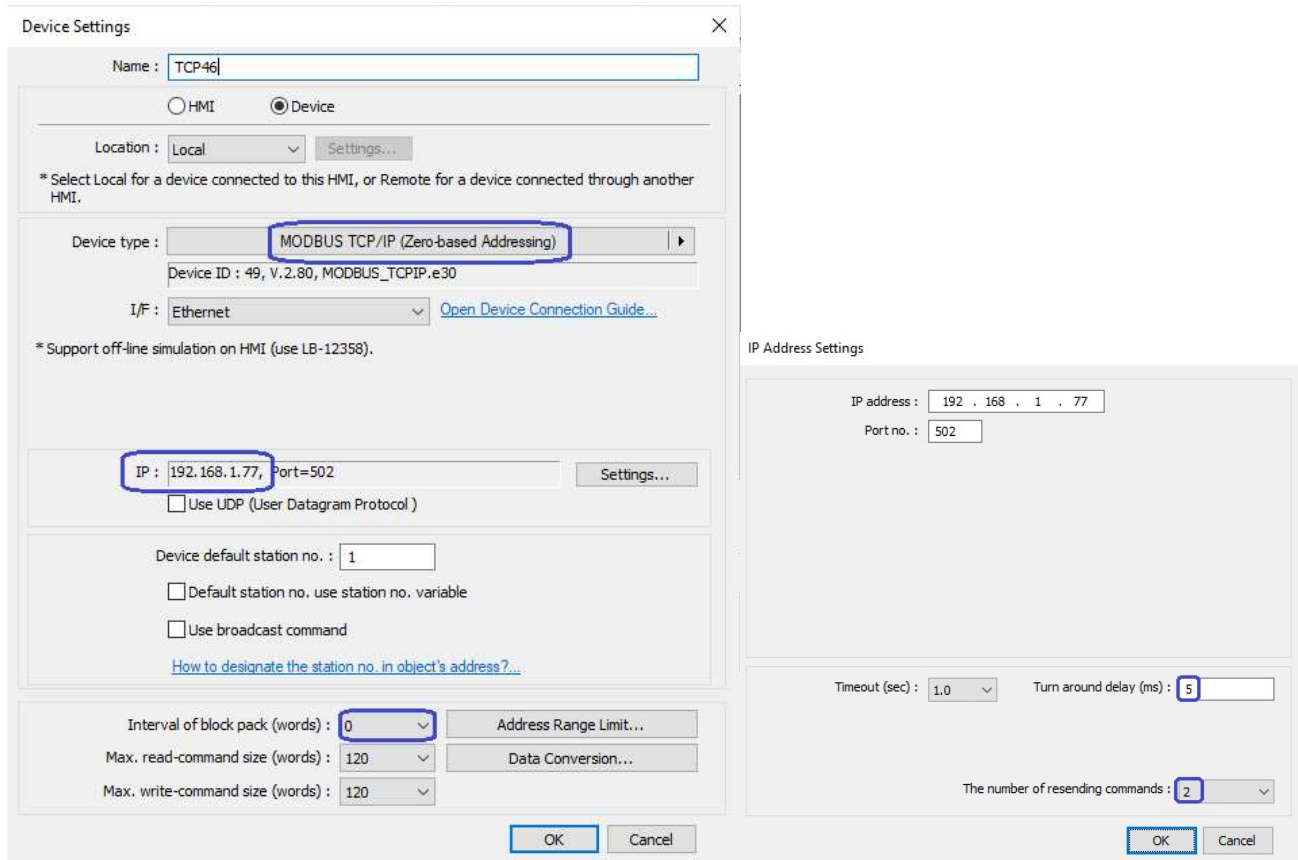
Em “System Parameter > Device Type”, adicione o driver “Modbus TCP/IP” ou “Modbus RTU” (ambos Zero-based Addressing) e inserimos o endereço de IP (Modbus TCP) ou os dados da serial definida no CLP (Modbus RTU).

Observação:

O driver da Weintek Modbus TCP/IP (zero-based) faz a comunicação em blocos para aumentar a velocidade de comunicação e atualização das telas, o tamanho do bloco por padrão é 32 endereços, delimitado pelo menor e o maior endereço, ou seja, se em uma tela existir o endereço 5 e o endereço 37 a IHM vai ler 32 endereços (5 ~ 37) a partir do endereço 5. Caso nesse intervalo exista algum endereço não mapeado no CLP, a IHM vai apresentar erro de comunicação ou poderá não mostrar os objetos nessa faixa de endereços.

No TCP46, como não existe uma tabela de endereços (cada endereço é criado individualmente) e facilmente poderá ocorrer esse problema.

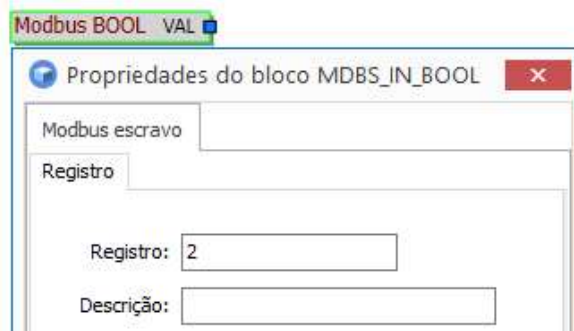
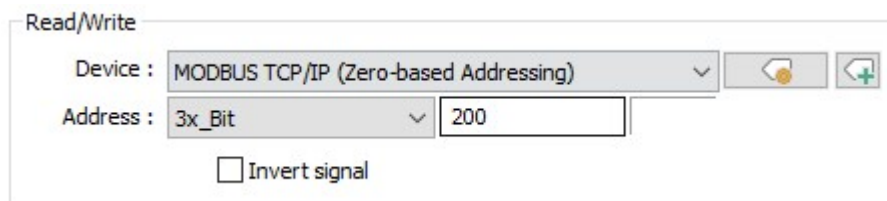
Para evitar isso, alteremos o tamanho do bloco de comunicação definido em “Interval of block pack (words)” de 32 para 0, assim cada endereço será lido individualmente.



Endereços e formatos nos objetos da IHM:

- Ler/escrever um Bit (0 ou 1)

(função = 3x_Bit ou 4x_Bit, endereço = 2 e Bit da word = 00) ou seja 3x_Bit 200
É acionado somente o primeiro bit da word, ou seja, endereço 2 = 1.



Ler/escrever em Word (função = 3x ou 4x)

Em formato, altere para 16-bit, unsigned = sem sinal e signed = com sinal.

Read/Write address

Device : MODBUS RTU (Zero-based Addressing) [v] [↶] [↷]

Address : 3x [v] 1 []

New Numeric Object

General Data Entry Format Security Shape Font

Display

Device data format : 16-bit Unsigned [v] Mask

Display format

Type : Numeric [v]

Left of decimal Pt. : 5 [] Right of decimal Pt. : 0 []

Modbus I16 VAL

Propriedades do bloco MDBS_IN_I16 [x]

Modbus escravo []

Registro

Registro: 1 []

Descrição: []

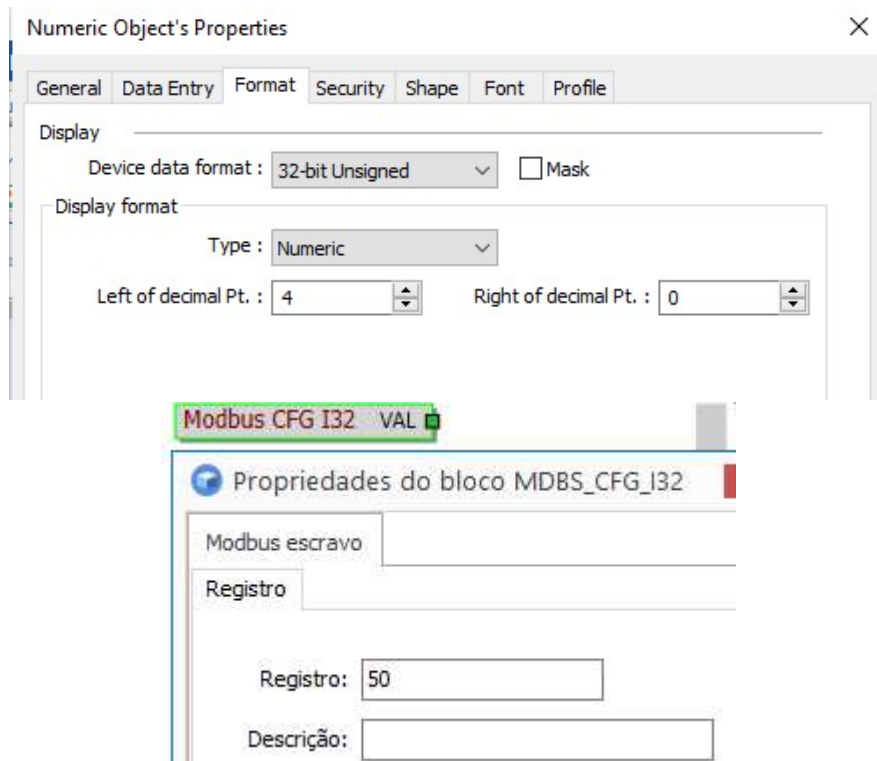
Ler/escrever em I32 (-2147483647 a 2147483647) (função = 5x)

Em formato, altere para 32-bit signed.

