

Manual dos CLPs

Tecnolog TCP46 e TCP46A

Versão: 1.61

TECNOLOG

www.tecnolog.com.br

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

 Telefone: (51) 3076.7800

 E-mail: vendas@tecnolog.ind.br

Esse índice utiliza hiperlinks para fornecer acesso instantâneo as informações desejadas, basta clicar no tópico desejado.

1. Componentes do sistema

- 1.1. Descrição da unidade básica e expansão
- 1.2. Características técnicas e especificações ambientais
- 1.3. Terminais da unidade básica e de expansão
- 1.4. Dimensões
- 1.5. Instalação e manutenção

2. Conexão das entradas/saídas digitais e analógicas

- 2.1. Ligações da unidade básica TCP46 e expansão TCP46EX
 - 2.1.1. Entradas digitais
 - 2.1.2. Entradas NTC, contato seco ou NPN
 - 2.1.3. Saídas digitais
 - 2.1.4. Entradas analógicas
 - 2.1.5. Saídas analógicas
- 2.2. Ligações da unidade básica: TCP46A
 - 2.2.1. Entradas digitais / NTC
 - 2.2.2. Saídas digitais
 - 2.2.3. Entradas analógicas
 - 2.2.4. Saídas analógicas

3. Interfaces de comunicação

- 3.1. Sistemas de comunicação disponíveis
- 3.2. Cabos de programação e comunicação
 - 3.2.1. Protocolo Modbus RTU
 - 3.2.2. Protocolo Modbus TCP/UDP

4. Configurações iniciais do TCP46 pelo WebServer

- 4.1. Acessando o WebServer
- 4.2. Recursos permitidos no WebServer

5. Software TProg

- 5.1. Instalação do software TProg
- 5.2. Abas de navegação
- 5.3. Atalhos do TProg

- 5.4. Iniciando um projeto
- 5.5. Novo diagrama / Importar diagrama
- 5.6. Ferramentas

6. Dados, operação e execução

- 6.1. Linguagem de blocos de função
- 6.2. Tipos de dados
- 6.3. Interligar blocos de função

7. Programando com o TProg

- 7.1. Inserindo um bloco de função
- 7.2. Identificação dos blocos
- 7.3. Label
- 7.4. Filtro das entradas digitais
- 7.5. Entradas digitais/NTC e entradas analógicas no TCP46A
- 7.6. Configurando entradas e saídas analógicas
- 7.7. Configurando feriados e horário de verão
- 7.8. Interfaces de conexão entre PC e CLP
 - 7.8.1. Conexão Modbus UDP
 - 7.8.2. Conexão Modbus RTU utilizando o conversor serial US485
- 7.9. Modos de operação (rodar, pausar e *debug*)
- 7.10. Compilação
- 7.11. Download (PC > TCP46)
 - 7.11.1. Soluções para erros de conexão
- 7.12. Monitoramento online
- 7.13. Forçando entradas digitais físicas
- 7.14. Tempo de varredura / Scan
- 7.15. Ordem de execução dos blocos
- 7.16. Impressão de diagramas
- 7.17. Endereçamento do TCP46EX e configuração no TProg
- 7.18. Tag de comunicação Mestre e Escravo
 - 7.18.1. Tag Modbus Mestre (MDBM)
 - 7.18.2. Tabela de erros Modbus Mestre
 - 7.18.3. Tag Modbus Escravo (MDBS)
- 7.19. Lista de registros Modbus (Tags)
- 7.20. Importar Tags na IHM Weintek (EasyBuilder Pro)

8. Tabela de endereços Modbus

9. Instruções de programação (Blocos)

9.1. Grupo IO – Entradas e saídas digitais e analógicas

BI – Entrada digital/Nx (contato seco)
BO – Saída digital
AI – Entrada analógica
AO – Saída analógica
PULSE_FREQ – Entrada de contagem rápida
PULSE_INPUT – Contador de pulso rápido
EXP_BI - Entrada digital/Nx (contato seco) da expansão
EXP_BO - Saída digital da expansão
EXP_AI - Entrada analógica/NTC da expansão
EXP_AO - Saída analógica da expansão
EXP_STATUS – Status de conexão das expansões

9.2. MDBS - Operandos para a comunicação Modbus escravo

MDBS_CFG_(BOOL/FLOAT/I16/I32)
Operando retentivo que permite a um dispositivo mestre ler um dado de entrada
MDBS_IN_(BOOL/FLOAT/I16/I32)
Operando volátil que permite a um dispositivo mestre ler um dado de entrada
MDBS_OUT_(BOOL/FLOAT/I16/I32)
Operando volátil que permite a um dispositivo mestre ler um dado de saída
MDBS_SCHEDULE – Programação horária via agenda do bloco ou rede Modbus

9.3. Grupo lógica - operações com as entradas

AND, AND3 - Operação AND (E) de duas ou três entradas
OR, OR3 - Operação OR (ou) de duas ou três entradas
NAND - Operação NAND (E negado) de duas entradas
NOR - Operação NOR (ou negado) em duas entradas
NOT - Operação NOT (negado)
XOR/XOR3 - Operação lógica (OU exclusivo) de duas ou três entradas
EDGE - Detecta a borda de subida/descida do sinal de entrada e copia para a saída
SET_RESET - Controla o estado da saída por um pulso nas entradas SET ou RST
TOGGLE - Inverte o status da saída OUT a cada borda de subida em IN
ENCODE - Identifica a entrada ativa mais significativa
BITS_TO_BYTE - Conversor de Bits para Byte
BITS_TO_WORD - Conversor de Bits para Word
BIT_COUNT - Contabiliza até 8 entradas ativas
BYTE_TO_BITS - Conversor de Byte para Bits
WORD_TO_BITS - Conversor de Word para Bit
EXTRACT - Extrai bits de uma palavra
MUX2_B - Multiplexador binário de 2 entradas
MUX8_B/MUX16_B - Multiplexador binário de 8 ou 16 entradas digitais
DEMUX2 - Demultiplexador binário de 2 saídas

9.4. Grupo teste – Comparações

- EQUAL - Verifica se o valor de entrada é igual ao valor de referência (=)
- NOTEQUAL - Verifica se o valor de entrada é diferente do valor de referência (!=)
- GREATEREQUAL - Verifica se o valor de entrada é maior ou igual a referência (>=)
- GREATER - Verifica se o valor de entrada é maior que a referência (>)
- LOWEREQUAL - Verifica se o valor de entrada é menor ou igual a referência (<=)
- LOWER - Verifica se o valor de entrada é menor que a referência (<)
- IN_BAND - Verifica se o valor de entrada está entre o valor mínimo e máximo (<>)

9.5. Grupo matemáticas

- LATCH_F / LATCH_I – Movimenta valores entre registros
- ADD2/3/4_F / ADD2/3/4_I - Soma os valores na entrada
- SUB2_F / SUB2_I – Subtrai dois valores de entrada
- AND2 - Operação binária AND (E) em duas entradas
- MUL2_F / MUL2_I – Multiplica dois valores de entrada
- NEG_F / NEG_I - Inverte o sinal do valor de entrada
- SIN - Calcula o seno do valor da entrada (em radianos)
- DIV2_F / DIV2_I – Divide os valores de duas entradas
- 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
- COS - Calcula o cosseno do ângulo de entrada (em radianos)
- TAN – Calcula a tangente do ângulo do valor da entrada (em radianos)
- POW_F - Calcula IN elevado a potência EXP
- ABS_F / ABS_I – Calcula o valor absoluto da entrada
- SQRT_F – Calcula a raiz quadrada do valor de entrada
- OR2 - Operação (OR) entre duas palavras de entradas
- XOR2 - Operação XOR binária entre 2 números inteiros

9.6. Grupo numérico

- LIMIT_F / LIMIT_I - Limita o valor de saída
- MAXVAL_F / MAXVAL_I - Registra o maior valor de entrada
- MAX_F / MAX_I - Move o maior valor atual entre 2 entradas
- MEAN2_F / MEAN2_I / MEAN3_F / MEAN3_I - Calcula a média aritmética
- MINVAL_F / MINVAL_I - Registra o menor valor da entrada IN
- MIN_F / MIN_I - Copia para OUT a entrada de menor valor atual
- MUX2_F / MUX2_I - Multiplexador de 2 entradas
- MUX3 / MUX4 / MUX8 / MUX16 - Multiplexador de 3, 4, 8 ou 16 entradas (ponteiro).
- SCALE2 / SCALE2_I - Aplica uma escala + offset à entrada
- SCALE3_F / SCALE3_I - Aplica uma escala com os limites via terminais
- SCALE_F / SCALE_I - Aplica uma escala com os limites via parâmetros
- FILTER_F / FILTER_I - Calcula a média do sinal de entrada
- SENSOR_MEAN - Calcula a média aritmética em até 3 sensores
- DELTA_HIST - Calcula a diferença entre o valor da entrada e um valor de referência

9.7. Grupo temporizadores

- COUNTER - Contabiliza o tempo que a entrada permaneceu ativa (Dec, Ms e Sec)
- DELAY / DELAY_MIN - Gera um atraso na saída a partir da entrada
- MIN_OFF - Saída desligada por um tempo
- MIN_ON - Saída ligada por um tempo
- MIN_ONOFF - Controla os tempos mínimos para cada estado da saída
- OFF_DELAY / OFF_DELAY2 - Gera um atraso (dec) para desligar a saída
- ON_DELAY - OUT liga após IN permanecer acionado durante o tempo “delay”
- TOFF_MS / TOFF_S - Temporizador para desligar
- TON_MS / TON_S - Temporizador para ligar
- OSCILLATOR - Oscilador com tempo liga/desliga configurável (dec)
- PULSE - Saída com pulso de largura fixa
- PWM - Saída digital PWM
- PWM_A - Saída analógica PWM
- UP_DOWN - Contador Up/Down com reset
- PULSE_COUNT - Contador de pulsos com saída retentiva
- HOUR_METER1 / HOUR_METER2 - Horímetro

9.8. Grupo conversão

- TOFLOAT – Converte um número inteiro para real
- TOINT – Conversor de real para inteiro com escala
- 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

MDBM_READ_BIT / MDBM_READ_INT / MDBM_READ_FLOAT
Permite a leitura de dados (Bit, Float, Int) em um equipamento escravo da rede

MDBM_WRITE_BIT / MDBM_WRITE_INT / MDBM_WRITE_FLOAT
Permite a escrita de dados (Bit, Float, Int) em um equipamento escravo da rede

MDBM_READ_BIT_MULTIPLE/ MDBM_READ_INT_MULTIPLE/
MDBM_READ_FLOAT_MULTIPLE
Permite a leitura na sequência de até 8 dados em um equipamento escravo da rede.

MDBM_WRITE_BIT_MULTIPLE/ MDBM_WRITE_INT_MULTIPLE/
MDBM_WRITE_FLOAT_MULTIPLE
Permite a escrita na sequência de até 8 dados em um equipamento escravo.

9.10. Grupo diversos

NV_FLOAT - Cria uma variável *Float* retentiva
SCHEDULE_OFFSET – Programação horária interna

9.11. Grupo HVAC

ABS_HUMITY – Calcula a umidade absoluta do ar (g/kg)
ENTHALPY – Cálculo da entalpia do ar
PRESS_TO_FLOW – Conversor de Pressão (PA) para vazão (m³/h)

9.12. Grupo sistema

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
RAMP – Rampa com tempo de transferência
ANALOG_STAGING4 - Rodízio de até 4 saídas analógicas com sinal individual
FLOAT_VALVE - Controla a posição de uma válvula *Float* de duas saídas digitais
INTERLOCK - Intertravamento de saídas
ONOFF_C - Controle ON/OFF para refrigeração
ONOFF_H - Controle ON/OFF para aquecimento
ONOFF_MULT4 - Controle ON/OFF até 4 estágio para refrigeração e aquecimento
ONOFF_MULT_C - Controle ON-OFF multi estágio para refrigeração
PID1B_F / PID1_F - Controle PID clássico
PID2B_F / PID2_F – Controle PI ISA
PRIORITY_LOCK - Intertravamento de saídas (até 32) com prioridade
SEQUENCE – Controle de saídas com sequência fixa e quantidade especificada
STAGING8 - Rodízio para até 8 saídas digitais por falha ou por tempo de operação
STATUS – Monitora o estado de uma saída (feedback) por uma entrada digital

10. Exemplos de aplicações

- 10.1. Função Modbus - Utilizando a IHM Weintek com o TCP46
- 10.2. TCP46 e módulo Tecnolog AM8DI ou AM8RO
- 10.3. TCP46 e módulo Tecnolog AM8C (entrada analógica)
- 10.4. IHM Weintek, TCP46 e AM8DI (Modbus TCP + Modbus RTU)
- 10.5. TCP46 e inversor Veichi AC10/AC300 via Modbus RTU
- 10.6. Bloco ENCODE para comparação de faixa de valores
- 10.7. Rampas e patamares com o bloco RAMP
- 10.8. Rede para acesso remoto VPN em IHMs Weintek
- 10.9. IHM Veichi Vi20 e TCP46

Apêndice A – Programação de sequências (Grafcet)



Sequências

Estruturas lógicas

Paralelismo e sincronidade

Máquina de Estados de Moore/Sequência de operações

Máquina de Estados em Série

Máquina de estados em modo passo a passo

Função sub-rotina

1. Componentes do sistema

1.1. Descrição da unidade básica e expansão

Unidade básica: TCP46

- Software de programação gratuito (TProg)
- Linguagem de programação por blocos de funções
- Configuração através de *WebServer* interno via *Browser*
- Monitoração *Online* da programação
- Protocolos de comunicação: Modbus RTU e Modbus TCP
- Permite até 8 módulos de expansão TCP46EX (até 414 pontos de IO)
- 18 entradas digitais PNP (24 Vcc), sendo 4 entradas rápidas
- 2 entradas rápidas de 6,8kHz (DI1 e DI4) e 2 entradas rápidas de 400 Hz (DI2 e DI3)
- 16 saídas digitais a relé para cargas até 250V e 2A
- 4 entradas analógicas (10 Bits) de 0/2 ~ 10V ou 0/4 ~ 20mA
- 4 entradas para sensor de temperatura NTC configurável para contato seco ou NPN
- 4 saídas analógicas (11 Bits) de 0/2 ~ 10V ou 0/4 ~ 20mA
- 1 Interface Ethernet 10/100M
- 1 Interface RS-485 com isolamento galvânica (principal)
- 1 Interface RS-485 sem isolamento galvânica (auxiliar)
- Relógio interno (RTC) mantido por bateria
- Processador de 32 Bits (72MHz)
- Memória para armazenamento: Flash 512Kb e RAM 64Kb
- Operações matemáticas com ponto flutuante
- Programação e dados mantidos por bateria
- Blocos de funções para: controle PID, máquina de estados, rampas e patamares, PWM, refrigeração, aquecimento, umidificação, desumidificação, controle de CO2, entalpia, ciclo economizador, controle de pressão, rodízio, agenda de horários e monitoramento e controle de circuitos elétricos.
- Alimentação de 90 ~ 240 Vca, 24 Vca ou 125 Vcc
- Permite fixação em trilho DIN

Expansão: TCP46EX

- Programação via ferramenta TProg;
- Processador 32 Bits;
- 18 Entradas digitais (máx. 20hz);
- 16 Saídas digitais;
- 4 Entradas analógicas;
- 4 Saídas analógicas;
- 4 Entradas NTC configurável para contato seco ou NPN
- 1 portas RS485 Isolada;
- Protocolos de comunicação: Modbus RTU e Modbus TCP.
- Alimentação 90-240VAC;
- Fixação trilho DIN;

1.2. Características técnicas e especificações ambientais

Alimentação	90 a 240V _{AC} (50/60Hz) ou 125V _{cc} . Utilizar fusível de 300 mA para a proteção.
Consumo	30 VA máximo.
Saídas digitais	Relés. Carga máxima 2A @ 250V _{AC} . Proteção para cargas indutivas através do varistor 250V _{AC} interno. Precisão de 2%
Saídas analógicas	Modo tensão: mínima impedância de carga 100k ohms. Modo corrente: máxima impedância de carga 500 ohms. Velocidade de escrita de 50 ms e resolução 11 bits (0 ~ 2048). Precisão de 2%
Fonte interna	24V _{cc} / 200mA para alimentação de sensores.
Entradas digitais	24V _{dc} (PNP), consumo de corrente ~ 200uA. Utilize a fonte interna do CLP para chavear as entradas digitais.
Entradas Temperatura	Sensor NTC de 10k, com curva tipo II ou tipo III (-50 a 150° C)
Entradas Analógicas	Configuráveis: 0-20mA, 4-20mA, 0-10V e 2-10V. Impedância ~100k ohms (modo tensão) e 500 ohms (modo corrente). Alimentação para sensores externos 24V com corrente máxima total de 160mA. Velocidade de leitura: 33 ms e resolução 10 bits (0 ~ 1024).
Comunicação Ethernet	10/100Mbps, sem polaridade de cabo (Auto MDI/MDI-X). Protocolos suportados: ARP, IPv4, UDP, TCP, HTTP, DHCP, Modbus/TCP, /IP, Ethernet.
Comunicação RS485	EIA-485 isolada com fonte interna. Isolação 1500V. Máxima 115200 bps.
Temperatura de operação	0 a 60 °C. Umidade máxima de 95% não condensável.
Massa	520 gramas.
Fixação	Trilho DIN 35mm.
Dimensões externas	212 x 90 x 63 mm.