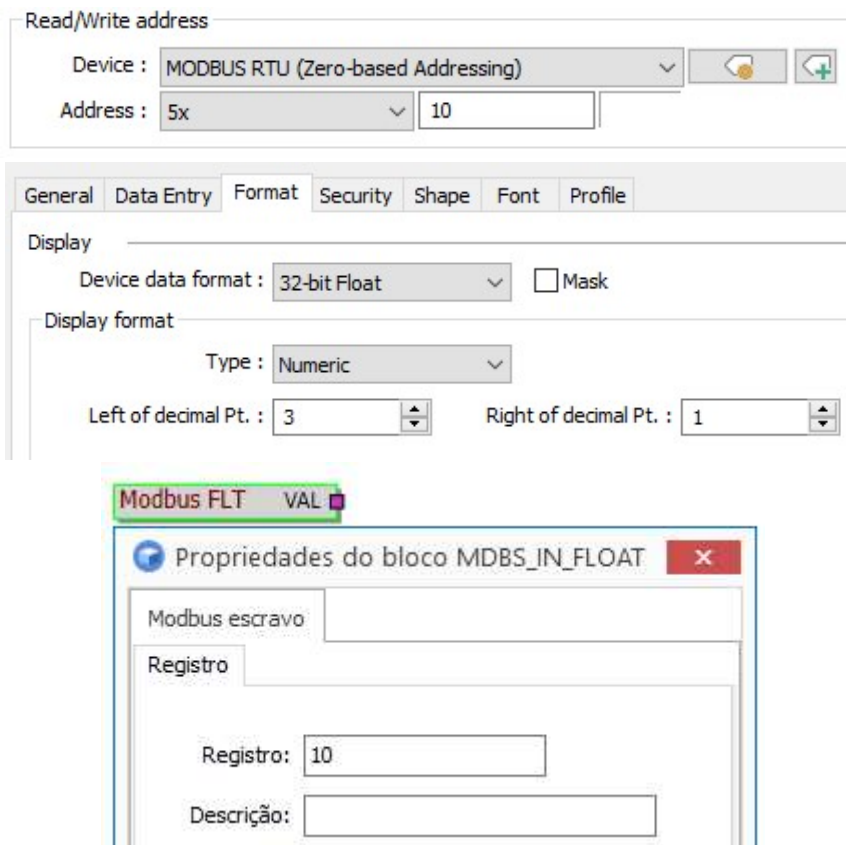
Read/Write address

Device : MODBUS TCP/IP (Zero-based Addressing) [v] [↶] [↷]

Address : 5x [v] 50 []



Ler/escrever em Float (função = 5x)
Em formato, altere para 32-bit Float.



10.2. TCP46 e módulos Tecnolog AM8DI ou AM8RO

A comunicação entre TCP46 e os módulos de expansão de entradas digitais AM8DI ou de saídas a relé AM8RO é possível através do protocolo Modbus RTU e meio físico RS485. A porta RS485 do módulo (D+ e D-) e a porta auxiliar superior ou inferior do TCP46 (D+ e D-) devem estar interligadas.

Exemplo utilizando TCP46 + AM8DI:

No módulo **AM8DI**, alteramos a **chave 3** (Protocolo Modbus RTU) e a **chave 8** (endereço 1) da DIP Switch para **ON**.

No *WebServer*, em (configurações > Modbus Mestre) definimos o TCP46 como mestre da rede, utilizando a porta RS485 auxiliar, conforme a foto a seguir:

Modbus Mestre

Geral

Timeout: ms

Tentativas:

Atraso entre leituras: ms

Modbus RTU

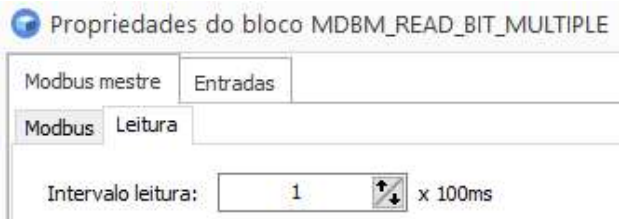
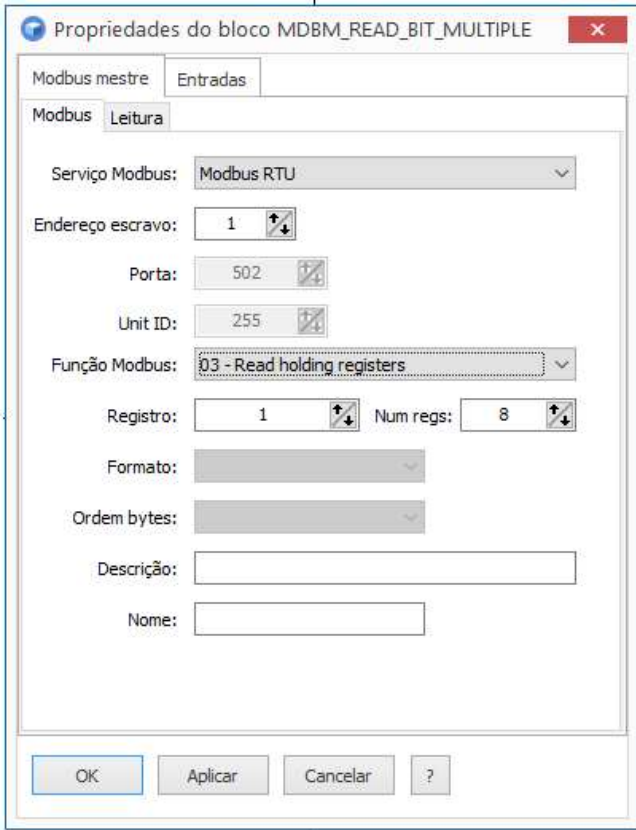
Porta RS485:

Velocidade:

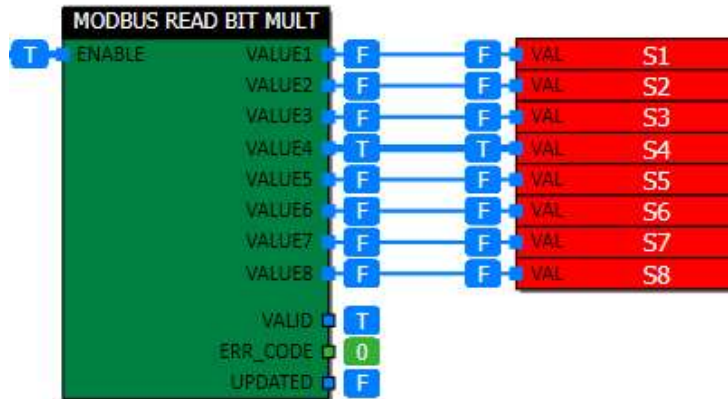
Paridade:

Stop bits:

Em um diagrama, inserimos o bloco "MDBM_READ_BIT_MULTIPLE". Definimos a função Modbus = 3x, registro = 1 (1 entrada do AM8DI) e Num regs = 8 (leitura de 8 registros), assim, o bloco realiza a leitura dos estados das 8 entradas e no caso aciona as respectivas saídas *Value x*. O endereço de escravo deve ser 1 (chave DIP 8 = on). O intervalo de leitura deve ser 100ms para uma resposta mais rápida.



AM8DI



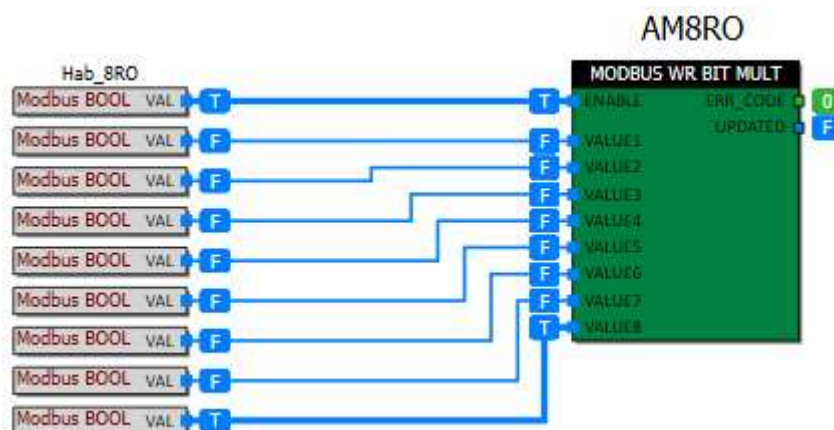
Exemplo utilizando TCP46 + AM8RO:

O módulo de saídas a relé da Tecnologia (AM8RO) fará os acionamentos das saídas a relé conforme o acionamento das entradas Modbus do tipo Bool.

Utilize um bloco Modbus mestre, agora de escrita (MDBM_WRITE_INT_MULTIPLE), definindo a função Modbus = 3x, registro = 0 e Num regs = 8.

Nesse caso, como a entrada referente VALUE8 = T, a saída 8 do AM8RO será ligada.

Para uma atuação mais rápida, na aba “Escrita” altere o parâmetro “intervalo de leitura” para 100ms.



10.3. TCP46 e módulo Tecnolog AM8C (entrada analógica)

Da mesma forma que no módulo AM8DI, a comunicação entre TCP46 e AM8C é possível através do protocolo Modbus RTU e meio físico RS485. A conexão é feita interligando os terminais D+ e D- do módulo com o CLP (conexão direta, D+ com D+ e D- com D-).

No módulo AM8C, alteramos a chave 2 (4 ~20mA), a chave 3 (Protocolo Modbus RTU) e a chave 8 (endereço 1) da DIP Switch para ON.

No *WebServer*, (configurações > Modbus Mestre), definimos a porta auxiliar como Modbus mestre conforme a foto a seguir:

Modbus Mestre

Geral

Timeout: ms

Tentativas:

Atraso entre leituras: ms

Modbus RTU

Porta RS485:

Velocidade:

Paridade:

Stop bits:

Chaves DIPs do módulo AM8C:

CHAVES 1 e 2: Tipo de entrada

Posição	Tipo entrada
00	0-20mA
01	4-20mA
10	Reservado
11	Reservado

CHAVE 4: Velocidade de comunicação:

Posição	Baud
0	9600 bps
1	19200 bps

CHAVE 3: Protocolo de comunicação

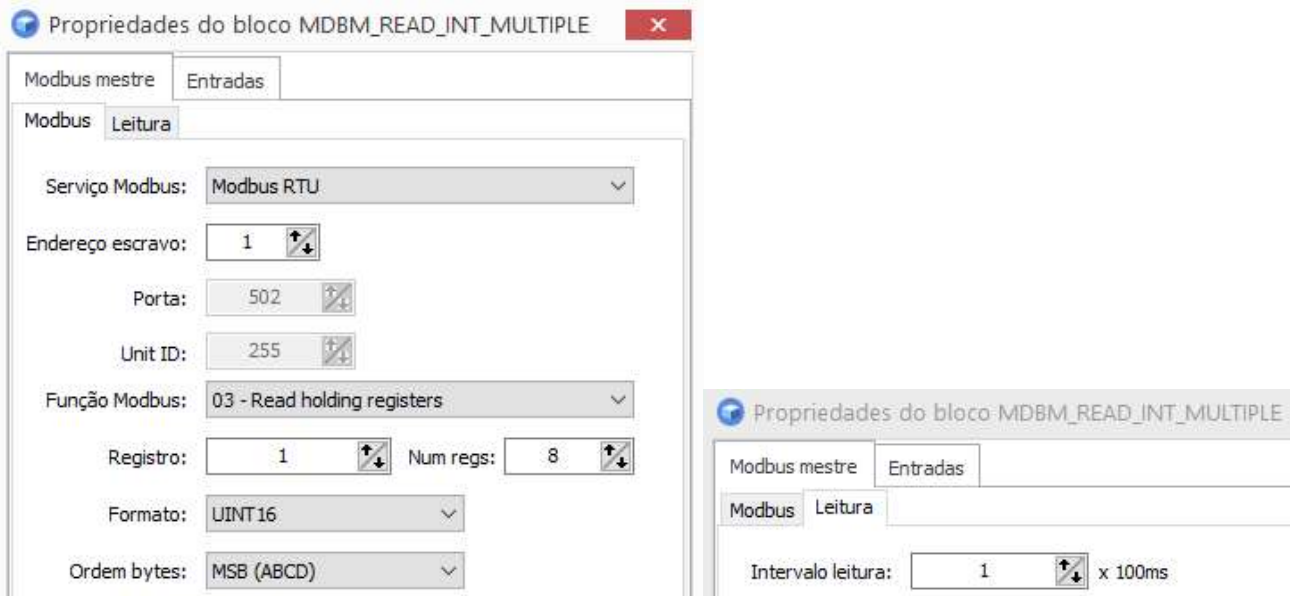
Posição	Protocolo
0	LG Inverter
1	Modbus RTU

CHAVES 5 a 8: Endereço do módulo na rede RS485 (em código binário):

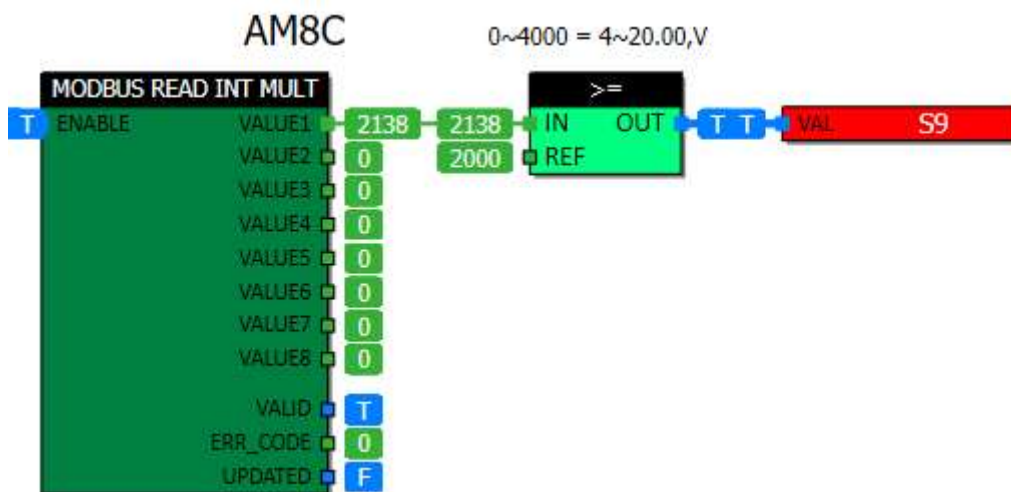
Posição	Endereço
0000	0*
0001	1
0010	2
...	...
1111	15

*O endereço 0 normalmente é reservado para o mestre da rede e não deve ser utilizado.

Em um diagrama, insira um bloco “Modbus Read Int Mult”, o endereço de escravo =1, função Modbus = 3, Registro = 1 e Num regs = 8. O registro 1 (mapa do AM8C) realiza a leitura da entrada analógica 1 (4~20mA) e fornece um valor entre 0~4000 em “Value”, sendo 0 (4mA) e 4000 (20mA), o registro Modbus = 2 fará a leitura do segundo canal até registro = 8 (leitura do canal 8). O parâmetro “intervalo de leitura” deve ser 100ms para uma coleta de dados mais rápida.



O bloco “Modbus Read Int Mult” fará a leitura dos 8 canais do módulo AM8C, o bloco “GreaterEqual_1” compara a entrada (0~4000) e se o valor for maior ou igual a referência (2000) aciona a saída 9.



10.4. IHM Weintek, TCP46 e AM8DI (Modbus TCP + Modbus RTU)

O acionamento das entradas do módulo AM8DI podem ser visualizados Bit a Bit pela IHM.

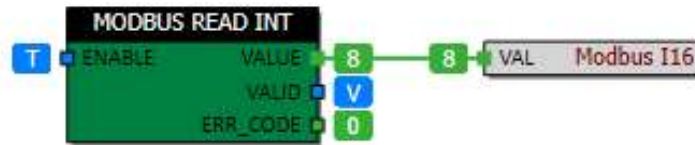
Conexão de rede:

IHM (mestre) com o TCP46 (escravo) via **Modbus TCP**.

TCP46 (mestre) com a expansão AM8DI (escravo) via **Modbus RTU** (RS485 auxiliar).