1.3. Terminais da unidade básica e de expansão

VAC (Ground/A/B): Aterramento (Ground) e alimentação (A e B) de 90~240VAC;

DO1 ~ DO16: 16 saídas digitais a relé;

D+ / COM / D-: RS485 principal (inferior);

DI1 ~ DI18: 18 entradas digitais **PNP**;

N1 ~ N4: 4 entradas configurável para NTC, contato seco ou **NPN**

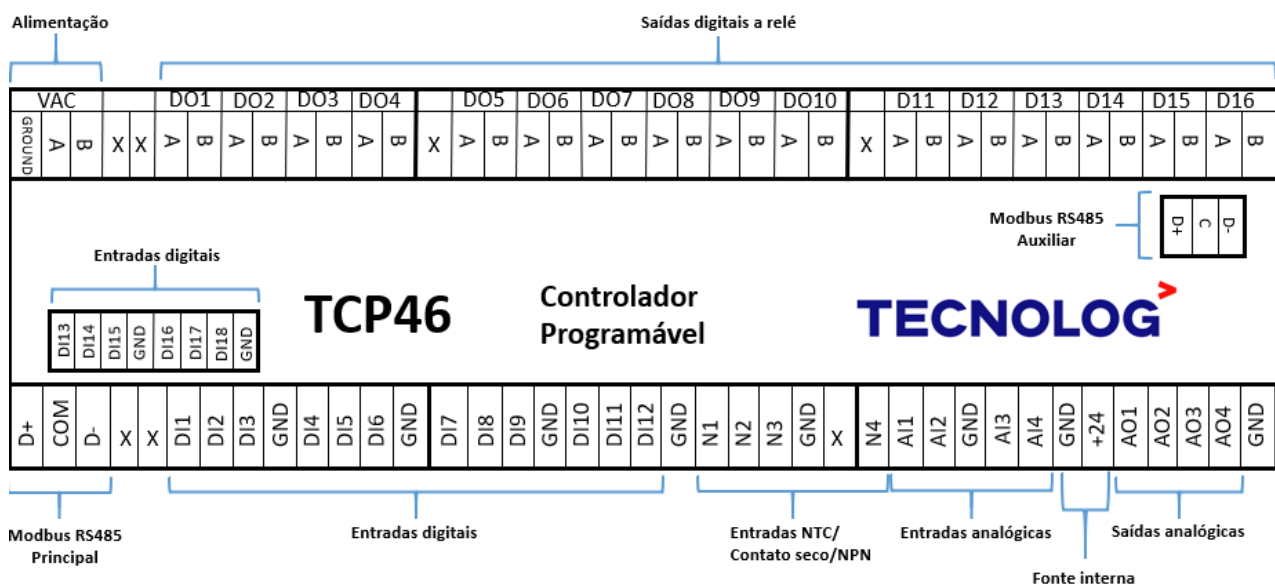
AI1 ~ AI4: 4 entradas analógicas de 10 Bits;

+24 / GND: Fonte interna de 24Vcc/200mA;

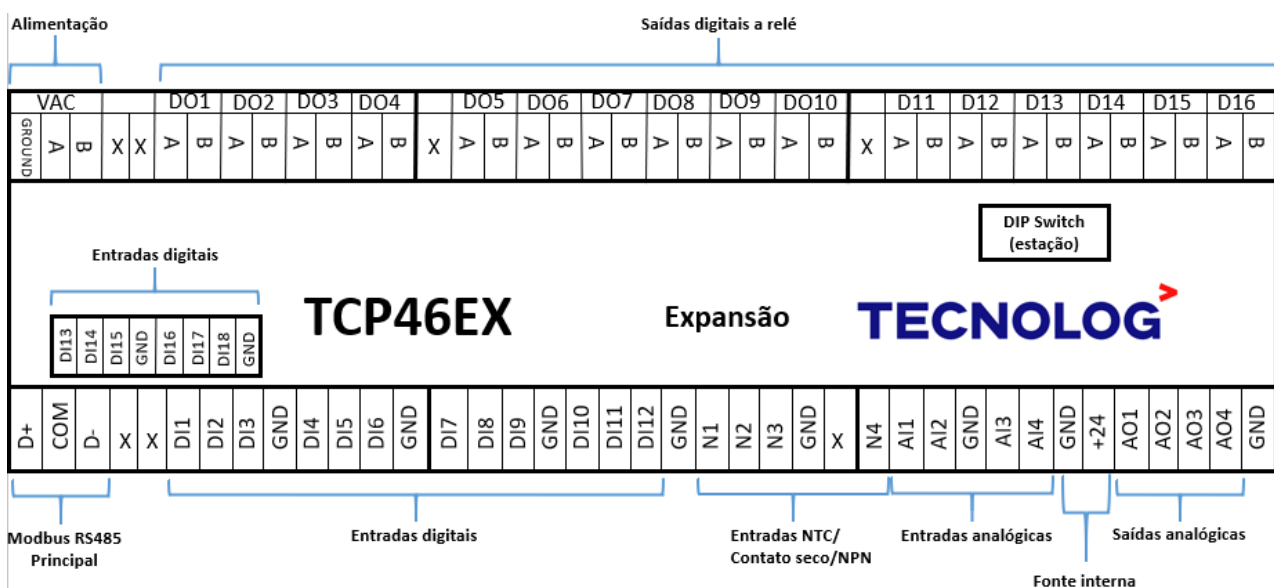
AO1 ~ AO4: 4 saídas analógicas de 11 Bits;

D+ / C / D-: RS485 secundária (superior).

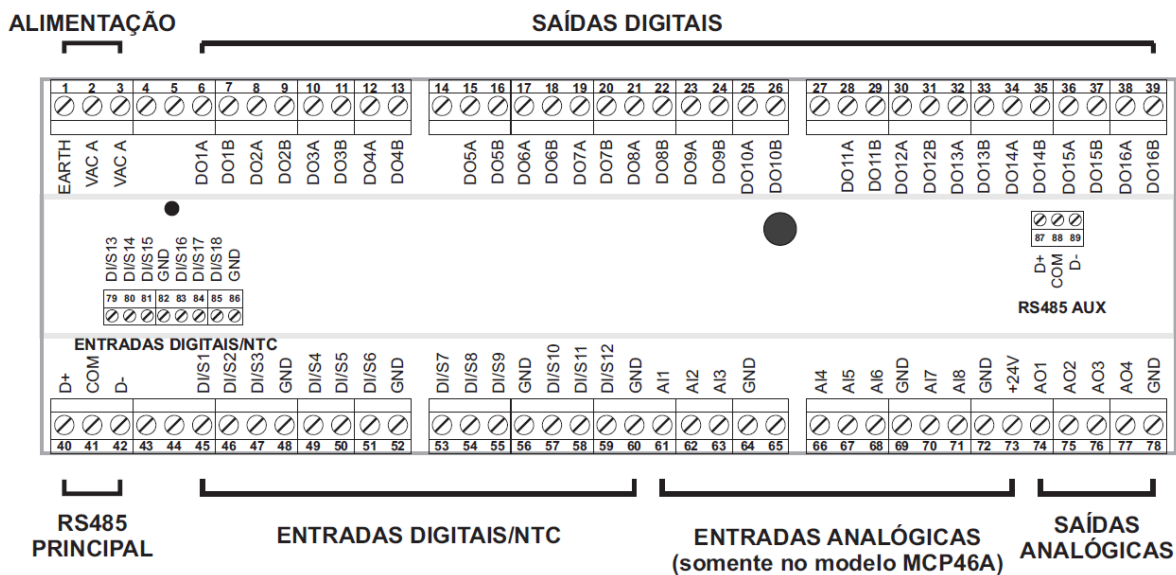
Terminais da unidade básica: TCP46



Terminais da unidade de expansão: TCP46EX



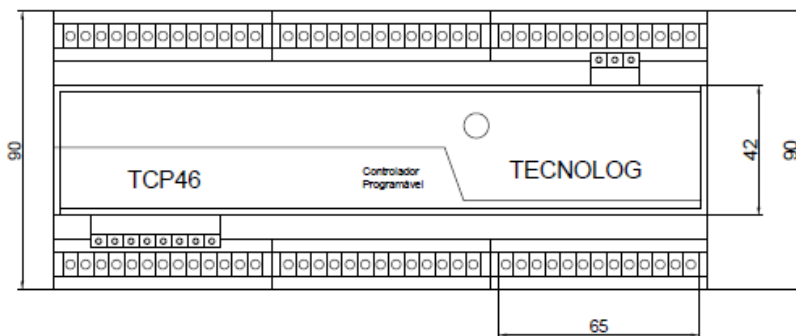
Terminais da unidade básica: TCP46A



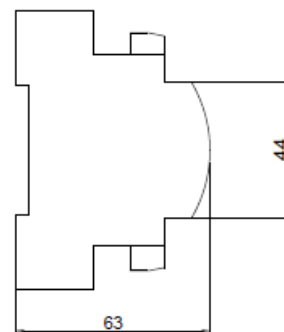
1.4. Dimensões

Dimensões do TCP46, TCP46A e TCP46EX: 212mm x 90mm x 63mm

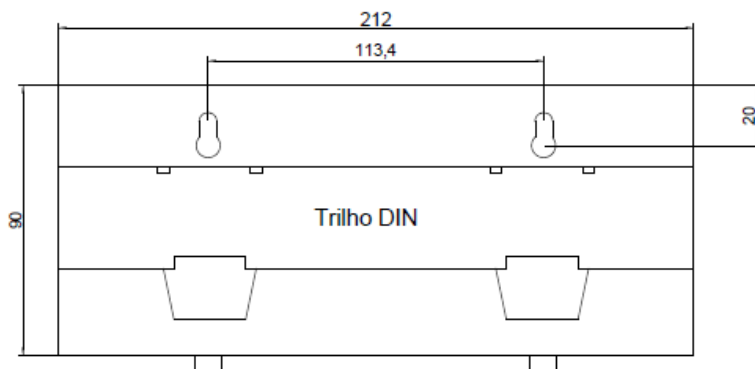
Vista superior



Vista lateral



Vista inferior



1.5. Instalação e manutenção

As recomendações abaixo devem ser seguidas para aumentar a confiabilidade do sistema e durabilidade do controlador frente as condições de trabalho.

Ambiente de instalação:

- O controlador deve ser instalado em um local com temperatura ambiente entre 0 a 55°C e umidade relativa entre 5 a 95%.
- Instale em um local que sofra mínimas vibrações.
- Evite a incidência direta da luz solar.
- Evite o contato com gases corrosivos e/ou inflamáveis.

Instruções de montagem:

- Monte o produto em um trilho DIN na posição horizontal.
- Coloque as canaletas para a passagem dos fios e outros equipamentos elétricos afastados no mínimo 100mm do controlador, isso melhora o fluxo de ar e a influência de ruídos eletromagnéticos.
- Não obstrua o acesso a trava inferior do CLP.
- Confirme o aperto em cada terminal para evitar um acidente.

Fiação recomendada

Utilize na alimentação do controlador fios com bitola de no mínimo 1,5mm² (15 AWG) e para as IO utilize bitolas entre 0,5 a 1,5mm² (20 a 15 AWG).

Use terminais tipo Garfo com compressão (crimpado) em todos os pontos para evitar mal contato ou curto-circuito entre os fios.

Devem ser utilizados cabos blindados para a comunicação Modbus RTU pois atenuam a interferências eletromagnéticas.

Aterramento

Conecte cada equipamento individualmente ao barramento terra utilizando fio (verde e/ou amarelo) com bitola superior a 1,5mm² (15 AWG).

Proteção contra sobrecarga

Utilize um fusível de 300mA na rede de alimentação, caso ocorra uma sobrecarga permanente (> 280Vca) esse componente evitará a queima da fonte.

Observação: caso o fusível queime novamente ou frequentemente, verifique cada ponto e os terminais para descobrir o que está danificando o sistema.

Proteção contra raios

Em caso de maior incidência de descargas atmosféricas, utilize o dispositivo de proteção contra surtos (DPS) adequado para a proteção do controlador e outros componentes elétrico/eletrônico.

Organização dos cabos

Diferencie o trajeto dos cabos de força dos cabos de controle, evitando assim interferências eletromagnéticas gerados pelos cabos de força.

Backup do arquivo do programa

É importante manter cópias em um local seguro (nuvem e no PC) do arquivo fonte da programação feita no CLP, pois **não é possível realizar o Upload do arquivo** (CLP > PC) caso alguma alteração seja necessária realizar no programa e não possua o atual arquivo de programa.

2. Conexão das entradas/saídas digitais e analógicas

Importante: Antes de iniciar a conexão elétrica ou acionar as entradas digitais, **verifique o modelo do CLP** (confirmando pela etiqueta lateral e a nomenclatura dos terminais) para reproduzir a correta ligação elétrica, visto que possui o modelo TCP46 (DIx = PNP) e o TCP46A (DI/Sx = NPN).

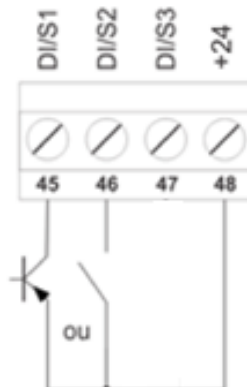
2.1. Ligações da unidade básica TCP46 e expansão TCP46EX

2.1.1. Entradas digitais

Os controladores possuem 18 entradas digitais 24Vcc com o tempo de atualização de **8,3 ms** (leitura até 120Hz).

As 4 primeiras entradas (DI1 ~ DI4) são entradas rápidas, sendo DI1 e DI4 até **6,8kHz** e DI2 e DI3 até **400Hz** e o bloco "**Pulse_Freq**" deve ser usado para a leitura dos pulsos nos canais rápidos.

As entradas digitais são do tipo PNP e devem ser acionadas pelo terminal +24 da fonte de tensão interna de 24Vdc.



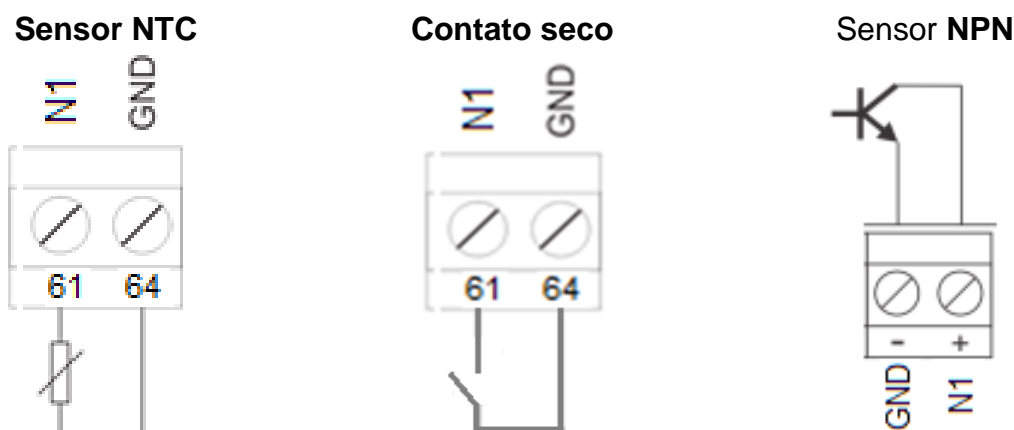
2.1.2. Entradas NTC, contato seco ou NPN

O TCP46 possui 4 entradas para a leitura de sensores NTC de 10k tipo II/III, contato seco ou sensor NPN (N1 ~ N4).

A definição por entrada é feita no TProg (árvore do projeto) > configurações > entradas > tipo de entrada analógica > N1 ~ N4. Maiores detalhes serão tratados posteriormente.

Importante: utilize fios de cores diferentes para identificar as entradas Nx das demais entradas Dlx, pois alimentá-las com sinal +24V irá queimá-las.

Abaixo estão apresentadas as possíveis conexões das entradas Nx:

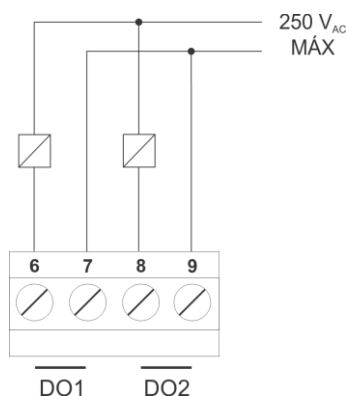


2.1.3. Saídas digitais

Os controladores possuem 16 saídas digitais a relé (DO1~DO16) para o acionamento de cargas diversas.

As saídas possuem proteção interna para cargas indutivas (varistores) e são projetadas para o acionamento de **cargas até 250V_{AC} e 2A**.

A carga deve ser ligada conforme mostrado abaixo:



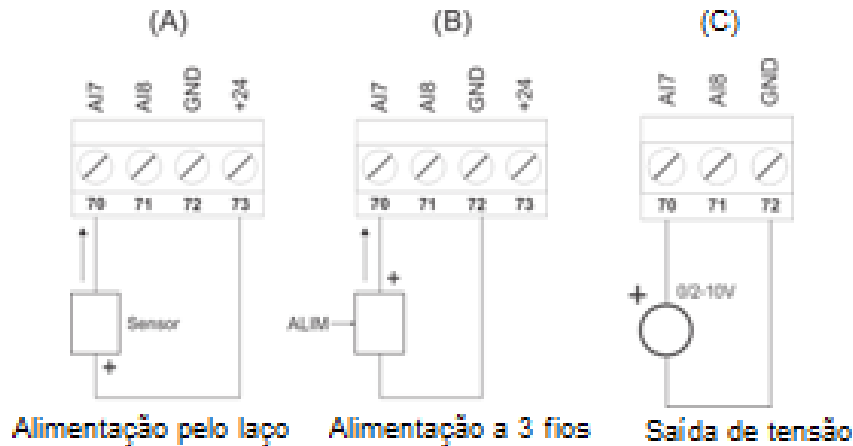
As saídas devem ser protegidas com fusíveis, principalmente quando alimentadas em 110/220Vca. Quando alimentadas em 24Vcc, a proteção contra curto-circuito será feita pela própria fonte, mas em fontes com saída **superior a 5A**, também recomenda-se o uso de fusíveis.

2.1.4. Entradas analógicas

O TCP46 possui 4 entradas físicas para leitura de: **0/4~20mA** ou **0/2~10V**.

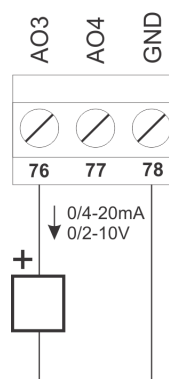
Para leitura de sinais de sensores externos o CLP possui uma fonte interna de 24Vcc/200mA (**pino 73**) para fornecer alimentação aos transmissores pelo laço de corrente (**A**).

Em casos onde a alimentação é a **3 fios**, a ligação deve ser feita conforme (**B**) e quando usamos sensores com saída de tensão, a ligação deve ser conforme (**C**). O tempo de leitura da entrada analógica é de **33ms**.



2.1.5. Saídas analógicas

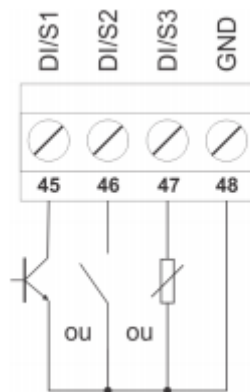
As 4 saídas analógicas são configuráveis para tensão (0/2~10V) ou corrente (0/4~20mA). A forma correta para a ligação de uma carga na saída analógica está apresentada abaixo, sendo essa, a ligação em ambos os modos de operação (tensão ou corrente). O tempo de resposta da saída analógica é de **50ms**.



2.2. Ligações da unidade básica: TCP46A

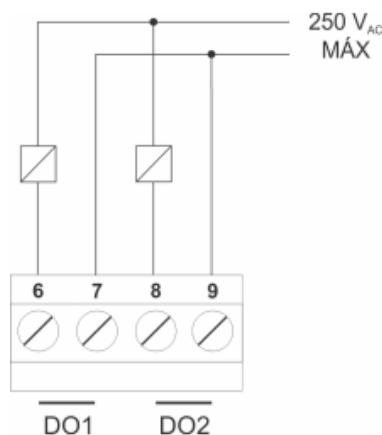
2.2.1. Entradas digitais / NTC

O TCP46A possui 18 entradas digitais (contato seco) ou para sensores de temperatura NTC (10k, curvas AN e CP). As entradas digitais podem ser utilizadas para monitoração do estado de um contato. **Não** pode ser aplicado nenhum potencial na entrada, com risco de danificar o equipamento.



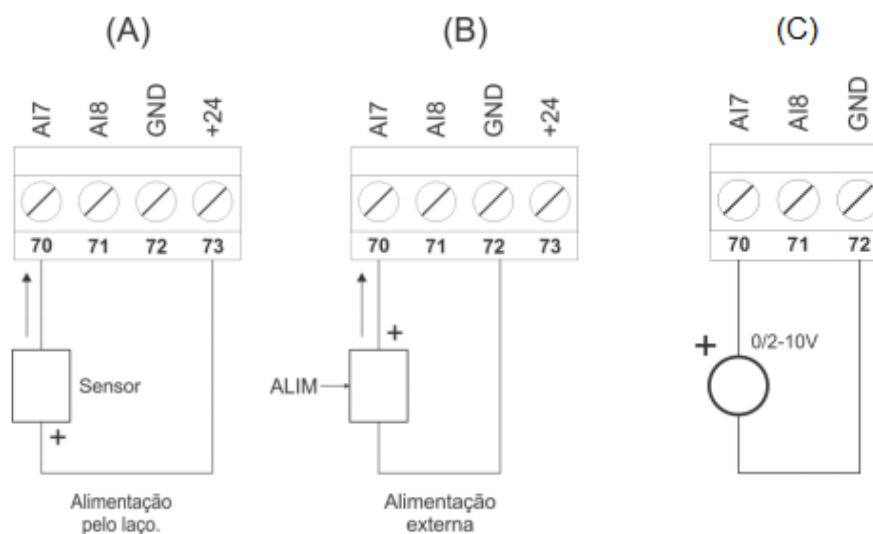
2.2.2. Saídas digitais

O TCP46A possui 16 saídas digitais a relé que podem ser usadas para acionamento de cargas diversas. As saídas foram projetadas para acionamento de cargas de até 250VAC em 2A. Possuem proteção interna para cargas indutivas (varistores).



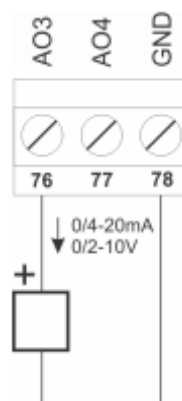
2.2.3. Entradas analógicas

As 8 entradas analógicas (AI1~AI8) do TCP46A aceitam sensores externos com saída em 0~20mA, 4~20mA, 0~10V ou 2~10V. O módulo pode fornecer uma alimentação de 24VDC para o laço de corrente. Nos casos em que esta alimentação é utilizada, a ligação deve ser feita conforma a figura A. Para casos onde a alimentação é externa, a ligação deve ser feita como indicado na figura B. Para sensores com saída tensão, o sensor deve ser ligado conforme a figura C.



2.2.4. Saídas analógicas

O TCP46A possui 4 saídas analógicas que podem ser configuradas no modo tensão (0/2~10V) e no modo corrente (0/4~20mA). A ligação abaixo é semelhante nos 2 modos de operação (tensão ou corrente).



3. Interfaces de comunicação

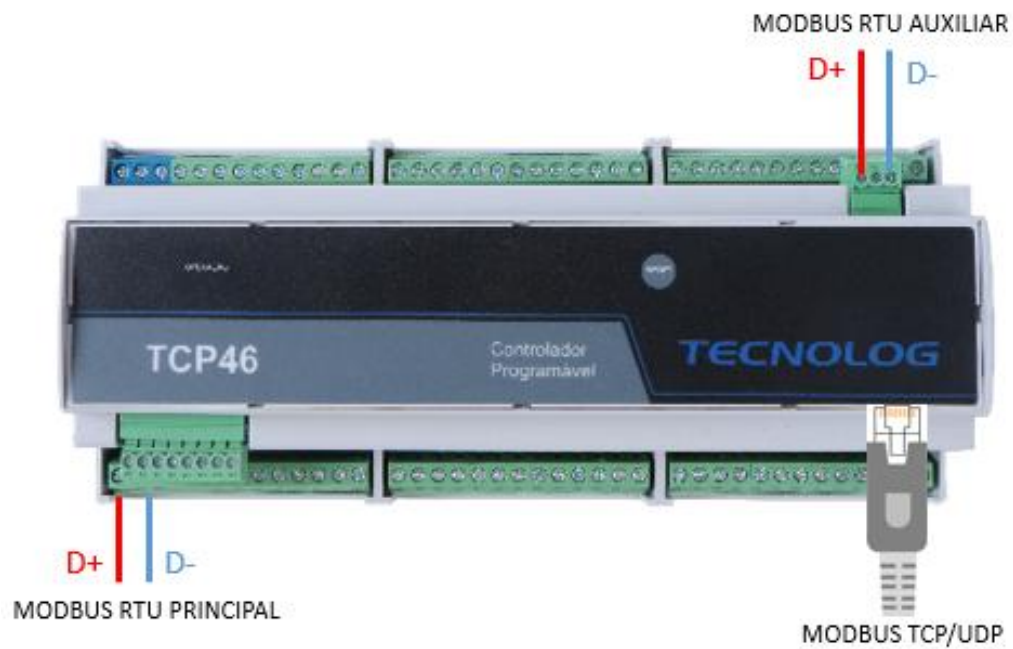
3.1. Sistemas de comunicação disponíveis

- Interfaces Seriais:

2x Modbus RTU (RS485), sendo 1 principal (inferior) e 1 auxiliar (superior).

- Interface Ethernet:

Modbus TCP/UDP: 1 porta (RJ45)

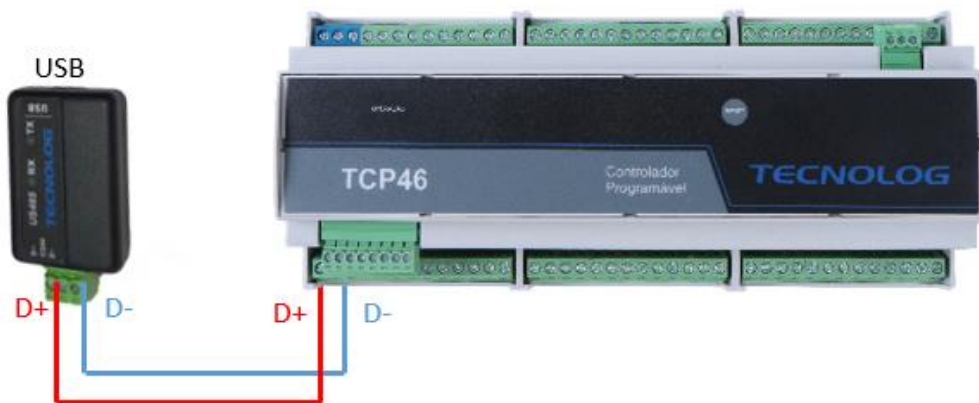


3.2. Interfaces de comunicação

O TCP46 possui três portas de comunicação, sendo uma porta ethernet (conector RJ45) para os protocolos Modbus TCP ou Modbus UDP e duas portas seriais para o protocolo Modbus RTU (meio físico RS485). As portas podem ser utilizadas para programação e comunicação entre dispositivos e com as expansões.

3.2.1. Protocolo Modbus RTU

Programação: utilize um conversor USB para serial RS485 (**US485 da Tecnologia**). Os fios são conectados ponto a ponto, conforme mostrado abaixo:

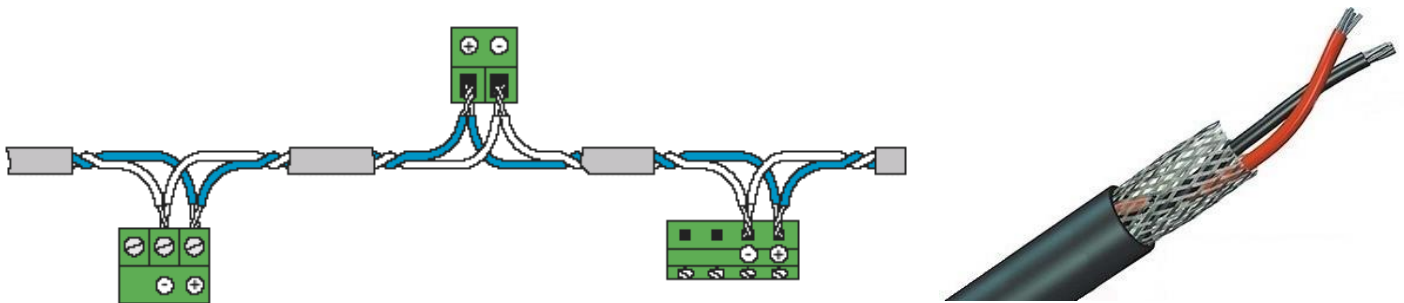


Comunicação: São permitidos até 32 escravos por canal RS485 e a topologia de rede para a conexão entre os dispositivos é a **Daisy chain** (foto abaixo).

Para reduzir as interferências eletromagnéticas geradas por outros periféricos, utilize um cabo com blindagem do tipo par trançado (foto abaixo).

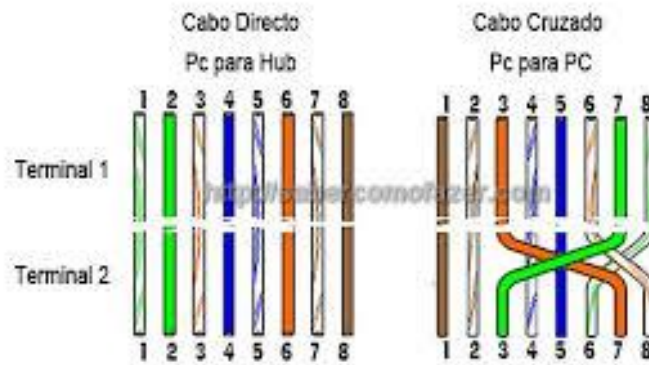
Como regra geral, quanto maior a rede (comprimento máximo de 1 km) menor deve ser a velocidade (taxa de transmissão em kbps).

Utilize resistores de 120 Ω nas duas extremidades em caso de rede acima de 100 m.



3.2.2. Protocolo Modbus TCP/UDP

Cabo de programação e comunicação: podem ser utilizados cabos de rede **direto** ou **cruzado** (*crossover*) com terminais RJ45.



Em uma rede Ethernet, os componentes devem ser ajustados para estar na **mesma faixa de IP**, mas com IPs diferentes.

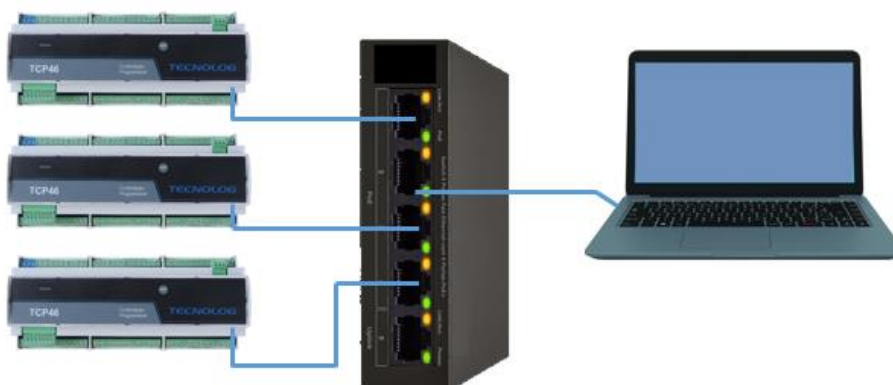
Ex: PC (192.168.1.16) - TCP46 1 (192.168.1.77) – TCP46 2 (192.168.1.78) - ...

Exemplos de conexão:

- Conexão ponto a ponto:



- Utilizando um switch:



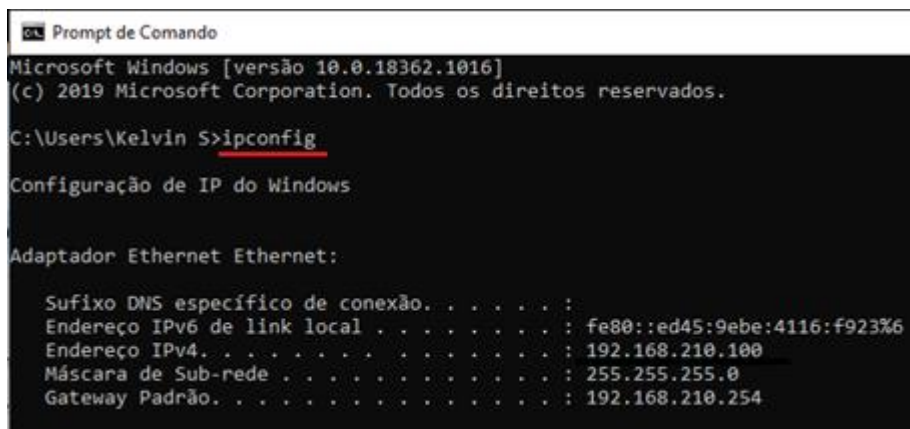
4. Configurações iniciais do TCP46 pelo WebServer

Todas as configurações básicas do TCP46 são feitas em um *browser* comum, através do WebServer interno do equipamento, sendo que a primeira conexão deve ser feita conectando diretamente o PC ao TCP46 sem utilizar a rede local.

4.1. Acessando o WebServer pela primeira vez

Antes de iniciar o procedimento de configuração da placa de rede do PC, recomendamos anotar os parâmetros atuais (Endereço IPv4, Máscara de Sub-rede e Gateway Padrão) para posterior reconfiguração.

Acesse: Iniciar > digite: "cmd" e pressione enter > digite: "ipconfig".



```

Prompt de Comando
Microsoft Windows [versão 10.0.18362.1016]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Kelvin S>ipconfig


Configuração de IP do Windows

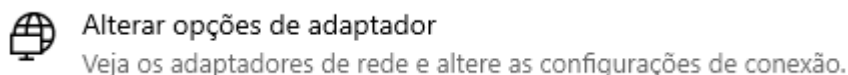
Adaptador Ethernet Ethernet:

    Sufixo DNS específico de conexão. . . . . :
    Endereço IPv6 de link local . . . . . : fe80::ed45:9ebe:4116:f923%6
    Endereço IPv4. . . . . : 192.168.210.100
    Máscara de Sub-rede . . . . . : 255.255.255.0
    Gateway Padrão. . . . . : 192.168.210.254
  
```

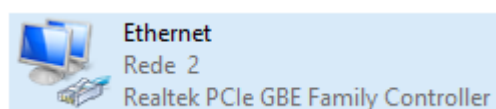
- Ligação direta entre PC e TCP46 usando um cabo de rede direto ou cruzado:

1. Conecte o cabo de rede entre o PC e TCP46
2. Na **primeira** comunicação entre TCP46 e PC é necessário alterar os dados da placa de rede do PC para que fique na mesma faixa de rede padrão do TCP46 (IP: 10.1.1.240).