Funcionamento:

O TCP46 solicita através do bloco mestre de leitura (MDBM_READ_INT ao Registro: **0** e Endereço: **1**) o valor do byte (decimal) referente as entradas do módulo que estão acionadas, esse valor é enviado pelo bloco MDBS_OUT_I16 (Registro: **100**) para ser lido na IHM.

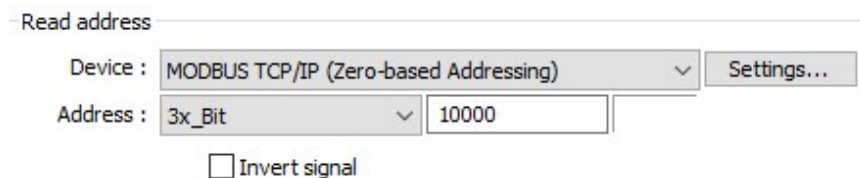


Convertendo esse valor decimal para binário é possível ver qual entrada do módulo está acionada:

- Valor **1**: 0000 0001 (Bit **0 On** – 3x_Bit 10000 – Ch1)
- Valor **2**: 0000 0010 (Bit **1 On** – 3x_Bit 10001 – Ch2)
- Valor **4**: 0000 0100 (Bit **2 On** – 3x_Bit 10002 – Ch3)
- Valor **8**: 0000 1000 (Bit **3 On** – 3x_Bit 10003 – Ch4)
- Valor **16**: 0001 0000 (Bit **4 On** – 3x_Bit 10004 – Ch5)
- Valor **32**: 0010 0000 (Bit **5 On** – 3x_Bit 10005 – Ch6)
- Valor **64**: 0100 0000 (Bit **6 On** – 3x_Bit 10006 – Ch7)
- Valor **128**: 1000 0000 (Bit **7 On** – 3x_Bit 10007 – Ch8)

Na IHM, colocamos um objeto “numeric display” com o endereço **3x100** para ler o valor do byte do AM8DI.

Inserimos uma “Bit Lamp” para cada entrada do módulo (totalizando oito), sendo o endereço 3x10000 referente ao canal 1.



Visualização do acionamento das entradas do AM8DI na IHM:



10.5. TCP46 e Inversor Veichi AC10/AC300 via Modbus RTU

Os parâmetros em preto são referentes ao modelo AC10 e os parâmetros em **vermelho** são referentes ao modelo AC300.

Configurações no WebServer:

Defina uma das portas RS485 ao canal Modbus Mestre com os parâmetros abaixo, essa é a configuração Modbus padrão dos inversores Veichi).

Modbus Mestre

— Geral —

Timeout: ms

Tentativas:

Atraso entre leituras: ms

— Modbus RTU —

Porta RS485:

Velocidade:

Paridade:

Stop bits:

Configuração no inversor:

F01.01 = 2 (comando de partida e parada via RS485) (**F00.02 = 2**)

F01.02 = 6 (comando da frequência do motor via RS485) (**F00.03 = 6**)

Registros:

Os endereços Modbus RTU do Veichi AC10 é apresentado em Hexadecimal e o endereço do bloco Modbus Mestre (MDBM) do TCP46 é definido em decimal. Para isso, cada endereço utilizado da tabela Modbus do inversor deve ser convertido para decimal.

Escrita da frequência: 3000 (hexa) > **12288** (**12228**) (decimal)

Comando de partida: 3001 (hexa) > **12289** (**12229**) (decimal)

Rampa de aceleração: 300E (hexa) > **12302** (**14**) (decimal)

Rampa de desaceleração: 300F (hexa) > **12303** (**15**) (decimal)

Address	Name	Read/Wr ite	dimension (range)	Description
0x2000 /0x3000	Given frequency	RW	0.01Hz (0.00~320.00)	Communication given frequency
0x2001 /0x3001	Command given	W	0x0000 (0x0~0x0103)	0x0000: invalid 0x0001: Forward run 0x0002: Reverse run 0x0003: Forward jog 0x0004: Reverse jog 0x0005: Deceleration stop 0x0006: Free stop 0x0007: Reset command 0x0008: Run prohibition command Communication writes 8 to 3001 address, the inverter is free to stop, need to write 9 to 3001 or power on again to run 0x0009: Run permission command 0x0101: Equivalent to F2.07 =1 [rotation parameter auto-tuning], plus run command 0x0102: Equivalent to F5.07 = 2 [static parameter auto-tuning], plus run command 0x0103: Equivalent to F5.07 = 3 [stator resistance setting], plus run command
F01.22 (0x0116) RUN	Acceleration time 1		V/F SVC The time required for the output frequency to accelerate from 0.00 Hz to the time reference frequency. 1 to 65000 s (F01.21 = 0) 0.1 to 6500.0 s (F01.21 = 1) 0.01 to 650.00 s (F01.21 = 2)	Model setting (0.01 ~ 650.00s)
F01.23 (0x0117) RUN	Deceleration time 1		V/F SVC The time it takes for the output frequency to decelerate from the time reference frequency to 0.00 Hz.	Model setting (0.01~650.00 s)

Diagrama TCP46: Os valores são inseridos na IHM e enviado pelo CLP via Modbus TCP, e o CLP (Mestre Modbus RTU) escreve as informações no Inversor (Escravo Modbus RTU) via RS485.



Programa da IHM Weintek

Frequência	ACC
30.00	5.00
Comando	DEC
3	5.00

Comando usando o objeto "numeric":
Se o endereço 12289 receber:
 1 = sentido horário
 2 = sentido anti-horário
 3 = parada por rampa de desaceleração

Obs: se preferir, crie três botões com o objeto "Set Word", cada objeto com um valor constante (acima mostrado) para os comandos, ao invés de inserir o número por um "numeric" acione os botões.

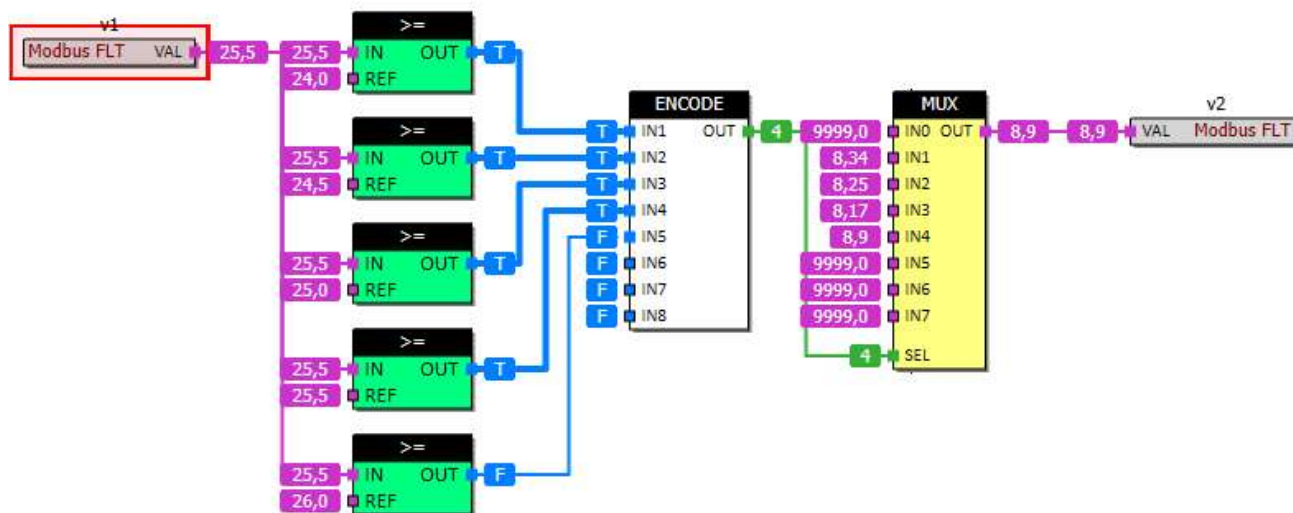
10.6. Bloco ENCODE para comparação de faixa de valores

Para criarmos uma função com até 8 comparações de entrada, usamos o bloco ENCODE em série com o bloco MUX. As saídas das comparações são ligadas as entradas do bloco ENCODE, a saída com o valor da entrada acionada é ligada ao terminal SEL do bloco MUX, que envia para a saída um valor definido nas entradas (fixo ou editável/Modbus), conforme SEL.