Procedimento: clique com o botão auxiliar do mouse sobre o ícone  (placa de rede) > Abrir configurações de rede e internet > Alterar opções de adaptador



Clique duplo sobre Ethernet



Propriedades > Protocolo IP versão 4 (TCP/IPv4) > Marque “Usar o seguinte endereço de IP” > altere o atual IP para 10.1.1.xxx (Ex: **10.1.1.199**) > Máscara: 255.55.255.0.

Obter um endereço IP automaticamente
 Usar o seguinte endereço IP:

Endereço IP:	10 . 1 . 1 . 199
Máscara de sub-rede:	255 . 255 . 255 . 0
Gateway padrão:	. . .

3. Feito isso, já é possível acessar o WebServer do equipamento através de um navegador de internet no endereço de IP padrão do TCP46: **10.1.1.240**.
4. Para acessar as configurações do TCP46, utilize: **usuário = config** e a **senha padrão = config**.

- Alterando o IP padrão do TCP46 para permitir a conexão através da rede local:

No Webserver acesse: Configurações > Ethernet > defina os dados de rede do TCP46 deixando-o na mesma faixa do IP original da rede local (Ex IP TCP46: **192.168.210.77**) juntamente com a Máscara de rede e Gateway originais anotados anteriormente.


Configurações - Comunicação

Endereço IP:	192.168.210.77
Máscara de rede:	255.255.255.0
Gateway:	192.168.210.254

Importante: utilize um endereço de IP livre na rede local para evitar conflito. Para verificar acesse: Iniciar (Windows) > cmd e digite “ping 192.168.210.77” e verifique se a resposta é “Host de destino inacessível”.

Retorne o cabo de rede do PC para o switch e utilize outro cabo de rede para conectar o TCP46 ao switch.

- Restaurando as configurações da placa de rede do PC:

Agora com o PC conectado à rede, clique novamente com o botão auxiliar do mouse sobre o ícone  (placa de rede) > Abrir configurações de rede e internet > Alterar opções de adaptador > Clique duplo sobre Ethernet > Propriedades > Protocolo IP versão 4 (TCP/IPv4) > marque a opção “Obter um endereço de IP automaticamente” se anteriormente estava assim, caso contrário marque “Usar o seguinte endereço de IP” e defina os dados anteriores > click OK

- Acessando o WebServer do TCP46 pela rede local:

Abra um navegador (browser) e digite o novo IP do CLP (<http://192.168.210.77>)

A partir desse ponto é possível acessar o WebServer e comunicar com a ferramenta de programação (TProg) pela rede local.

- Retornar o IP do TCP46 para o padrão de fábrica:

Caso seja necessário restabelecer o IP padrão do equipamento (10.1.1.240), pressione a tecla RESET (frontal) durante 10 segundos (com o equipamento ligado). O led de operação (frontal) pisca rapidamente para indicar que o processo foi realizado com sucesso.

Obs: realizando esse procedimento, todos os dados de rede configurado no processo anterior serão perdidos.

4.2. Recursos permitidos no WebServer:

Aba Monitoração:

- **E/S Digitais:** Visualizar o status das entradas e saídas digitais.

Monitoração - E/S Digitais																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Entradas:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Saídas:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- **Entradas analógicas/ Saídas analógicas:** valor percentual (relacionado a grandeza selecionada) das entradas e saídas analógicas.

Monitoração - Entradas Analógicas			Monitoração - Saídas Analógicas	
Entrada	Analógicas	Saída	Valor	
1	0.0 °C	1	50.0 %	
2	0.0 °C	2	30.0 %	
3	0.0 °C	3	20.0 %	
4	0.0 °C	4	80.0 %	

- **Equipamento:** data e hora do equipamento.

Monitoração - Equipamento	
Data	05/11/2020
Hora	14:37:15

Aba Ajustes:

- **Relógio:** Possibilita alterar a data e hora do equipamento.

Ajustes - Relógio	
Data/Hora	_____
Usar data/hora do PC:	<input checked="" type="checkbox"/>
Data:	<input type="text" value="05/11/2020"/>
Hora:	<input type="text" value="14:37:51"/>

- **Comandos:** retorna à configuração padrão ou reinicia o equipamento.

Aba **Configuração:**

- **Comunicação:** Acessa os parâmetros de comunicação dos protocolos Modbus TCP (Ethernet), Modbus Escravo e Modbus Mestre (Modbus RTU) e permite habilitar o uso das expansões em uma porta serial.

Importante 1: Sempre que um parâmetro de comunicação for alterado, o CLP deve ser necessariamente reiniciado (desligar, aguardar 5 segundos e ligar novamente o CLP) para que as **alterações sejam validadas**.

Importante 2: Defina a porta serial (principal ou auxiliar) **somente uma vez** em uma das opções, pois definir a mesma porta serial em mais de uma opção pode ocasionar falhas de comunicação (Ex: usar a porta principal para Modbus mestre e para as expansões).

Configurações - Comunicação	
Ethernet	
Endereço IP:	192.168.1.77
Mascara de rede:	255.255.255.0
Gateway:	192.168.1.1
Modbus Escravo	
Endereço Modbus:	254
Porta 1	
Porta RS485:	Desabilitado
Velocidade:	9600 bps
Paridade:	Nenhuma
Stop bits:	1 stop
Atraso transmissão:	20 ms
Porta 2	
Porta RS485:	Desabilitado
Velocidade:	9600 bps
Paridade:	Nenhuma
Stop bits:	1 stop
Atraso transmissão:	20 ms
Modbus Mestre	
Geral	
Timeout:	2000 ms
Tentativas:	2
Atraso entre leituras:	50 ms
Modbus RTU	
Porta RS485:	Desabilitado
Velocidade:	19200 bps
Paridade:	Nenhuma
Stop bits:	1 stop
Expansões	
Porta RS485:	Desabilitado

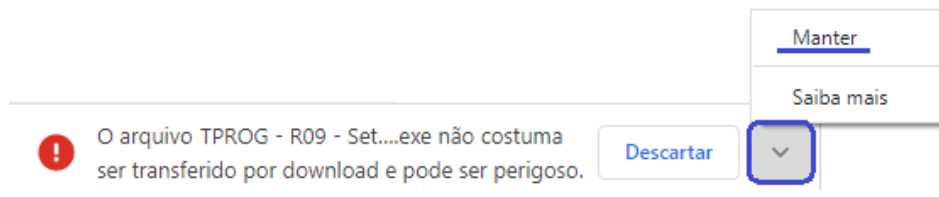
- **Diversos:** Alterar a senha para acessar o WebServer do equipamento.

5. Software TProg

5.1. Instalação do TProg

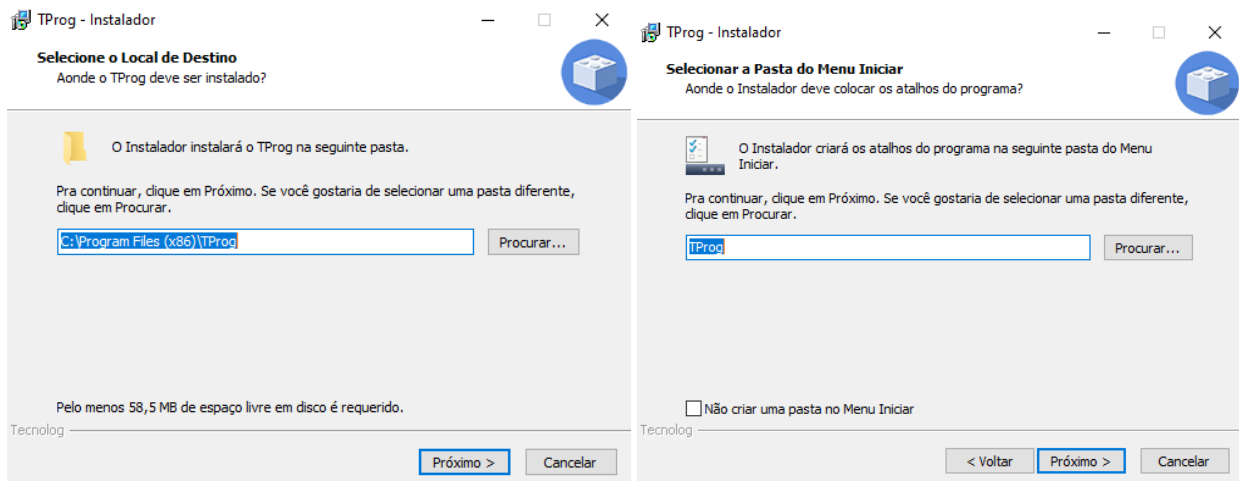
O software **TProg** é a ferramenta de desenvolvimento da programação e pode ser fornecida gratuitamente pela Tecnolog quando solicitada. É possível instalá-lo em sistemas operacionais Windows 7 ou superior.

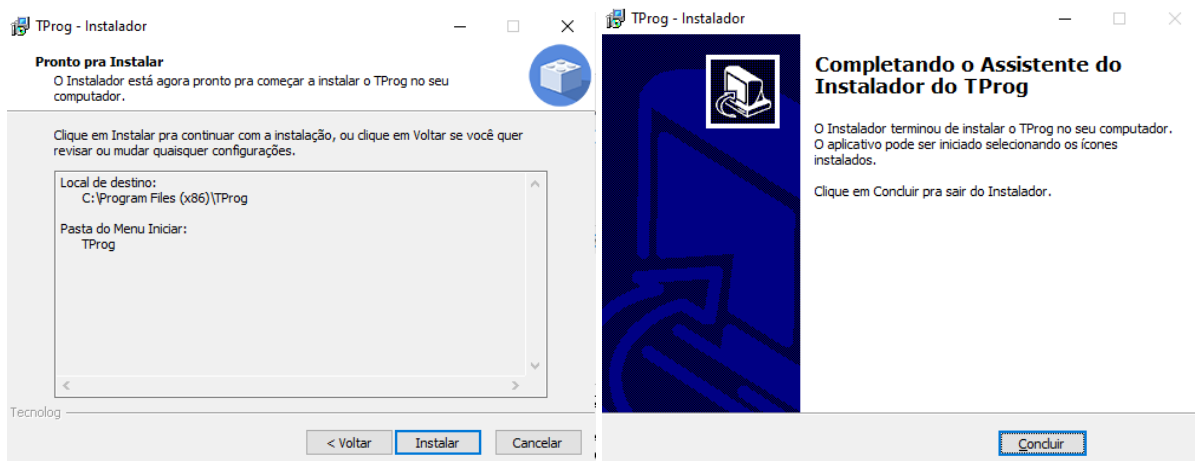
Observação: caso o arquivo de instalação seja bloqueado pelo PC ao ser baixado, clique em “propriedades > manter” para liberá-lo para download.



Instalação:

1. Desative o seu antivírus ou faça a permissão do software Tprog no seu antivírus.
2. Execute o arquivo “Tprog – Setup.exe” e siga os passos do assistente de instalação (fotos) que irá orientá-lo durante o processo de instalação automática.
3. O botão “próximo” inicia e avança o processo e o botão “cancelar” finaliza o processo de instalação.
4. O diretório padrão de instalação é: “C:\Program Files (x86)\TProg”, e para alterá-lo clique em pesquisar e aponte o diretório de instalação desejado.



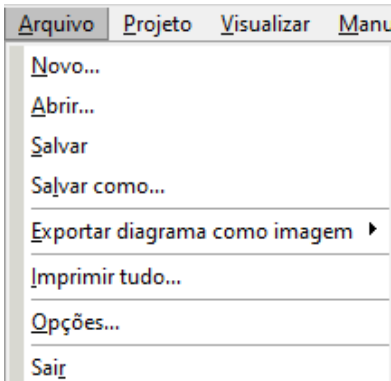


Obs: Após a conclusão da instalação, o ícone para acesso rápido ao software estará na área de trabalho disponível para ser usado.

Atualizar/desatualizar o TProg:

Desinstale a atual versão do TProg (W7: iniciar > adicionar ou remover programas > pesquise "TProg" > desinstalar) antes de instalar o software de uma versão diferente.

5.2. Abas de navegação



Arquivo:

Novo...: Novo projeto

Abrir...: Abrir um arquivo de projeto

Salvar: Salva o projeto atual

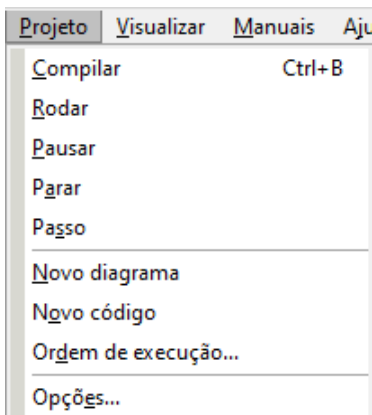
Salvar como...: Salva o projeto atual em um novo arquivo

Exportar diagrama como imagem: (BPM, PNG ou JPG)

Imprimir tudo...: Imprime todos os diagramas do projeto

Opções...: Acessa as opções do projeto

Sair: Fecha o TProg



Projeto:

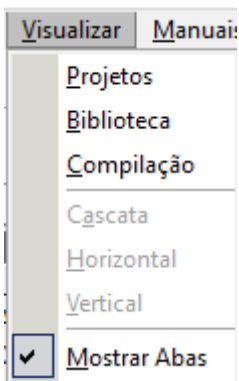
Compilar, Rodar, Pausar, Parar e Passo: Modos de operação

Novo diagrama: Cria um novo diagrama

Novo código: Em breve (linguagem textual – ST)

Ordem de execução...: Ordena os blocos de funções

Opções: Ajustes da interface, conexão e compilação



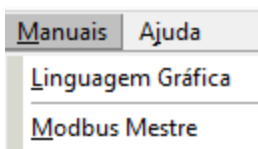
Visualizar:

Projetos: Abre ou fecha a caixa de projetos (**árvore**)

Biblioteca: Abre ou fecha a caixa biblioteca (blocos)

Compilação: Abre ou fecha a caixa de compilação

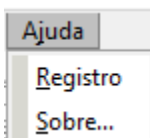
Mostrar abas: Mostrar as abas alinhadas ou expandidas



Manuais:

Linguagem gráfica: Breve descrição sobre a linguagem de blocos.

Modbus Mestre: Descrição detalhada sobre o protocolo e blocos.

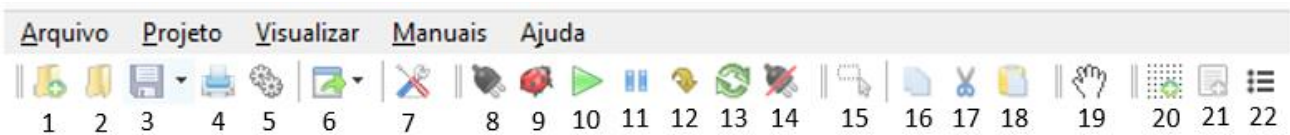


Ajuda:

Registro: Chave de permissão do TProg (licença)

Sobre...: Detalhamento sobre as versões do TProg

5.3. Atalhos do TProg



1. **Novo projeto (Ctrl + N)** > Cria um novo projeto a ser programado
2. **Carregar projeto (Ctrl + O)** > Carrega um projeto existente
3. **Salvar (Ctrl + S)** > Salva o projeto atual e “Flecha para baixo” **Salvar como (Ctrl + Shift + S)** > salva uma cópia do projeto atual
4. **Impressão de diagramas (Ctrl + P)** > Imprime/gera .PDF dos diagramas existentes
5. **Opções do projeto** > Configurações gerais do controlador (modelo, versão das configurações, interface e endereço de programação).
6. **Exportar diagramas para foto** > Formatos disponíveis (BMP, PNG ou JPG)
7. **Compilar (Ctrl + B)** > Executa o compilador para encontrar erros na programação
8. **Conectar ao equipamento e iniciar *debug*** > Conecta ao controlador quando a programação no TProg é igual a programação do CLP (comparador de arquivos)
9. **Iniciar *Debug* (Ctrl + Shift + D)** > Compila e descarrega a programação (download) do TProg para o controlador
10. **Rodar (F5)** > Permite a CPU executar o processamento cíclico do programa
11. **Pausar** > Pausa o processamento do programa temporariamente e mantém os estados das memórias e das E/S.
12. **Passo (F8)** > Ferramenta que realiza apenas 1 scan do CLP a cada acionamento (o CLP deve estar em modo “Pausar”)
13. **Resetar** > Reinicia as memórias e as E/S e altera o modo para Stop.
14. **Desconectar** > Desconecta logicamente o TProg do CLP (usado para editar o projeto)
15. **Selecionar tudo (Ctrl + A)** > Seleciona todos os blocos de função e fios no diagrama presente
16. **Copiar (Ctrl + C)** > Ferramenta de edição para copiar um objeto
17. **Recortar (Ctrl + X)** > Ferramenta de edição para recortar um objeto
18. **Colar (Ctrl + V)** > Ferramenta de edição para colar um objeto
19. **Arrastar (Ctrl + H)** > Permite movimentar o diagrama
20. **Novo diagrama** > Cria um novo diagrama no projeto
21. **Novo código (em breve)**
22. **Ordem de execução dos blocos** > Permite alterar a ordem de execução dos blocos de função.

Outros atalhos:


Aumentar o Zoom: Ctrl + Scroll para cima

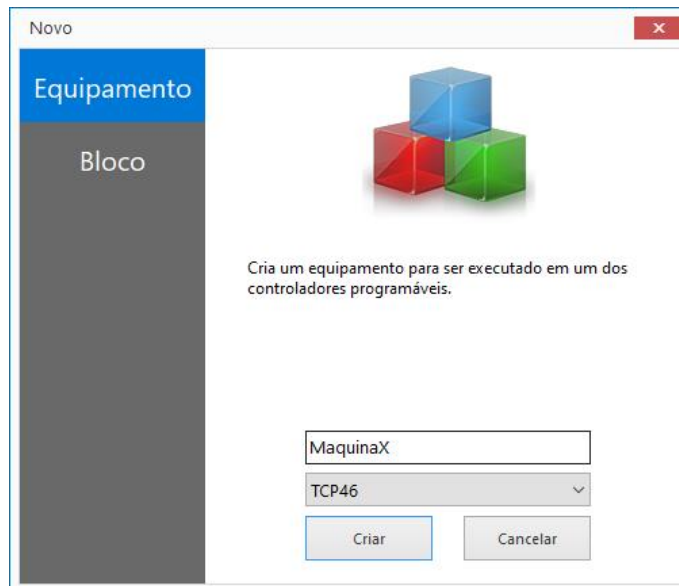
Diminuir o Zoom: Ctrl + Scroll para baixo

Duplicar um objeto: clique com o mouse sobre o objeto e clique “D” (teclado)

Seleciona/deseleciona um objeto: Shift + click sobre objeto

5.4. Iniciando um projeto

Há 3 formas de criar um novo projeto, através do ícone , através das abas “Arquivo > Novo” ou pelo atalho (**Ctrl + N**). Defina o nome do projeto e o modelo do controlador.



5.5. Novo diagrama / Importar diagrama

Os diagramas são áreas disponíveis para criar, dividir e organizar a programação, é possível importar/exportar diagramas para replicar e aproveitar uma programação já feita.

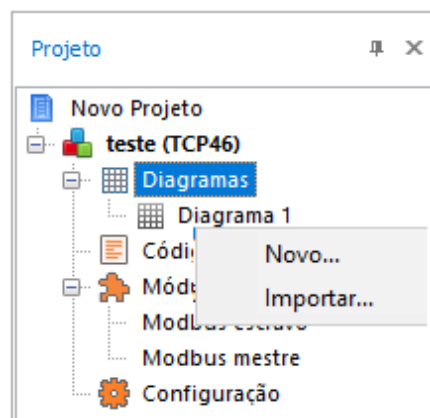
Importante: nomeie os diagramas **sem iniciar com número ou sem utilizar caracteres especiais**, como: parênteses, vírgula, entre outros.

(**Correto:** IOs / **Incorreto:** 01entradas / entradas_1).

Novo diagrama: botão direito do mouse em: “Diagramas” > “novo”

Importar diagramas: botão direito do mouse em: “Diagramas” > “importar”

Todos os diagramas são lidos ao mesmo tempo (em paralelo) e para interligá-los utilize *Labels* (detalhado em 7.3).



Renomeando o projeto: o nome “Novo Projeto” é definido por padrão ao criar um novo projeto e para renomeá-lo clique com o botão auxiliar do mouse > renomear.

Ferramentas



➤ Inserir linhas



➤ Inserir *label* (substitui uma linha)



➤ Inserir caixa de texto (após inserida, F2 para editar o texto e duplo click sobre o bloco para alterar as propriedades da forma e texto).

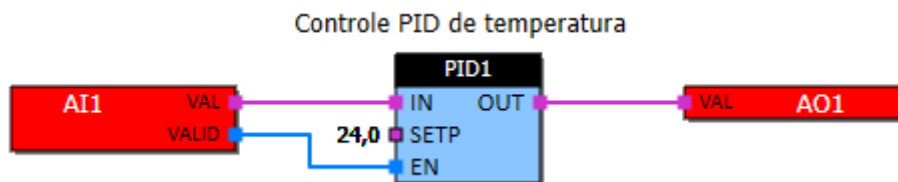


➤ Inserir um retângulo para agrupar visualmente os blocos

6. Dados, operação e execução

6.1. Linguagem de blocos de função

A plataforma utiliza uma linguagem **gráfica**, os blocos de funções são conectados formando um diagrama combinacional que executa a lógica definida pelos blocos.




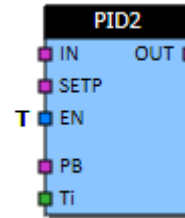
A lógica é continuamente atualizada, gerando novos valores para as saídas de cada bloco a cada ciclo de execução do CLP. **Um ciclo representa a atualização de todos os blocos do projeto.** A velocidade de execução depende do tamanho das lógicas programadas juntamente com a capacidade de processamento do controlador.

Em modo depuração, o sistema informa a velocidade média de execução (**ciclos/s**), localizado na parte inferior direita do TProg (descrita em 7.10).

6.2. Tipos de dados

A linguagem de blocos possui suporte para 3 tipos de dados: **booleano**, **inteiro (I16 ou I32)** e **real** (ponto flutuante). O tipo de dado é representado pela cor do pino abaixo.


BOOLEANO (Dado: Bit e Pino: azul)
INTEIRO (Dado: Int16 ou Int32 e Pino: verde)
REAL (Dado: Float e Pino: roxo)



Faixa de valores por registro:

Bool: entrada 0 ou 1

I16: valores de -32.768 a 32.767

I32: valores de -2.147.483.648 a 2.147.483.647

Float: valores de -99999999,9 a 99999999,9

Obs: em monitoração, os valores no TProg são apresentados com 1 ou 2 casas após a vírgula, porém na memória/endereço o valor está completo.

6.3. Interligar blocos de função

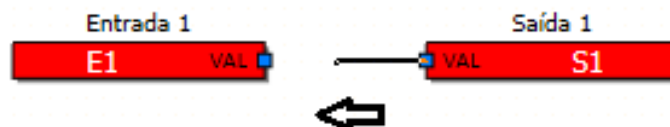
Somente é possível conectar os blocos que tiveram o **mesmo formato de dado** (mesma cor do pino). **Ex:** azul > azul (bool), verde > verde (int) e roxo > roxo (float).

Ex prático: Conectar um bloco de entrada digital BI (bool) a um bloco de saída digital BO.

Insira um bloco de função BI com a entrada 1 (E1).

Insira um bloco de função BO com a saída 1 (S1).

Interligue os blocos clicando no terminal azul do bloco da direita e mantendo pressionado até o terminal azul da esquerda, o cursor irá trocar da seta para uma mão, indicando que o local está conectado e que pode soltar o botão do mouse.



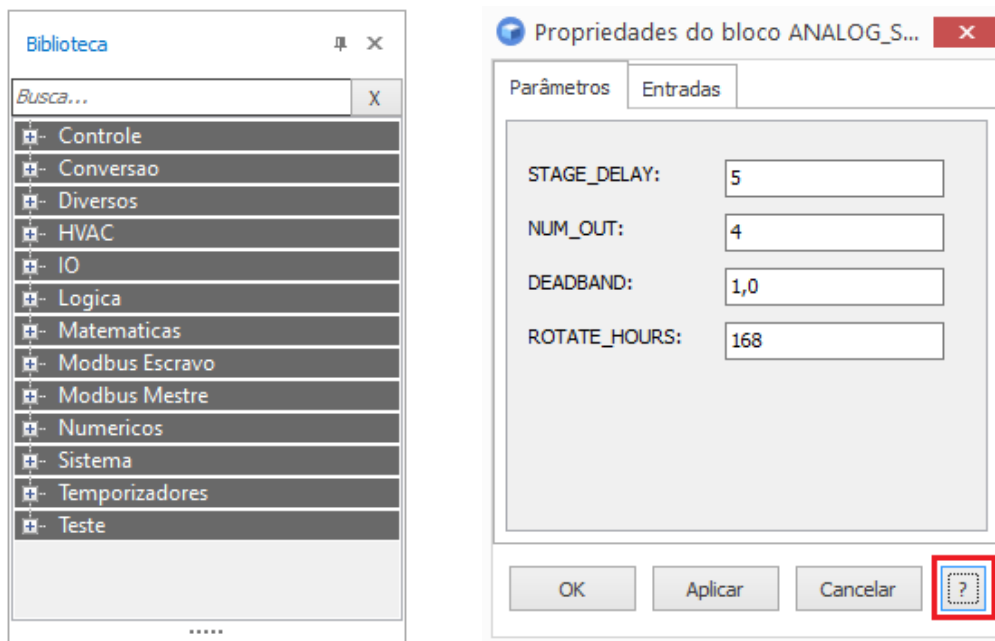
7. Programando com o TProg

7.1. Inserindo um bloco de função

Na biblioteca de instruções (canto esquerdo do TProg) encontram-se os blocos de funções divididos em grupos de funcionalidades.

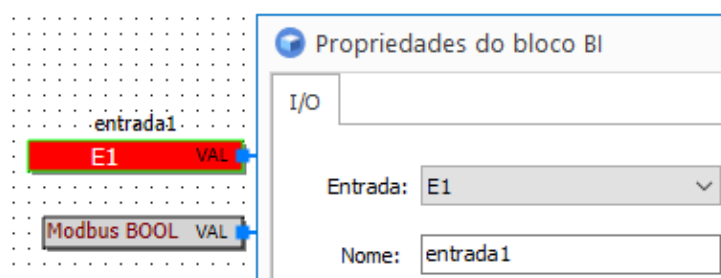
Para inserir o bloco no diagrama, clique (mouse) e mantenha segurado sobre o bloco e arraste a um diagrama, soltando o botão do mouse no local onde o objeto será inserido.

As propriedades do bloco são acessadas através de um duplo click sobre o próprio bloco e para visualizar detalhes do funcionamento, clique em “?”.



7.2. Identificação dos blocos

Para identificar e organizar as variáveis usadas na programação é essencial atribuir nomes a elas, para isso, clique duas vezes sobre o bloco de função e insira um nome.



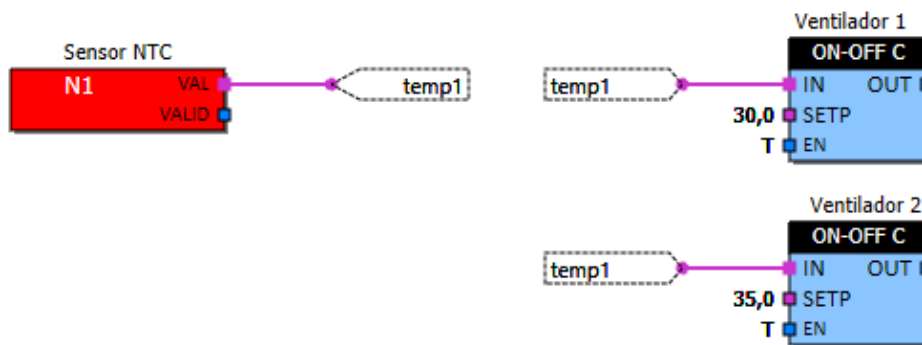
7.3. Label

LABELs é uma ferramenta gráfica que permite a conexão entre blocos sem a necessidade de usar uma linha. Eles possuem a função de organizar o diagrama gráfico, evitando que uma quantidade muito grande de linhas deixe a lógica poluída, dificultando a visualização do programa.

A conexão é **global**, o mesmo *label* “de saída” pode ser usado diversas vezes para interconectar os blocos em diagramas diferentes. O *label* faz a conexão entre todos os demais *labels* de mesmo nome (letra maiúscula e minúscula diferencia), realizando a ligação conectadas a ele.

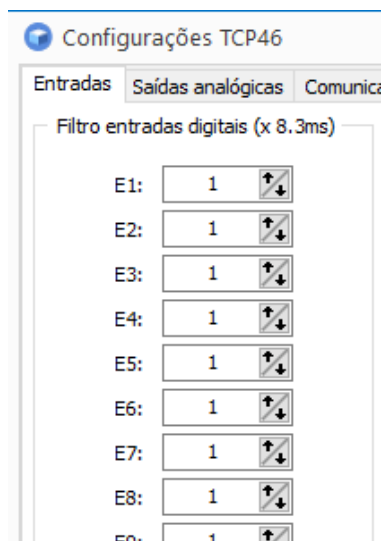
Importante: use *label* ao invés de repetir as entradas físicas (E1 ~ E18), isso diminui o tempo de *Scan*, aumentando a velocidade de leitura da programação.

Ex: o *label* “temp1” está copiando o valor de temperatura recebido pelo NTC (conectado a N1) para os blocos de resfriamento.



7.4. Filtro das entradas digitais

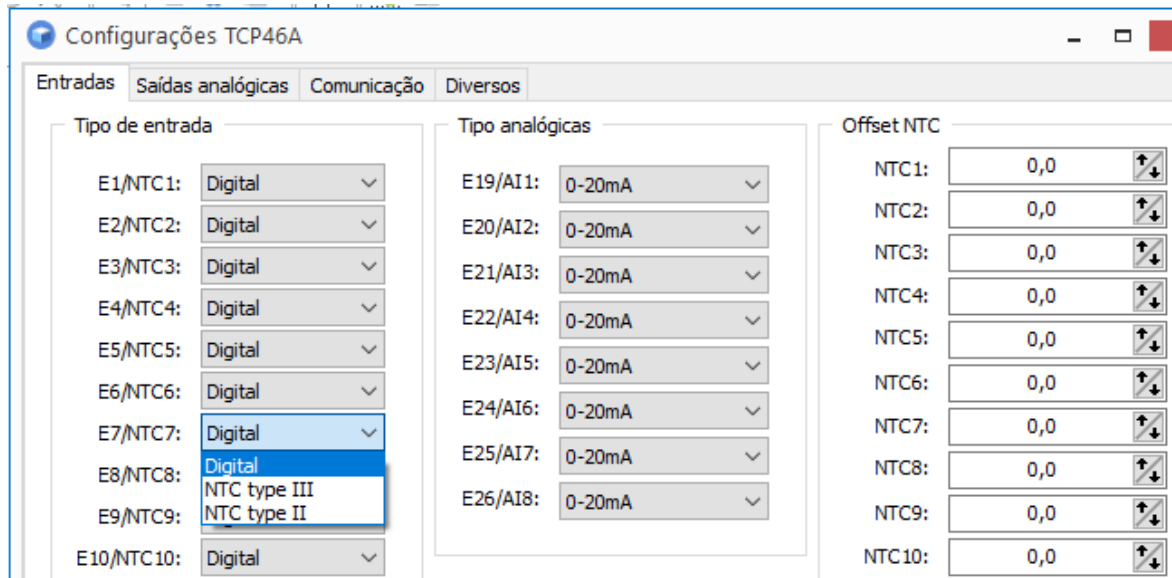
As entradas digitais são lidas pelo processador a cada 8,3ms, mas é possível adicionar um filtro para evitar leituras incorretas devido ao repique das chaves mecânicas (**bouncing**). O filtro é especificado para cada entrada (E1 ~ E18) através de múltiplos do valor 8,3ms, em: Configurações > Entradas > Filtro entradas digitais (x8.3ms).



7.5. Entradas digitais/NTC e entradas analógicas no TCP46A

Todas as 18 entradas digitais (DI1~DI18) do modelo TCP46A podem ser configuradas individualmente para acionamento digital ou leitura de sensores NTC tipo II/III + Offset.

O TCP46A possui 8 entradas analógicas configuráveis para tensão (0/2~10V) ou corrente (0/4~20mA).



7.6. Configurando entradas e saídas analógicas

É possível a leitura (entradas) e escrita (saídas) de **tensão** (0/2~10V) ou **corrente** (0/4~20mA) e a leitura de sensores NTC (°C). A grandeza e escala pode ser ajustada de forma independente por canal.

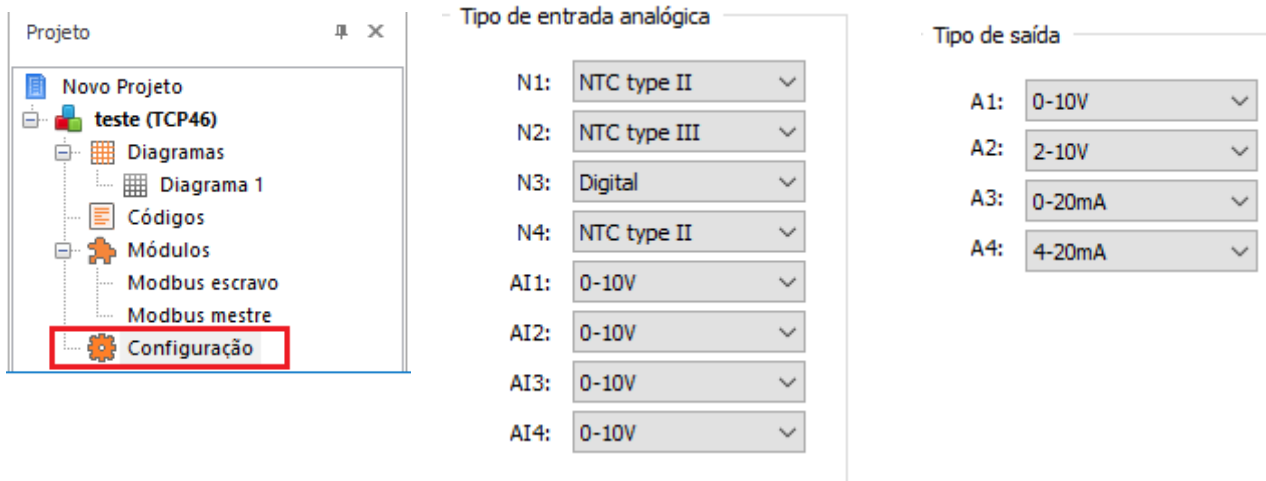
Entradas analógicas: Acesse: Projeto > Configuração > Entradas analógicas

Entradas N1 ~ N4: sensor NTC tipo II ou tipo III, configurável para contato seco ou NPN.