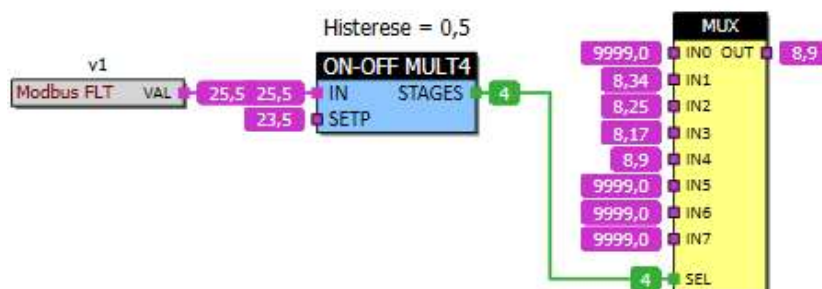
No exemplo são feitas 4 comparações dentro dos limites de 24 a 26 Vdc (faixas de 0,5 Vdc) e caso o valor de entrada esteja fora dos limites, saturamos a saída (Out = 9999).



IN	INx (enc) = Out (enc) = Sel (mux)	Out (mux)
< 24	0	9999,0
24	1	8,34
24,5	2	8,25
25	3	8,17
25,5	4	8,9
> 26	5	9999,0



É possível realizar o mesmo exemplo utilizando o bloco ONOFF_4, com histerese = 0,5, porém com apenas 3 estágios, pois o quarto será utilizado como fim de escala.



10.7. Rampas e patamares com o bloco RAMP

Nesse exemplo, realizamos 3 rampas, mas é possível o controle de até 15 rampas utilizando os blocos MUX16.

Para evitar o início da rampa logo após passar o clp para o modo “rodar”, inicie SEL=0 e RAMP OUT=0.

Após um pulso na entrada E1/DI1, o sistema é iniciado, o contador incrementa 1 em COUNT, tornando SEL=1 e passando para a segunda linha da tabela (IN=50 e TIME=5). A saída OUT funcionará em rampa (a saída atualiza o seu valor até se igualar a IN, no caso de 0 para 50). Quando a saída chegar ao patamar desejado IN=OUT=50, a saída “DONE” irá acionar um temporizador (contendo o valor de permanência, dado pelo MUX com TIME=10). Após o tempo decorrido, a saída do temporizador irá incrementar o contador, tornando SEL=2, pulando para a linha 3 da tabela (IN=100 e TIME=5) e operando da mesma forma que SEL=1. O mesmo procedimento é feito quando SEL=3 (IN=150 e TIME=5). Quando SEL=4, o contador é resetado pelo bloco de igualdade, a saída RAMP OUT é decrementada, COUNT=SEL=0 e o sistema aguarda um novo acionamento em E1.

Lógica de blocos:

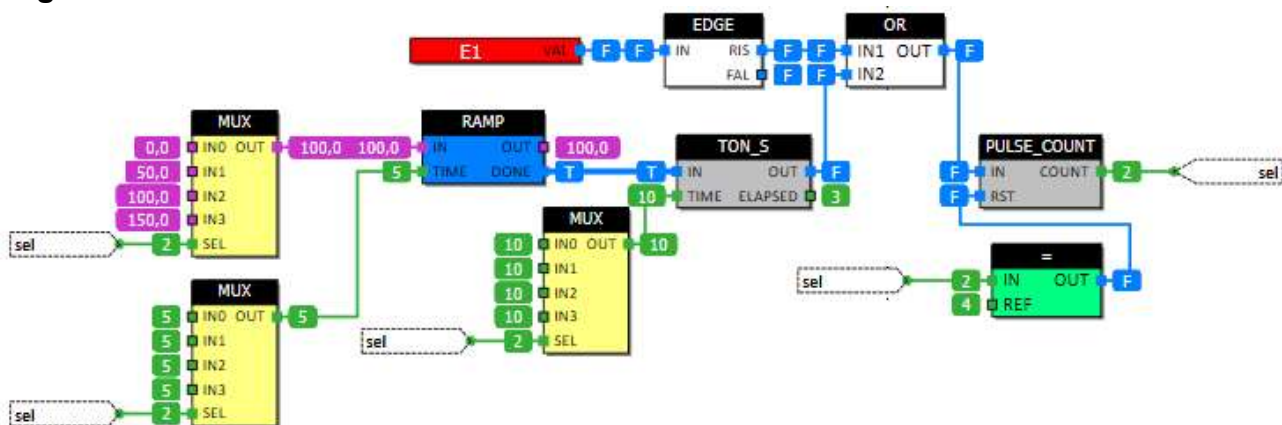
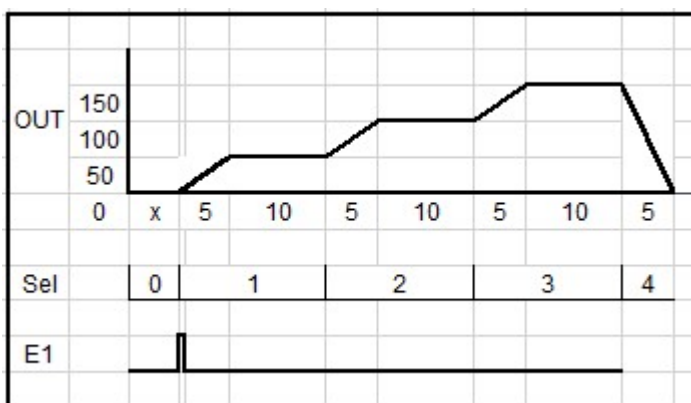


Tabela de dados e respostas:

SEL	IN	TIME (s)	T Permanência (s)
0	0	5	aguardando E1
1	50	5	10
2	100	5	10
3	150	5	10
4	0	5	10



Obs: Os blocos MUX contém os valores da tabela do bloco RAMP (os blocos MDBS_CFG _FLOAT ou _INT nas entradas INx dos MUXs, tornam esses valores definidos via Modbus, por exemplo, através das receitas da IHM).

10.8. Rede para acesso remoto VPN em IHMs Weintek

O acesso remoto VPN permite acessar os equipamentos ligados a rede local da IHM usando a Internet para manutenções técnicas, buscando o monitoramento e resolução de problemas de forma segura.

Permitem VPN

- Em IHMs que possuem 2 portas LAN, a porta **LAN 1** (principal) **deve estar conectada** à internet e a porta **LAN 2** não está vinculada a rede Internet, mas permite a comunicação ponto a ponto entre IHM e CLP.



- Em IHMs que possui **somente 1 porta LAN** (LAN 1), todos os dispositivos devem usar IPs dentro da faixa de rede da internet local.



- O acesso VPN ao TCP46 é possível através da porta serial Modbus RTU RS485. A LAN 1 deve estar conectada à internet (com o IP dentro da faixa de rede local) e a porta serial deve estar conectada a uma das portas RS485 (D+ e D-) do CLP. No TProg, a conexão da interface para download deve estar definida para Modbus RTU.



10.9. IHM Veichi Vi20 e TCP46

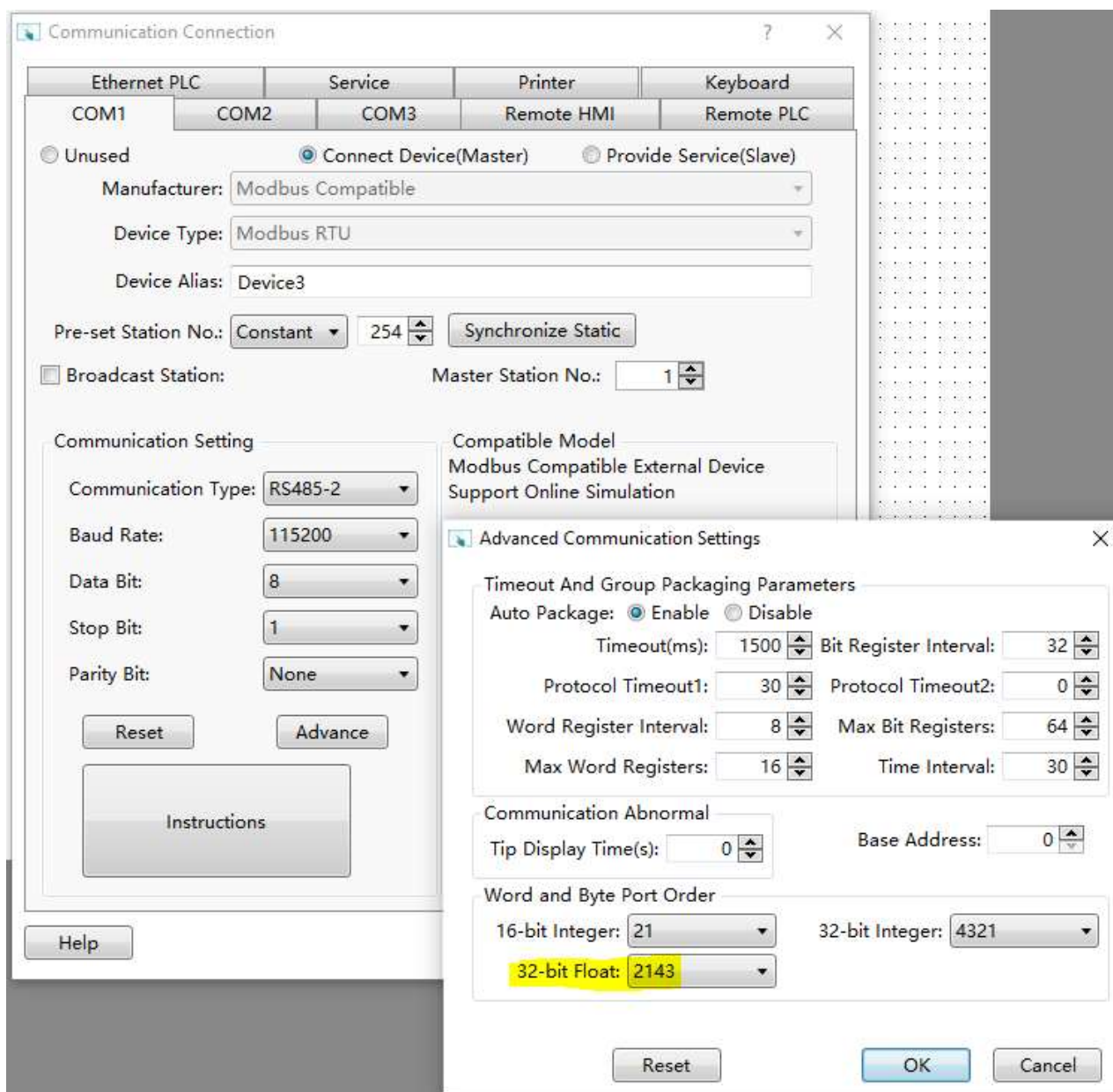
A comunicação entre a IHM Veichi Vi20 com o CLP Technolog TCP46 é possível usando o protocolo Modbus RTU e meio físico RS485.