Entradas AI1 ~ AI4: leitura de tensão ou corrente.

Saídas analógicas: Acesse: Projeto > Configuração > Saídas analógicas

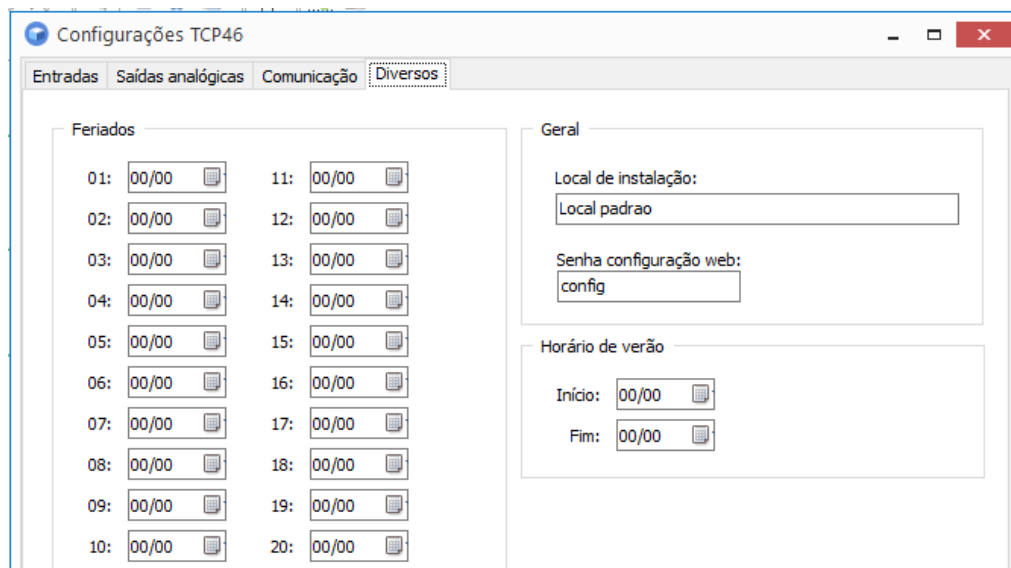
Saídas AO1 ~ AO4: escrita de tensão ou corrente.



7.7. Configurando feriados e horário de verão

O TCP46 possui RTC (relógio em tempo real) e pode ser usado em conjunto com as funções **Schedule_Offset** e **MDBS_schedule**. Para o correto funcionamento especifique os feriados anuais (até 20), data inicial e final do horário de verão.

Acesse: Projeto > Configuração > Diversos



7.8. Interfaces de conexão entre PC e CLP

Para conectar o CLP ao PC para iniciar a programação, uma interface de conexão deve ser definida, podendo ser **Modbus UDP**, **ModbusTCPGateway** ou **Modbus RTU**.

Definimos em: “**Arquivo > Opções > Conexões**”.

Diferença entre o Modbus TCP e UDP:

O User Datagram Protocol (UDP) é um protocolo simples da camada de transporte, permite a aplicação e o envio de um datagrama encapsulado em um pacote IPv4 ou IPv6. O UDP não utiliza o bit de confirmação, permite uma velocidade de transmissão maior e a confiabilidade na comunicação é a mesma obtida no TCP, pois a confirmação é feita pelo protocolo Modbus, que sempre exige do escravo uma confiabilidade do recebimento da mensagem, **por isso utilizamos a conexão UDP**.

O protocolo TCP faz uso do *Acknowledgment* (bit de confirmação), garantindo redundância na confiabilidade na troca de dados pois agrega em seu **header**, bits de controle de fluxo e recebimento.

7.8.1. Conexão Modbus UDP:

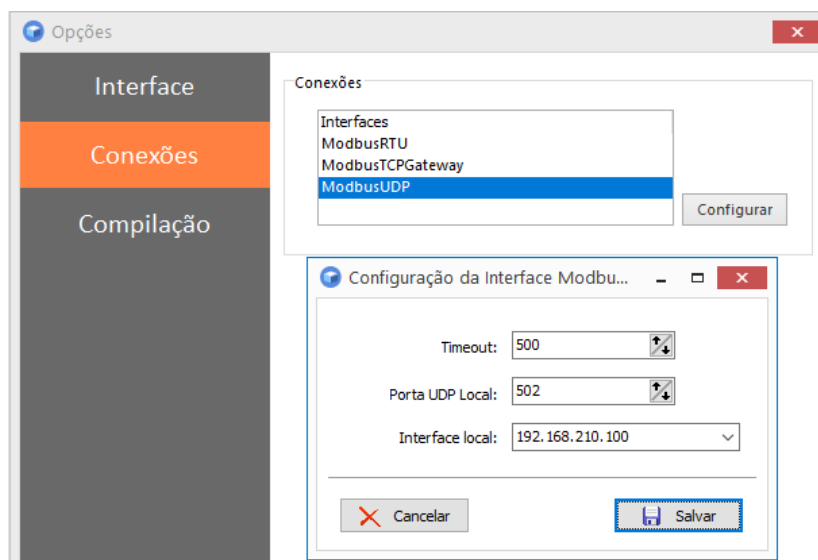
Passos para a conexão UDP:

Passo 1:

Certifique que o cabo de rede está conectado entre PC e CLP ou entre PC, CLP e roteador, e que os endereços de IPs estejam na **mesma** faixa de rede.

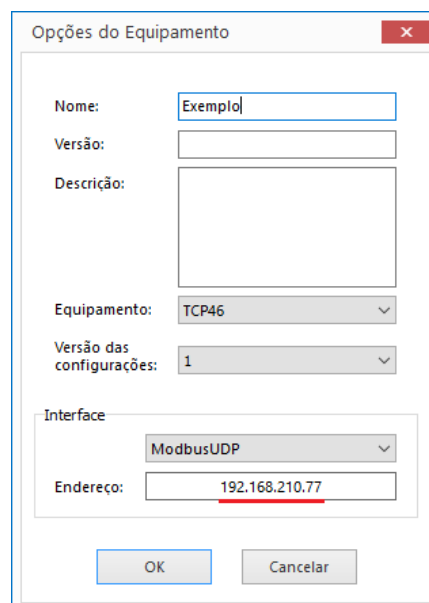
Passo 2:

Insira o **IP do PC** em: Arquivo > Opções > Conexões > ModbusTCP > Configurar > Digite o IP em “Interface local”.





Passo 3:

Insira o **IP do CLP** em: Projeto > Opções > Altere a interface para “ModbusUDP” > Digite em endereço o IP do CLP.



Obs: o passo 3 é necessário sempre que criamos um novo projeto no TProg.

Pronto, já é possível conectar o PC e o TCP46 pelo protocolo Modbus UDP!

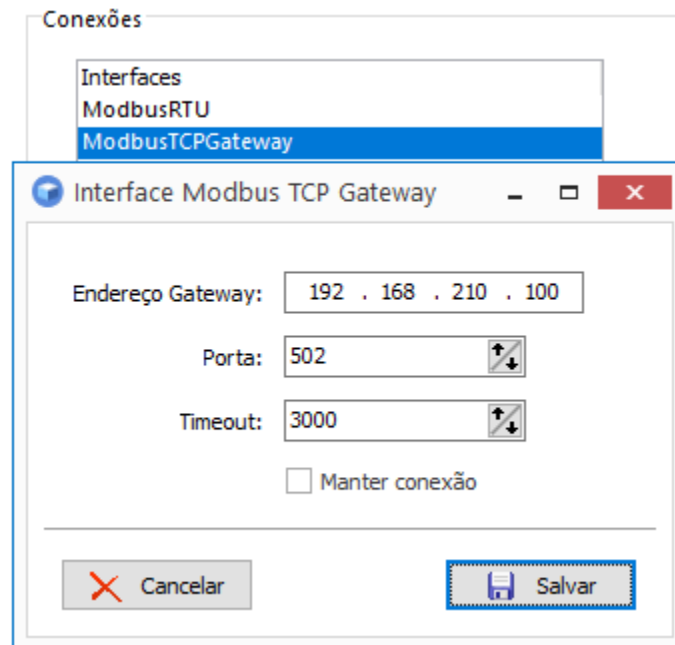
Teste a conexão clicando na joaninha  e verifique se os modos de operação (ao lado da joaninha) estão assim: .

7.8.2. Conexão Modbus TCP

O TCP46 não é um Gateway Modbus TCP para RTU, ou seja, não é possível empacotar uma solicitação em Modbus TCP e desempacotar em Modbus RTU para enviar a um dispositivo escravo conectado à rede serial do TCP46 ou o processo inverso, porém, é possível conectar o TCP46 ao TProg de uma maneira facilitada e sem definir um “Unit ID”.

Passos para a conexão Modbus TCP:

Arquivo > opções > conexões > selecione “ModbusTCPGateway” > insira o IP do CLP em “Endereço Gateway” e mantenha os demais parâmetros: Porta = 502 e Timeout = 3000.



7.8.3. Conexão Modbus RTU utilizando o conversor serial US485:

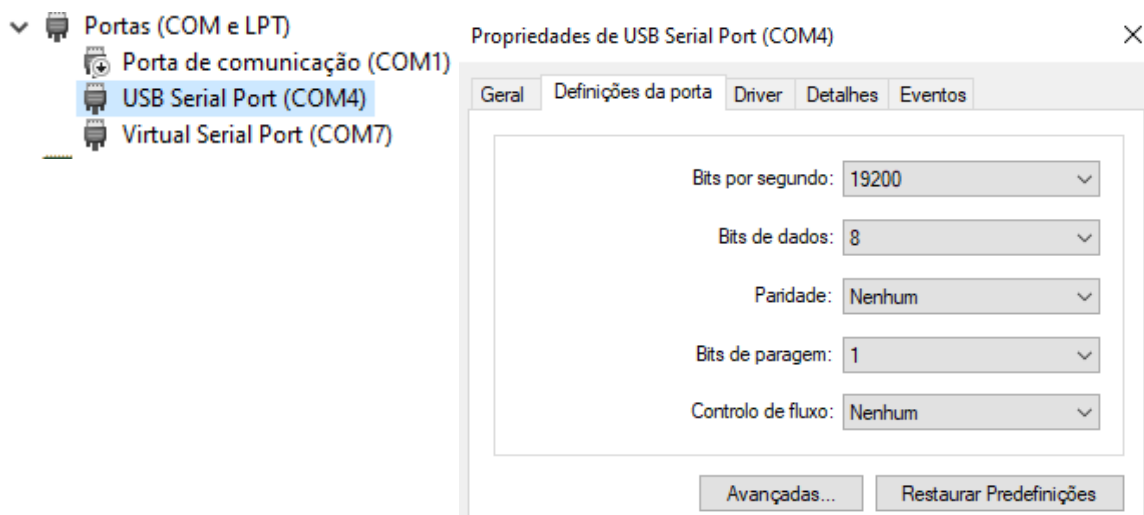
No TCP46, esse protocolo industrial é transmitido pelo meio físico RS485, os dados são enviados e recebidos de modo assíncrono.

Para essa conexão, um conversor serial do tipo USB para RS485 (**US485 da Tecnolog**) é necessário para criar uma porta serial virtual (COM) no computador.

Passos para a conexão RTU:

Passo 1: Ao usar o conversor US485 pela primeira vez, é necessário baixar o driver do conversor no site da Tecnolog e instalá-lo.

Passo 2: Acesse as propriedades do conversor e anote os parâmetros da Porta COM. Acesse: Iniciar > Gerenciador de dispositivos > Portas (COM e LPT) > Encontre o driver "Virtual Serial Port" > duplo click sobre o driver do conversor



Passo 3: No WebServer,

Acesse: Modbus escravo > Porta 1 > escolha uma porta (auxiliar ou principal) que está conectado o conversor serial > copie os parâmetros do conversor serial.

Modbus Escravo

Endereço Modbus:

Porta 1

Porta RS485:

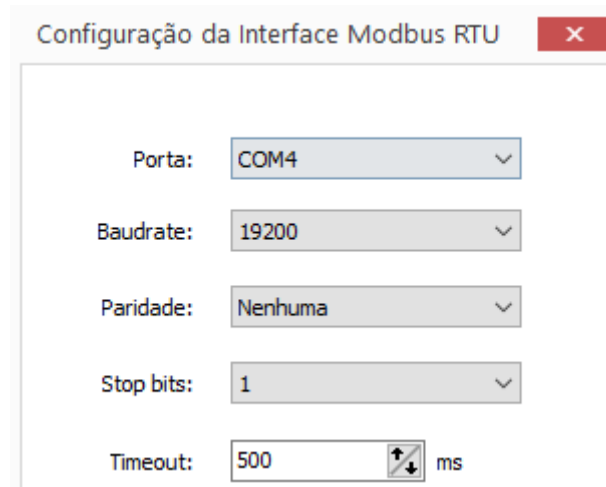
Velocidade:

Paridade:

Stop bits:

Atraso transmissão: ms

Passo 4: No Tprog, acesse: “Arquivo > Opções... > Conexões > ModbusRTU > Configurar > Copie os parâmetros do driver do conversor serial > OK”

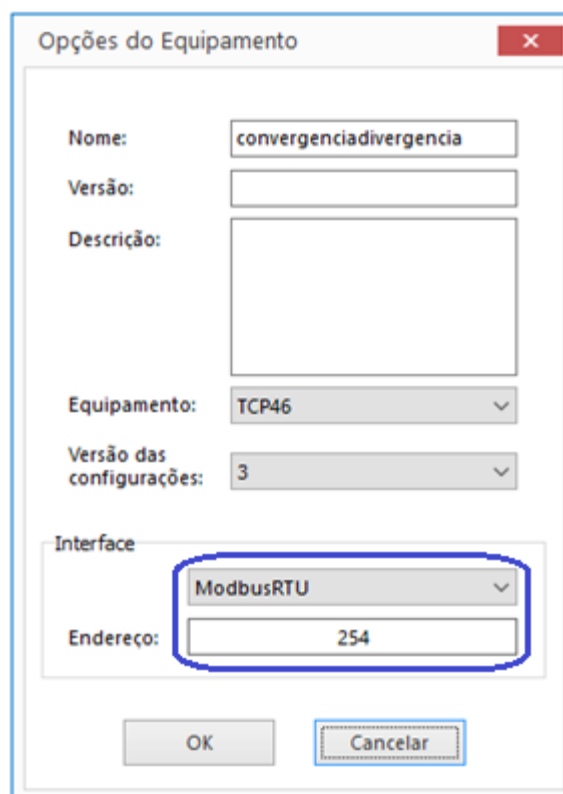


The screenshot shows a dialog box titled "Configuração da Interface Modbus RTU". It contains the following settings:

- Porta: COM4
- Baudrate: 19200
- Paridade: Nenhuma
- Stop bits: 1
- Timeout: 500 ms

Passo 5: Ajuste a interface Modbus RTU no TProg

Acesse: “Projeto > Opções > Interface (ModbusRTU) > defina ModbusRTU > digite o endereço Modbus escravo do equipamento (declarado no WebServer).



The screenshot shows a dialog box titled "Opções do Equipamento". It contains the following settings:

- Nome: convergenciadivergencia
- Versão: (empty)
- Descrição: (empty)
- Equipamento: TCP46
- Versão das configurações: 3
- Interface: ModbusRTU (highlighted with a blue box)
- Endereço: 254

Buttons: OK, Cancelar

7.9. Modos de operação (rodar, pausar e debug)

Modo Rodar (Projeto > Rodar)

Permite a CPU executar o processamento cíclico do programa.

Ao alterar para o modo "Rodar", as memórias e funções não retentivas são inicializadas com zero, as E/S são atualizadas, a comunicação serial é iniciada e o funcionamento do sistema é verificado.

Modo Pausar (Projeto > Pausar)

Permite pausar o processamento do programa temporariamente de modo a manter os estados das memórias e as E/S.

Ao alterar para o modo **rodar**, o processamento cíclico do programa retoma de onde foi pausado.

Modo Debug/Compilar (Projeto > Compilar)

Essa função permite eliminar erros de endereçamentos antes do download da programação para o CLP e para verificar a ocupação das memórias do hardware.

Atalho: **Ctrl + B**

Obs: não é preciso compilar o projeto todas as vezes antes do download.

7.10. Compilação

O compilador é uma ferramenta para verificação de erros, os mesmos são descritos na caixa de compilação e o bloco que contém o erro é destacado (vermelho) no diagrama.

Atalho para compilar: Ctrl + B ou acesse as abas: Projeto > Compilar

O compilador apresenta a ocupação das memórias do hardware quando não apresentar erros na lógica.

Ocupação das memórias: disponibiliza o valor atual ocupado (0 ~ 100%) das memórias de programa, dados, configurações e acumuladores.

Compilação

```

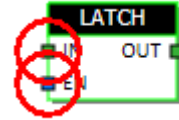
- [11:26:06] Compilando diagrama "Diagrama 1"
- [11:26:06] Compilando módulo "Modbus escravo"
- [11:26:06] Linkando
- [11:26:06] Espaços ocupados: Memória de programa: 162 (0,1%) | Memória de dados: 17 (0,2%) | Configurações: 0 (0,0%) | Acumuladores: 0 (0,0%)
- [11:26:06] Projeto compilado com sucesso
  
```

Erros de compilação conhecidos:

Link desconectado: bloco com fio desconectado ou há algum parâmetro do bloco a ser preenchido.

Compilação

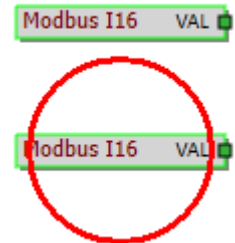
- [-] [11:13:18] Compilando "tprog_teste"
 - [ERRO GRÁFICO] [11:13:18] Link está desconectado e não contém valor default.
 - [ERRO GRÁFICO] [11:13:18] Link está desconectado e não contém valor default.



Bloco com endereço do registro duplicado!: dois ou mais blocos com o mesmo registro de rede.

Compilação

- [-] [11:23:20] Compilando "tprog_teste"
 - [11:23:20] Compilando diagrama "Diagrama 1"
 - [11:23:20] Compilando módulo "Modbus escravo"
 - [ERRO COMPILADOR] [11:23:20] MDBS: bloco com endereço do registro duplicado!




7.11. Download (PC > TCP46)

É possível descarregar o programa (TProg > CLP) após configurar a interface de conexão (item 7.8).

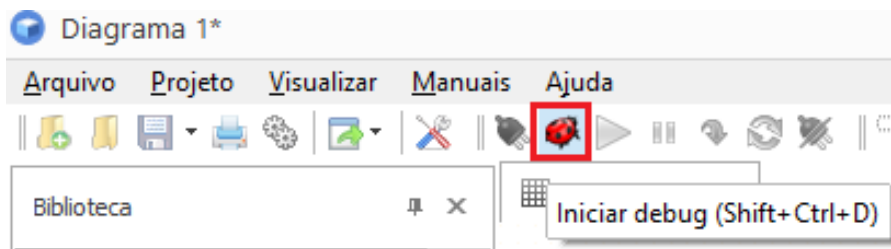
O programa permanece no CLP por tempo indeterminado (mesmo com a falta de energia elétrica), pois o programa é gravado em uma memória não volátil (Flash).



Importante: O Upload (CLP > PC) não é permitido, faça backup do arquivo final da programação do seu CLP e o mantenha seguro.







Download da programação: Utilize o *debug* (depurar) sempre que alterar a programação ou reiniciar o software. A nova programação irá **sobrepôr** o programa atual do CLP sempre que realizar uma depuração.







Para depurar: ícone  ou o atalho (Shift + Ctrl + D).

Após depurar, clique em “Rodar”  para iniciar a programação (*Scan*).



Reconexão sem alteração: Quando a programação não for alterada e o CLP estiver desconectado do TProg, utilize o ícone  para reconectar e  para desconectar.

Status do CLP conectado:      

Status do CLP não conectado:      

Obs: é possível interromper o carregamento do programa. Para isso, durante a inicialização do equipamento, clique rapidamente a tecla frontal 5 vezes. Caso o programa não seja carregado ou ocorrer um erro de execução, o led frontal pisca rapidamente.

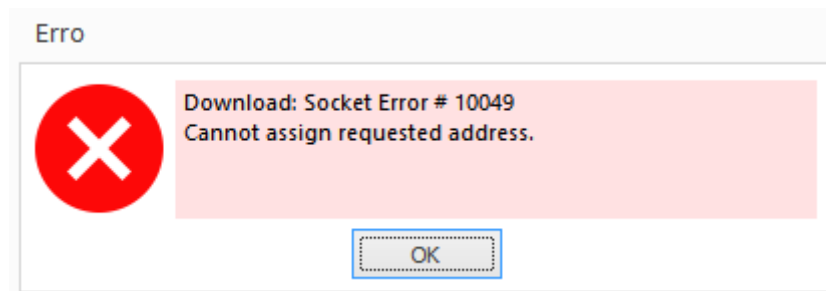
7.11.1. Soluções para erros de conexão

Esse capítulo mostra soluções para os erros apresentados no TProg ao tentar se conectar, desconectar ou fazer download da programação (PC > TCP46).

Download Socket Error # 10049

Causa: IPv4 não definido no TProg.

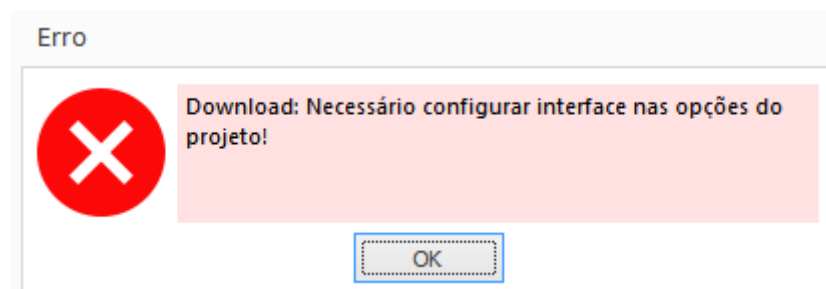
Ação: Configure a interface de conexão Modbus em: Arquivo > Opções > Conexões



Download: Necessário configurar interface nas opções do projeto!

Causa: IP do TCP46 não definido no TProg.

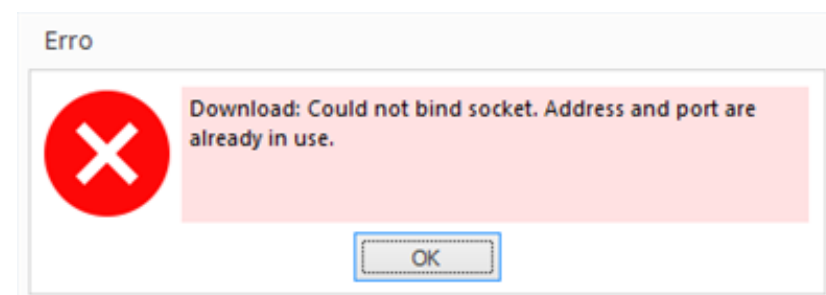
Ação: Configure a interface de conexão Modbus em: Projeto > Opções > Interface



Download: Could not bind socket. Address and port are already in use.

Causa: O endereço IPv4 do computador está definido incorretamente no TProg ou o CLP está desligado.

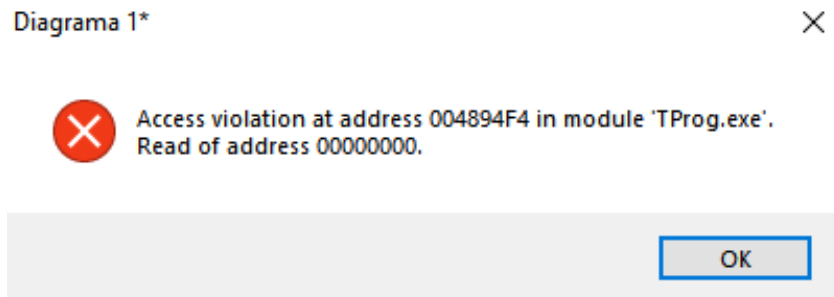
Ação: Defina o IPv4 do PC em: Arquivo > Opções > Conexões > ModbusUDP > Interface local ou reinicie o TProg.



Access violation at address 004894F4 in module "TProg.exe", Read of address 00000000

Causa: O controlador já está desconectado (erro que apresenta ao tentar se desconectar do CLP)

Ação: a edição da programação já está disponível no TProg



Download: Erro identificando o equipamento!

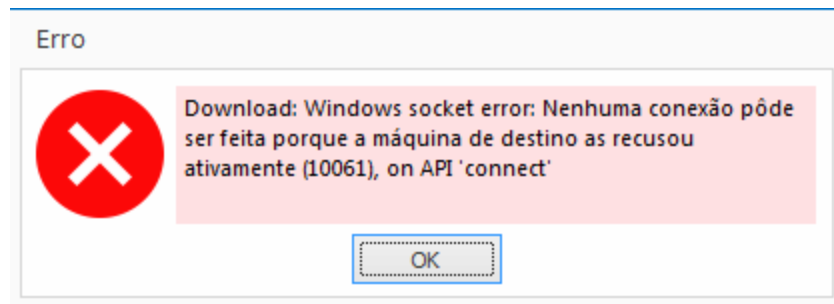
Causa: controlador em falha, cabo de rede desconectado ou equipamento desligado

Ação 1: desligue o CLP, aguarde 10 segundos e ligue o equipamento.

Ação 2: verifique se o CLP está energizado e com o cabo de rede conectado

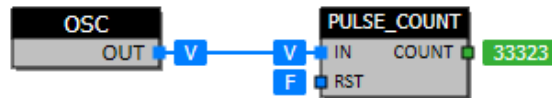
Download: Windows socket error: Nenhuma conexão pôde ser feita porque a máquina de destino as recusou ativamente (10061), on API 'connect'.

Ação: acesse "Opções do Equipamento" e altere a interface de ModbusTCPGateway para ModbusUDP e no "Endereço" digite o IP do CLP.



7.12. Monitoramento online

O monitoramento online da programação é iniciado após estabelecer a conexão entre TCP46 e TProg e alterar para o modo de operação “rodar”. Os valores são constantemente atualizados (a cada 1 segundo), permitindo verificar o status dos pinos e os valores atuais nos blocos e variáveis.



Verdadeiro: estado T (*true*) ou V (verdadeiro) e a espessura da linha aumenta.



Falso: estado em F (*false/falso*) e a espessura da linha não muda.



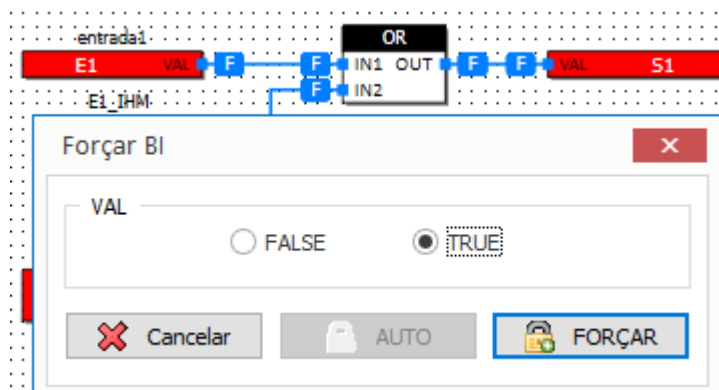
NAN: incoerência lógica no bloco (Ex: realizar uma divisão por 0, definir o valor mínimo e o valor máximo de um bloco igual a zero, entre outros).

Dica: utilize blocos MDBS_IN para inserir/simular valores de entrada.

7.13. Forçando entradas digitais

É possível forçar as entradas digitais para simular situações de acionamentos, para isso, o CLP deve estar conectado e em modo “Rodar”. **O forçamento é apenas para o monitoramento** do TProg, o endereço Modbus da entrada não irá alterar o valor.

Com um duplo clique sobre a entrada escolhida, clique em “Forçar”, o estado (VAL) altera de “False - F” para “True - V”, ativando a entrada. Forçar novamente a mesma entrada, o estado altera de “V” para “F”, retornando ao estado normal liberando o ponto. Outra forma de liberar o forçamento é clicar em Auto.



Os forçamentos são indicados por um retângulo vermelho sobre o ponto.



7.14. Tempo de varredura (Scan)

O tempo de varredura (*Scan*) é determinado pela quantidade de blocos de funções utilizados, pontos de leitura das IO, tráfego de dados entre as portas de comunicação, processamento interno e outros.

O número de **ciclos/s** atual é visualizado no canto inferior direito do TProg em modo “Rodar”.


Conectado a ihm_tcp46_am8di | FW: 1.02 | Rodando | 2136 ciclos/s

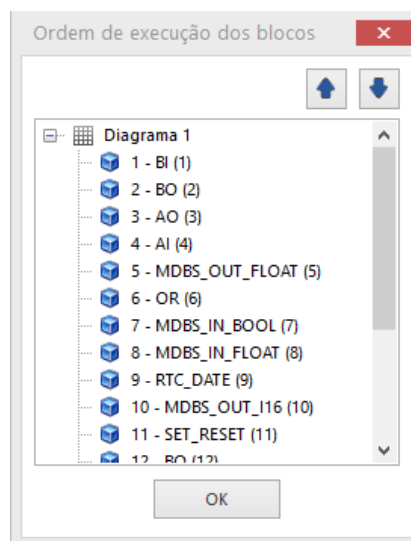
Para calcular o **Scan**: $T = 1 / \text{ciclos}$ (em segundos)

Tempo de processamento para cada instrução:

- Tempo de *Scan* sem programa: **50us**
- Tag Modbus mestre = **25us**
- Uma saída física = **10us**
- Leitura de uma entrada física = **28us**
- Tag Modbus escravo = **6us**
- Leitura de um Tag escravo pelo mestre Modbus = **5us**


7.15. Ordem de execução dos blocos

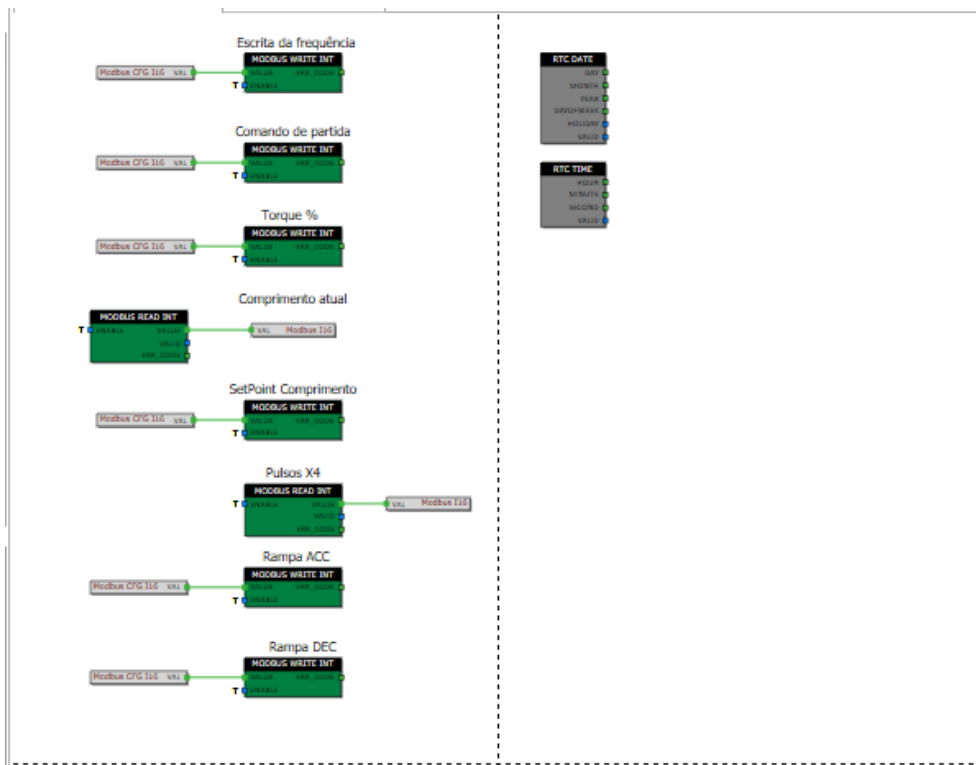
A ordem de execução dos blocos segue a ordem cronológica definida na programação para cada diagrama, e pode ser visualizada na aba (Projeto > ordem de execução...) ou pelo ícone . Em cada diagrama essa ordem dos blocos pode ser livremente modificada utilizando as setas sobe e desce para deslocar o bloco selecionado na lista.



7.16. Impressão de diagramas

Os diagramas podem ser impressos ou salvos no formato (.pdf) para fins de documentação, cada diagrama é um documento e cada divisão do diagrama é uma página desse documento.

Acesse: Arquivo > Imprimir tudo ou o atalho (Ctrl + P), clique no ícone print  e selecione a impressora ou o driver para gerar um documento no formato PDF.



7.17. Endereçamento do TCP46EX e configuração no TProg

É possível acrescentar até 8 expansões para cada CPU TCP46 (totalizando 414 pontos). O tempo de atualização de cada remota é 25ms, e a cada remota adicional devemos somar mais 25ms ao tempo de atualização das remotas.

Passo 1: definir o endereço do módulo

A soma binária das chaves **8 a 10** da *DIP Switch* frontal da expansão, sendo 1 (on = ligada) e 0 (off = desligada).

Posição	Endereço
_000	0
_001	1
_010	2
_011	3
_100	4
_101	5
_110	6
_111	7

Passo 2: Habilitar o uso das expansões na CPU

Acesse: WebServer > Configurações > Expansões > Porta RS485, e escolha a porta principal (inferior) ou auxiliar (superior).

Obs: essa opção irá aparecer para versões do firmware da CPU igual ou superior a V1.10 e caso não apareça entre em contato com o suporte técnico para atualizar a CPU.