No software Vi20 Studio:

Adicionar o driver: selecione a porta COM a ser usada, click em “Connect Device (Master)” > “Modbus Compatible” > Modbus RTU

Parâmetros da rede RS485: em “Pre-set Station No.:" defina o número escravo modbus do CLP (default = 254), Type: RS485-2, Baud Rate: 115200 kbps, Data bit: 8, Stop Bit:1 e Parity bit: none.

Parâmetros avançados: click em “Advance”, altere “Base Address” = 0, “Timeout (ms)” = 1500 e “32-Bit Float” = 2143 (swap necessário para usar variável Float).



No Webserver do TCP46:

Para acessar o Webserver digite o IP do CLP em um navegador de internet (IP padrão: 10.1.1.240 e login/senha padrão: config).

Para configurar os dados da rede Modbus acesse: Configuração > Modbus Escravo.

Defina a porta RS485 (principal ou auxiliar) e verifique os parâmetros de rede (devem ser iguais aos que foram inseridos no driver da IHM).

Modbus Escravo

Endereço Modbus:

Porta 1

Porta RS485:

Velocidade:

Paridade:

Stop bits:

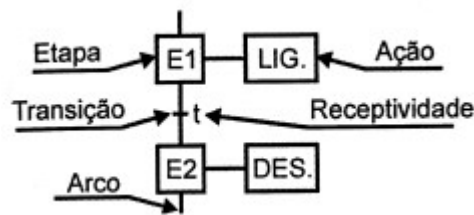
Atraso transmissão: ms

Apêndice A - Programação de sequências (Grafcet)

O Grafcet é uma técnica para descrever comportamentos sequenciais e fornecer uma modelagem lógica mais facilitada e organizada do que a representação *Ladder*. Possui a característica de representar graficamente o comportamento de um sistema automatizado.

Embora o TCP46 não possua a linguagem GRAFCET, utilizamos os conceitos dessa linguagem em conjunto com o bloco máquina de estados para montar uma sequência de programação.

Elementos do Grafcet: estado/etapas, transições, arcos, receptividade, ações e regras de evolução.



- Sequências

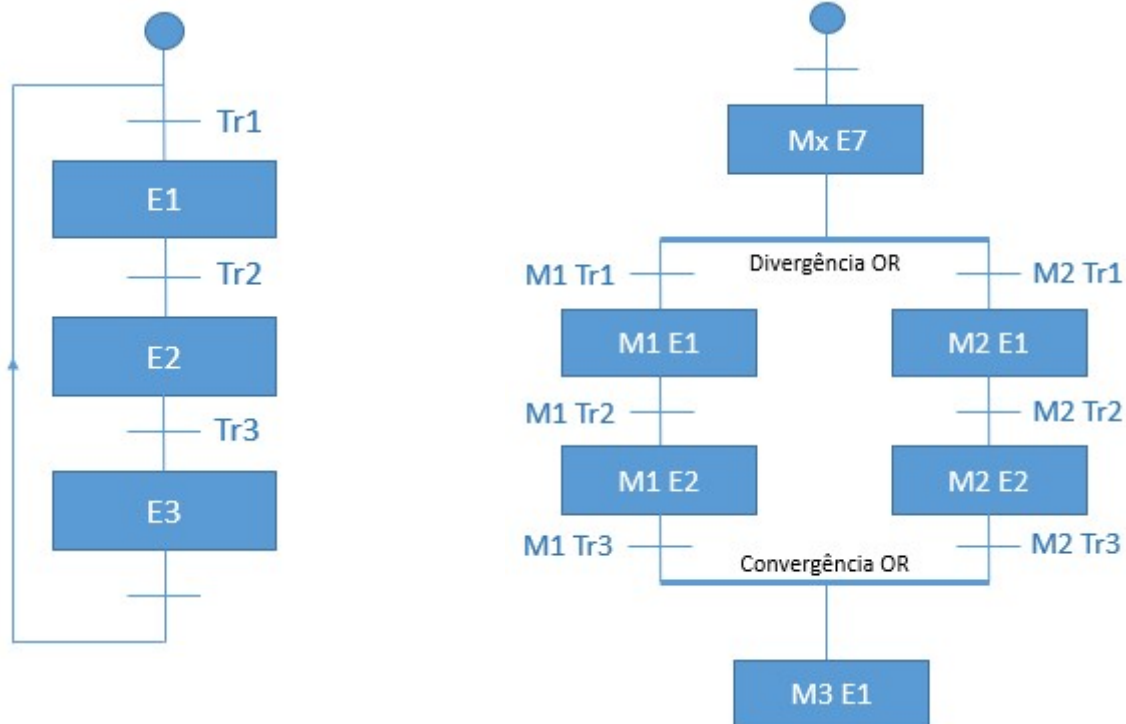
Sequência única

É uma cadeia de etapas e transições dispostas de forma linear.

Uma etapa é seguida de apenas uma transição, e uma transição é seguida de apenas uma etapa.

Sequência paralela

Sequências paralelas são as sequências únicas que são ativadas de forma simultânea por uma mesma transição.



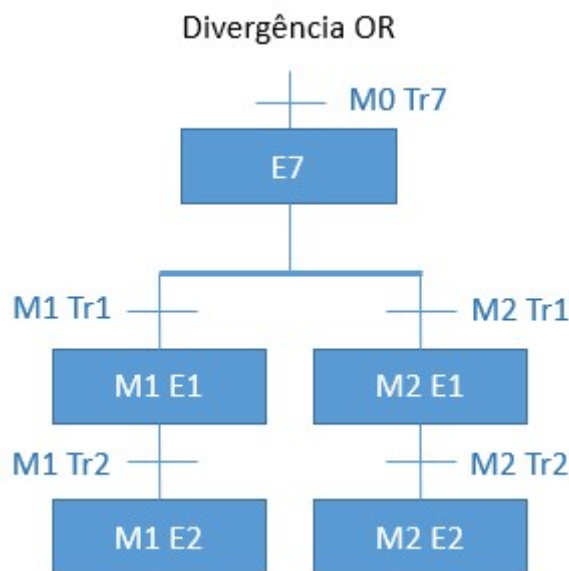
Estruturas lógicas

Divergência OR

Uma divergência OR/seletiva é precedida por uma etapa e sucedida por sequências iniciadas por transições.

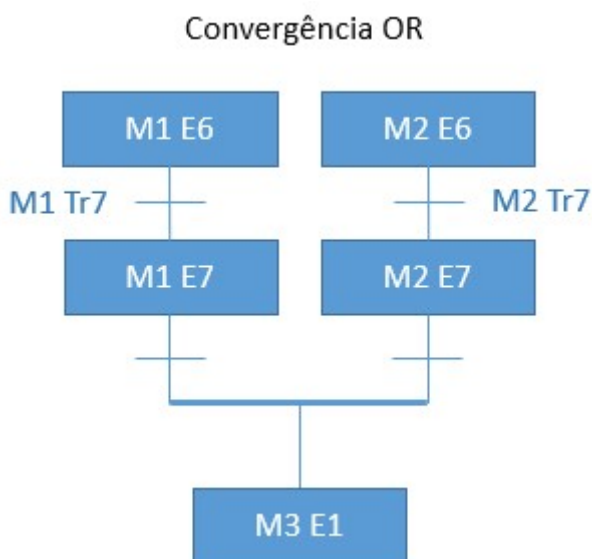
A máquina M1 E1 ativa se: Mx E7 ativa, e satisfaça a receptividade da transição M1 Tr1.

A máquina M2 E1 ativa se: Mx E7 ativa, e satisfaça a receptividade da transição M2 Tr1.



Convergência OR

Uma convergência OR/seletiva é sucedida por uma etapa e precedida por sequências finalizadas por transições.



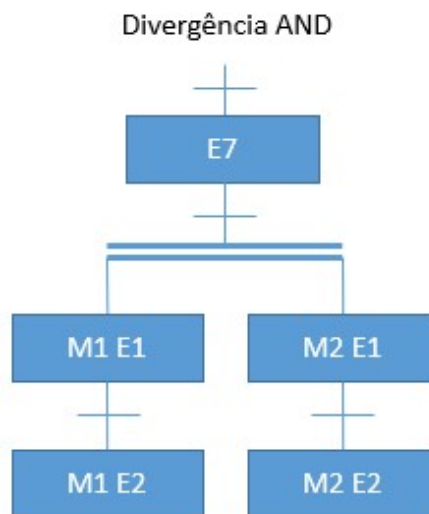
Paralelismo e sincronicidade

Configurações quando necessitamos que duas ou mais sequências (máquinas de estados) devem ser executadas ao mesmo tempo.

O paralelismo só é encerrado quando todas as suas sequências estiverem concluídas (sincronicidade).

Divergência AND

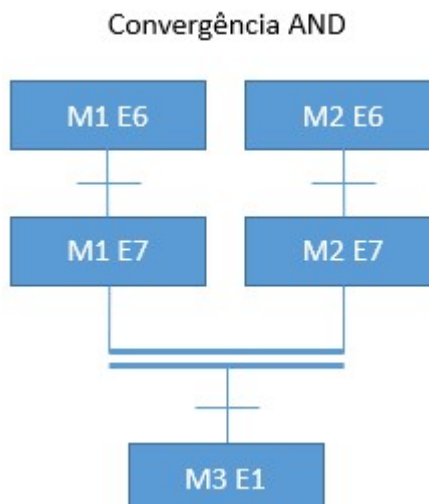
Uma divergência simultânea é precedida por uma transição e sucedida por sequências iniciadas por etapas.



Convergência AND

Uma convergência AND/simultânea é sucedida por uma transição e precedida por sequências terminadas por etapas.

O paralelismo só é encerrado quando todas as suas sequências estiverem concluídas (sincronicidade).



Máquina de Estados de Moore/Sequência de operações

A máquina de estados é o melhor método para descrever operações sequenciais e também permite uma fácil visualização do fluxo de controle.

A máquina de estados é composta por estados e transições:

Um **estado** descreve uma situação no comportamento sequencial do sistema, onde apenas um estado pode estar ativo de cada vez (estado atual). Os estados anteriores não influenciam o comportamento e são ignorados pela lógica, e a única possibilidade de saída desse estado é a transição que leva o sistema para o próximo estado.

A passagem de um estado para o outro é feita por uma **transição**, que é a condição lógica que precisa ser satisfeita para que a transição ao próximo estado ocorra.

Essa técnica de descrição de sistemas dispensa os intertravamentos da lógica combinatória convencional (*ladder logic*) pois apenas os eventos relativos à situação atual do sistema (estado atual) são avaliados pela lógica.

Exemplo: o processo de uma furadeira automática.

Descrição: o operador aperta o botão (E1), ligando o motor para descer o conjunto mecânico (S2) e o motor da broca (S1). Ao encontrar o fim de curso inferior (E2), o motor para de descer o conjunto mecânico é desligado e inicia uma contagem de tempo de 5 segundos. Logo após, desliga o motor da broca (S1) e liga o motor do conjunto mecânico para subir (S3). Quando encontrar o fim de curso superior (E3) o processo é finalizado.

Entradas:

E1: Botão para ligar (não retentivo)

E2: Fim de curso inferior (FC1)

E3: Fim de curso superior (FC2)

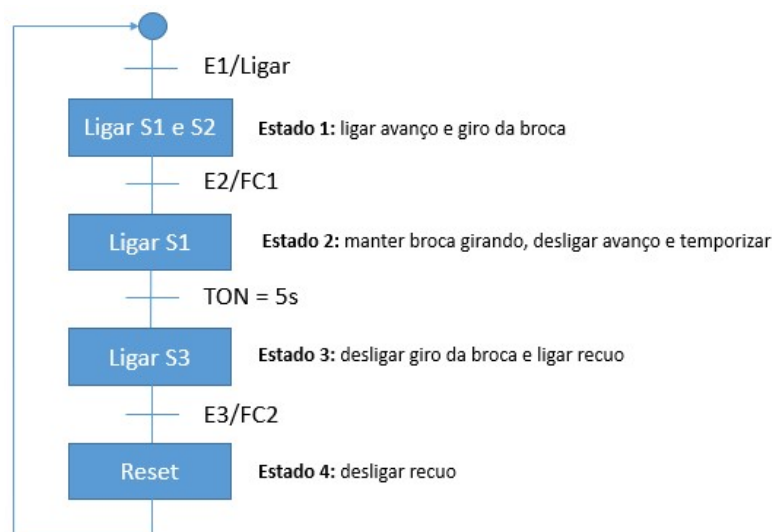
Saídas:

S1: Giro da broca

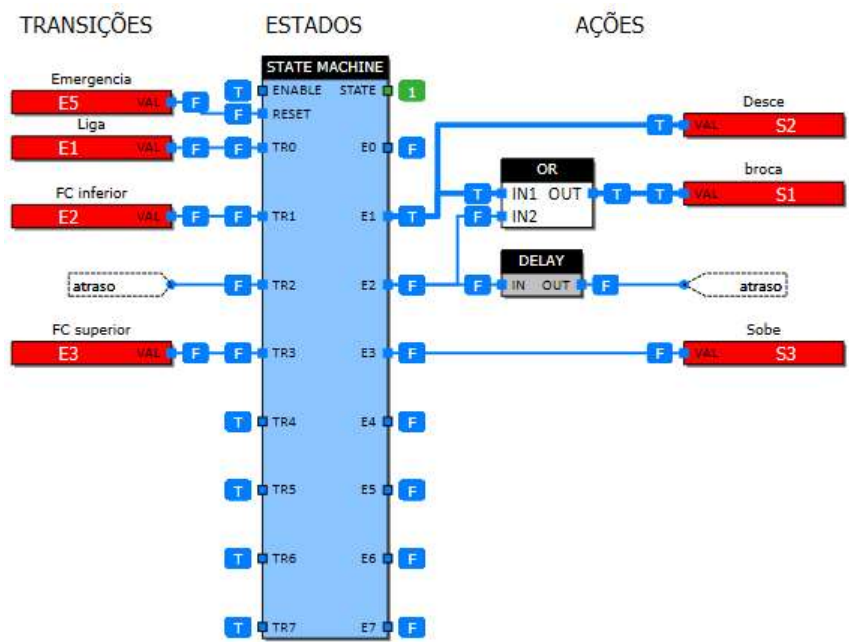
S2: Desce o conjunto mecânico

S3: Sobe o conjunto mecânico

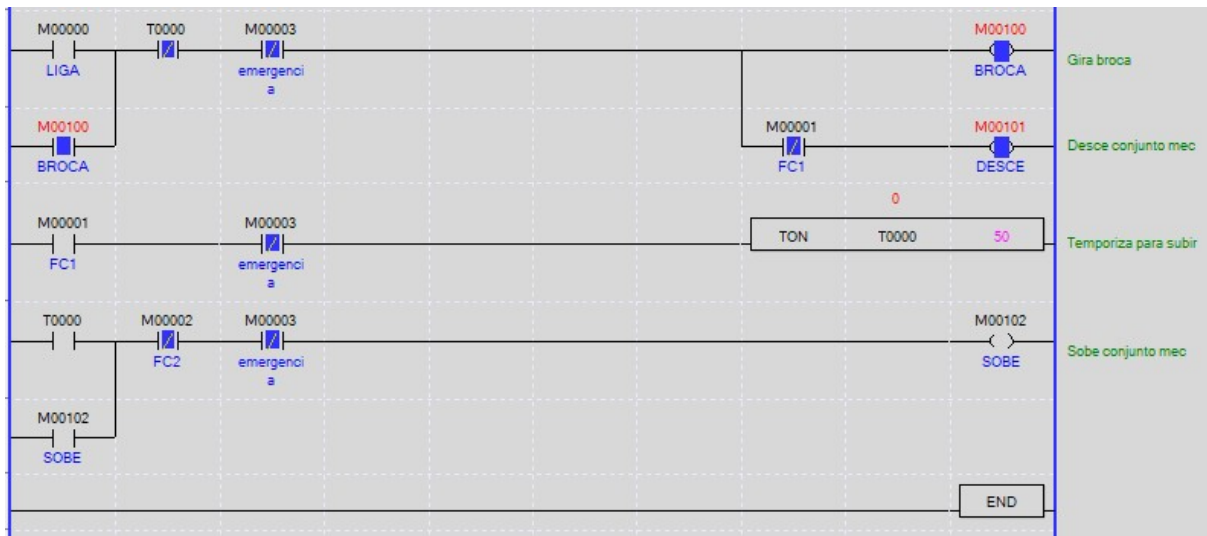
Máquina de estados da furadeira automática:



Programação do bloco State Machine para a operação da furadeira automática:



Lógica Ladder equivalente para a operação de furadeira automática:



A construção na lógica Ladder é mais complexa e exige maior experiência do programador para se chegar a um programa compacto como o acima apresentado, enquanto que na máquina de estados a construção é basicamente intuitiva e **segue diretamente a descrição do funcionamento da máquina.**

Nesse exemplo comparamos as duas linguagens de programação e podemos observar que o bloco de função **permite visualizar em qual dos 4 estados a máquina se encontra** em um determinado instante, **facilitando o diagnóstico de problemas.**

Outra vantagem é um único terminal para a função de emergência, enquanto que no *Ladder* é necessário um contato NF para intertravamento em cada linha lógica.

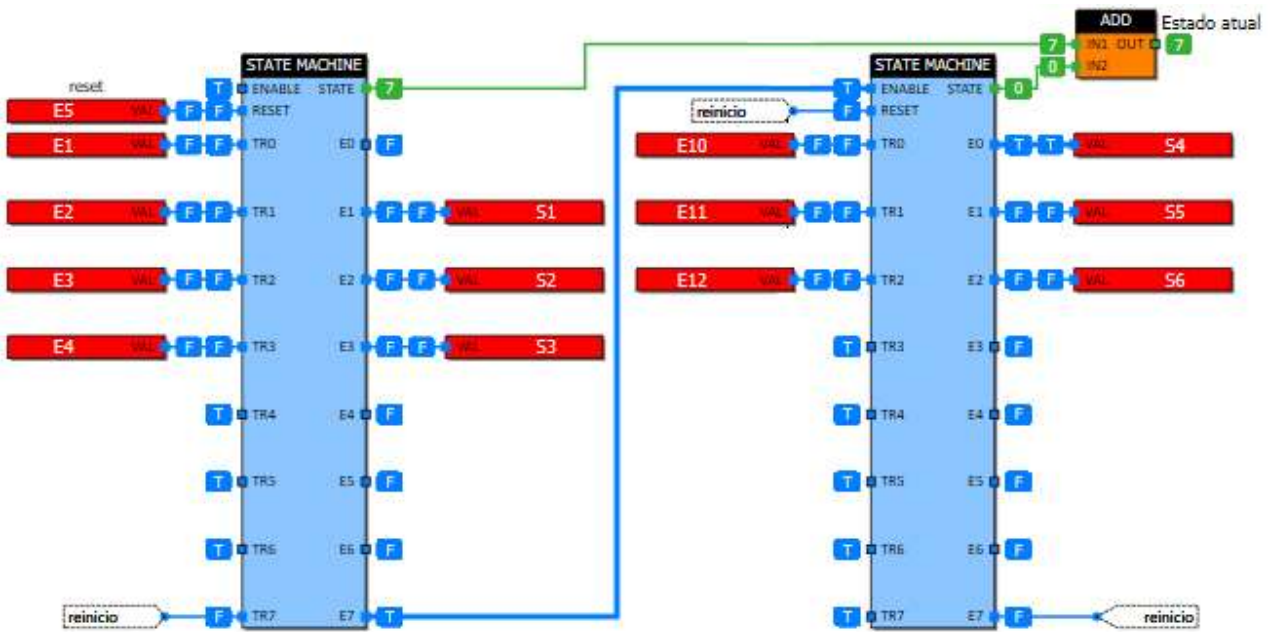
Máquina de Estados em Série

A ligação de máquinas de estados em série possibilita a ampliação do número de estados, através da interligação do estado E7 de um bloco ao terminal Enable do próximo bloco, aumentando a quantidade de estados em múltiplos de 7.

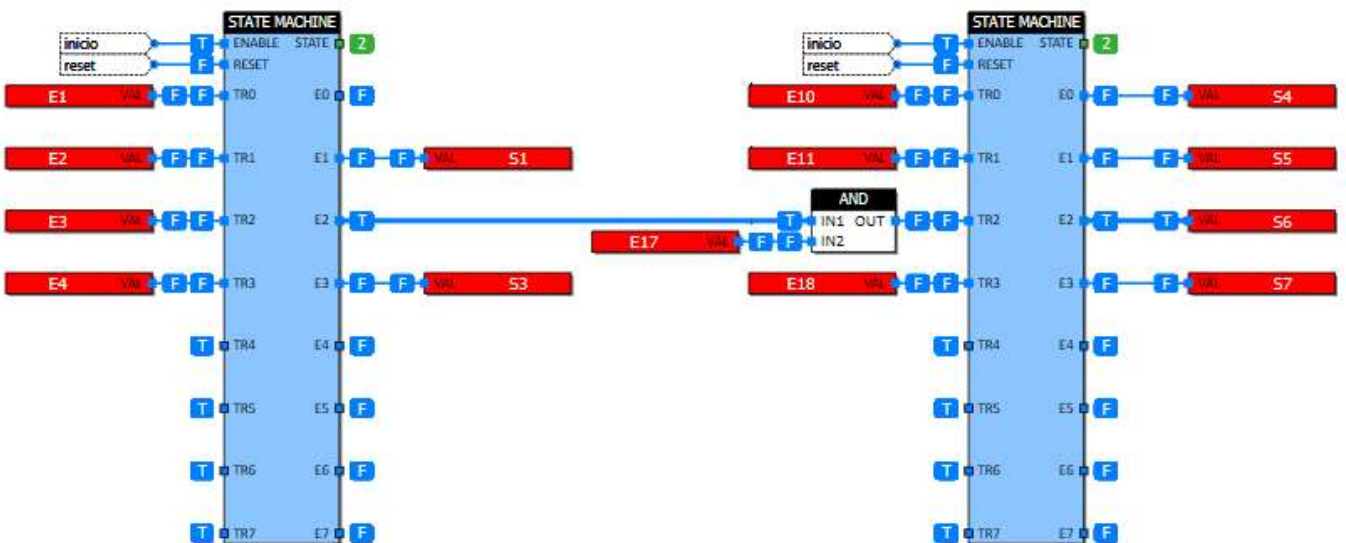
O último estado do último bloco precisa resetar todos os blocos, desabilitando todas as saídas, colocando a máquina inicial no estado 0 e permitindo condições para o início de um novo ciclo.

Os estados e transições não utilizados são ignorados.

No exemplo abaixo temos uma máquina de 15 estados e o estado atual pode ser monitorado pela soma dos terminais STATE. No caso, $7+0 = 7$ (estado 7).



Na configuração abaixo o estado 2 da máquina 1 condiciona a transição 2 da máquina 2, indicando que a sua sequência depende da operação da primeira máquina.



Máquina de estados em modo passo a passo

O acionamento da entrada Enable por uma borda de subida (Edge) permite controlar a máquina em modo passo a passo, pois nesse caso a condição fica válida por apenas 1 ciclo e o bloco permite apenas uma transição de estado a cada ciclo.

Mesmo com todas as transições válidas, apenas uma troca de estado ocorrerá a cada ciclo, ou seja, são necessárias no mínimo 7 varreduras para completar um ciclo do bloco.

Se Enable permanecer acionado, o funcionamento da máquina de estados fica em modo automático.

No exemplo abaixo demonstramos como fazer o condicionamento da entrada Enable para utilizar esse recurso.