Expansões

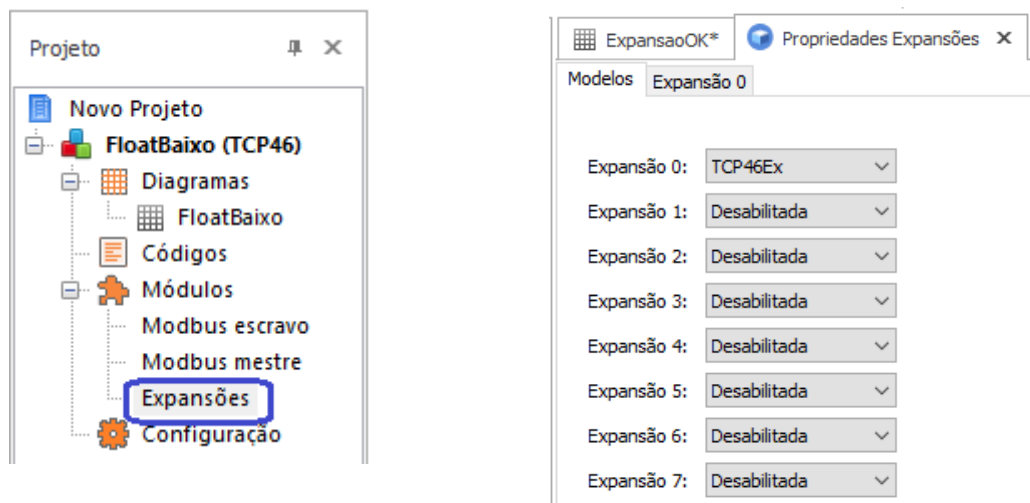
Porta RS485:

Gravar

Passo 3: Habilitar as expansões no TProg

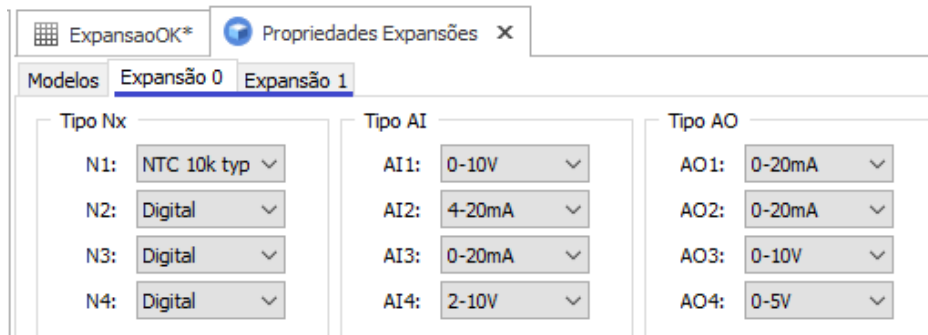
Acesse: árvore do projeto > **expansões**.

Habilite as expansões TCP46EX a serem utilizadas (0~7) conforme sua estação.



Passo 4: Definir a grandeza das entradas e saídas analógicas e NTC das expansões

Após habilitar as expansões, suas respectivas abas de configurações irão aparecer. É preciso definir os pontos individualmente conforme sua aplicação e caso não defina as entradas Nx (1~4), o compilador apresentará o erro “Esperado número ou define tipo inteiro”.



Passo 5: Inserir um bloco EXP

Os blocos usados para as expansões estão na biblioteca grupo **IO**, sendo:

EXP_**BI** = entrada digital/NTC (contato seco)

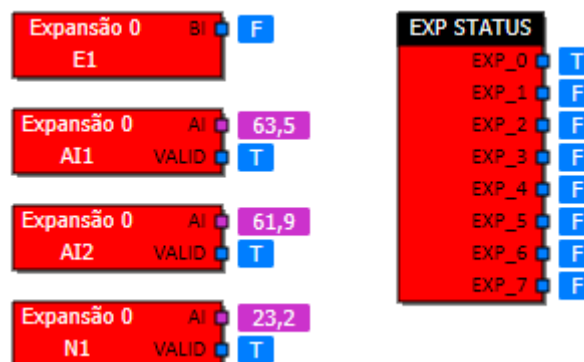
EXP_**BO** = saída digital

EXP_**AI** = entrada analógica/NTC

EXP_**AO** = saída analógica

EXP_**STATUS** = validade em cada expansão

Obs: não é possível utilizar as entradas rápidas nas expansões.



Passo 6: Definir no bloco EXP o número da expansão e o ponto

Após inserir o bloco no diagrama, para direcioná-lo:

- Duplo click sobre ele > escolha o número da expansão > escolha o ponto

Led operação nas expansões:

O Led de operação indica o funcionamento da expansão, sendo:

- Led **piscando lento**: conexão estabelecida;
- Led **piscando rapidamente**: conexão **não** estabelecida ou perdida.

7.18. Tags de comunicação Mestre e Escravo

7.18.1. Tag Modbus Mestre (MDBM)

Os blocos Modbus Mestre oferecem suporte ao protocolo Modbus na modalidade mestre, permitindo ao controlador iniciar leituras e escritas em outros controladores na rede Modbus. Parte do tratamento da comunicação é feito automaticamente pelo firmware do equipamento, simplificando a aplicação para o programador. Os blocos disponíveis deste módulo permitem fácil interface de rede com a lógica interna do controlador.

Protocolo Modbus suportados:

- Modbus RTU (porta RS485).
- Modbus/TCP (porta ethernet).
- Modbus/UDP (porta ethernet).

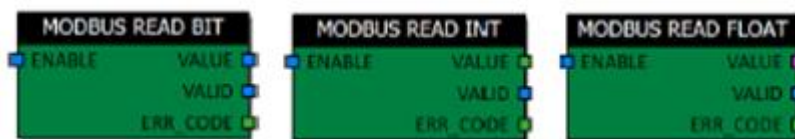
Funções Modbus suportadas:

- 01x – Read coils
- 02x – Read inputs
- 03x – Read holding registers
- 04x – Read input registers
- 05x – Write single coil
- 06x – Write (preset) single register.
- 15x – Write multiple coils
- 16x – Write (preset) multiple registers

Tipos de blocos:

Existem 2 grupos de blocos (**leitura e escrita**) e cada grupo possui suas variações de acordo com o tipo de dado: **bool**, **int** ou **float**.

Blocos de leitura: o controlador faz a leitura da variável do equipamento Modbus escravo e disponibiliza o seu valor para o uso na lógica interna.



ENABLE: entrada que controla a comunicação na rede. Quando falsa, a leitura é interrompida. Obs: é possível que um disparo de leitura já esteja ativo e o controlador ainda faça uma última leitura antes de ser interrompido.

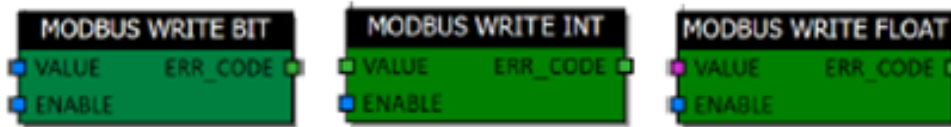
Intervalo de leituras: parâmetro de disparo entre uma leitura e outra (em décimos de segundo). As leituras são controladas pela entrada **ENABLE** caso este parâmetro seja zero, basta o pulso de 1 ciclo nesta entrada para agendar uma leitura. Caso a entrada seja mantida ativada, a leitura é contínua.

VALUE: fornecem os valores lidos para a lógica interna.

VALID: saída booleana indica se o valor lido (e apresentado) é válido. Caso tenha ocorrido algum erro de leitura, o sinal **VALID** fica com o valor falso.

ERR_CODE: saída que indica o código de erro ocorrido na última tentativa de leitura. Consultar a tabela (8.15.3) do manual para os códigos existentes.

Blocos de escrita: o controlador escreve um valor da lógica interna a um equipamento Modbus escravo da rede.



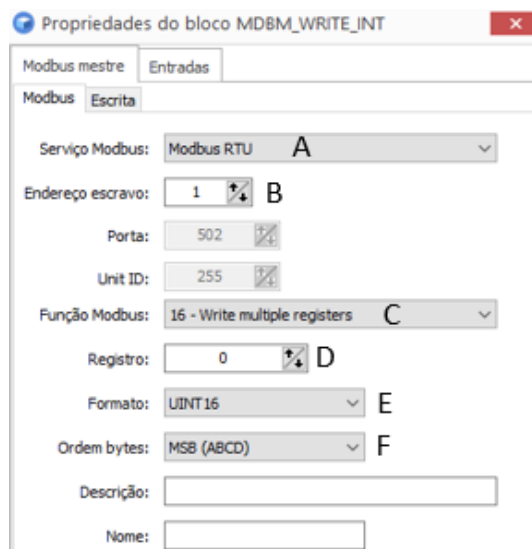
Existem 3 formas de disparar a escrita: habilitação manual, tempo ou variação de valor.

Habilitação manual: A entrada *ENABLE* habilita a comunicação na rede e permite controlar o disparo de forma manual. Se a entrada estiver desativada, nenhuma escrita é feita, independente dos outros 2 parâmetros. Caso os disparos por tempo e variação estiverem zerados, a entrada *ENABLE* controla manualmente a escrita. Basta um pulso

Intervalos regulares de tempo: é controlado pelo parâmetro do bloco 'Intervalo escrita', configurado em décimos de segundo. Caso este parâmetro seja zero, o disparo por tempo é desabilitado.

Variação do valor da entrada: caso a variação em relação à última escrita seja maior ou igual ao parâmetro 'Disparo variação', uma escrita será agendada. No caso do tipo de dado BIT, a escrita é feita sempre que o valor é alterado. Se o valor da variação é configurado como zero, este tipo de disparo fica desabilitado.

Configuração do bloco:



A – Serviço (Modbus RTU, TCP e UDP)

Esta configuração seleciona a variação do protocolo e a interface a ser usada para a comunicação. Os serviços disponíveis dependem do controlador usado.

B – Endereçamento

Neste parâmetro é configurado o endereço escravo do equipamento a ser lido/escrito.

Modbus RTU: disponibilizado na interface RS485, apenas o endereço do equipamento na rede é configurado, variando de 1 a 255.

Modbus/TCP e Modbus/UDP

Configurados o endereço IP do equipamento na rede e a porta Modbus.

A porta **502** é padrão no protocolo Modbus.

Caso o equipamento seja um gateway Modbus/TCP para Modbus RTU, será necessária a configuração do parâmetro Unit ID, que define o endereço do equipamento na rede RS485 abaixo do gateway.

C – Função Modbus

Seleciona a função Modbus a ser usada para a leitura/escrita.

A lista disponível varia conforme o tipo do bloco.

- Blocos de leitura:

Tipo BIT: funções 01 (Read coils) e 02 (Read inputs) fazem as leituras das entradas e saídas binárias do equipamento. Já as funções 03 (Read holding registers) e 04 (Read input registers) fazem a leitura de um registro (16 bits) Modbus e convertem o dado para binário. Um valor diferente de zero no registro é mostrado como TRUE.

- Blocos de escrita: usando as funções 06 (write single register) ou 16 (write multiple registers) são usados os valores 0 e 1 na escrita do registro para os valores de entrada FALSE/TRUE respectivamente.

D – Registro (0 ~ 65535)

Seleciona o endereço do registro a ser lido/escrito no equipamento da rede. Este valor é enviado sem nenhum tipo de offset. Alguns equipamentos utilizam uma nomenclatura de endereçamento onde é necessário usar um offset para encontrar o endereço real do registro.

E – Formato

Esta configuração, habilitada para o tipo de dado INTEIRO define o formato do número no equipamento destino. Pode ser uma das opções:

- UINT16 (inteiro sem sinal de 16 bits) – ocupa 1 registro Modbus.
- INT16 (inteiro com sinal de 16 bits) – ocupa 1 registro Modbus.
- UINT32 (inteiro sem sinal de 32 bits) – ocupa 2 registros Modbus.
- INT32 (inteiro com sinal de 32 bits) – ocupa 2 registros Modbus.

F – Ordem de bytes (Swapped)

Esta opção permite inverter a ordem dos bytes que formam o valor no equipamento destino. Isto permite a comunicação com equipamentos de diversos fabricantes que definem ordens de bytes diferentes na memória. A nomenclatura padrão de um FLOAT (4 bytes), por exemplo, é ABCD, onde A é o byte mais significativo e D o menos.

Usando como exemplo um float de valor 12.3 (bytes ABCD = 4144CCCD), a escrita no equipamento destino, ocupando 2 registros Modbus será feita da seguinte forma nas ordens:

ABCD:41 44 CC CD-> REGISTRO 1:41 44
 REGISTRO 2:CC CD
BADC:41 44 CC CD-> REGISTRO 1:44 41
 REGISTRO 2:CD CC

CDAB:41 44 CC CD-> REGISTRO 1:CC CD
 REGISTRO 2:41 44

DCBA:41 44 CC CD-> REGISTRO 1:CD CC
 REGISTRO 2:44 41

Configurações do protocolo

As configurações básicas do protocolo Modbus Mestre nos controladores são:

Timeout: Tempo máximo de espera pela resposta nas requisições.

Retries: Número de tentativas de comunicação em caso de falha.

O controlador pode ter configurações adicionais para o protocolo e interfaces. Consulte o manual dele para maiores informações.

7.18.2. Tabela de erros Modbus Mestre

A saída *ERR_CODE* dos blocos Modbus mostra o número do erro.

Valores abaixo de 100 são gerados pelo equipamento Modbus (exceções) e valores acima de 100 são gerados no controlador.

Obs: O número 0 significa que o bloco não apresentou um erro de comunicação.

01	Illegal function
02	Illegal data address
03	Illegal data value
04	Server device failure
06	Server device busy
11	Gateway target device failed to respond
100	Timeout
101	Parâmetro inválido
102	Falha de alocação de memória
103	Interface desconectada
104	Resposta inválida
105	Erro de CRC
106	Transação inválida
107	Erro no TCP
199	Leitura não executada

7.18.3. Tag Modbus Escravo (MDBS)

Os blocos de comunicação MDBS recebem ou enviam dados da lógica interna para um dispositivo mestre da rede. Por padrão, nenhuma variável da lógica é acessível via protocolo, a não ser que um bloco de comunicação seja explicitamente criado.

Existem 3 tipos de MDBS: **IN**, **CFG** e **OUT**.

- **Blocos IN:** é um bloco de entrada, o dispositivo recebe um dado do dispositivo mestre da rede e esse dado é utilizado na lógica interna.
- **Blocos CFG:** é um bloco de entrada com o dado retentivo, ou seja, não se perde na falta de energia. Estes blocos podem ser usados para a criação de *Set Points* acessíveis via rede.
- **Blocos OUT:** é um bloco de saída no qual um dispositivo escravo envia um dado da lógica interna envia um valor a ser lido por um dispositivo mestre.

Modbus BOOL VAL

Modbus CFG I16 VAL

VAL Modbus FLT

Propriedades do bloco:

Registro: endereço definido para a troca de dados com o dispositivo mestre da rede.

Limites para escrita: Disponível para os blocos MDBS_IN e MDBS_CFG. Valor mínimo (Mínimo) e valor máximo (Máximo) de entrada permitido no registro. Caso o valor inserido esteja fora dos limites, o valor não é recebido pelo CLP e permanece o último dado válido recebido.

Valor inicial: valor do registro quando o CLP passar para “rodar”.

Modbus escravo

Registro

Registro:

Descrição:

Limites para escrita

Mínimo:

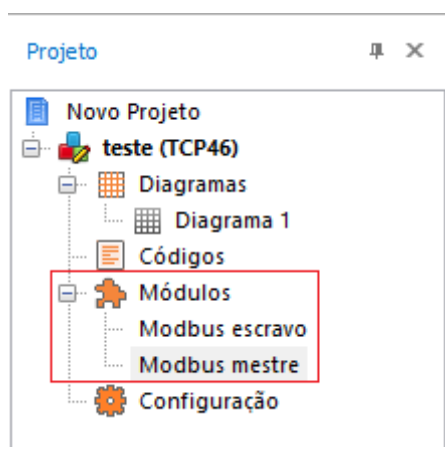
Máximo:

Valor inicial:

Nome:

7.19. Lista de registros Modbus (Tags)

Os registros Modbus (escravo e mestre) utilizados na programação podem ser visualizados e editados na janela: “Projeto” > “Módulos”



A lista de registros Modbus (Tags) pode ser impressa ou exportada (.xlsx ou excel) para fins de importação de Tags em IHMs (ver 8.18.1), supervisão ou banco de dados.



Diagrama	Nome	Registro	Tamanho	Tipo	Escrita?	Config?	Máximo	Descrição
Diagrama 1		1000	2	FLOAT	N	N		
Diagrama 1	E1_IHM	100	1	BIT	S	N	1	
Diagrama 1	AO4_IHM	1002	2	FLOAT	S	N	999999999	
Diagrama 1	SET_IHM	10	1	BIT	S	N	1	
Diagrama 1	RESET_IHM	11	1	BIT	S	N	1	
Diagrama 1	S1_IHM	500	1	BIT	N	N		

7.20. Importar Tags na IHM Weintek (EasyBuilder Pro)

Função que permite usar as Tags criadas no TProg para criar o projeto na IHM Weintek. Clique [aqui](#) para acessar o exemplo para comunicação entre IHM Weintek e CLP TCP46.

Tags exportadas pelo TProg:

	A	B	C	D	E	F	G	H	I	J
1						Dados Modbus Escravo				
2										
3	Diagrama	Nome	Registro	Tamanho	Tipo	Escrita?	Config?	Mínimo	Máximo	Descrição
4	OUT1	10	1	INT16	S	N	-32768	32767		
5	OUT2	12	2	FLOAT	S	N	-999999999	999999999		
6	OUT3	14	1	INT16	S	N	-32768	32767		
7	OUT4	16	2	FLOAT	S	N	-999999999	999999999		

Ajustando o excel para o padrão do EB Pro:

O arquivo excel exportado pelo TProg precisa ser ajustado para que o EB Pro identifique as características das Tags e possa importá-las corretamente. Para isso, ajuste ou crie o arquivo .xls conforme o padrão da IHM, demonstrado abaixo:

Foram utilizadas uma Tag de cada tipo (float, word e bool) para facilitar o entendimento.

Nome	Driver	Formato	Registro	x	Tipo de dado
OUT3	MODBUS TCP/IP (Zero-based Addressing)	3x	14		16-bit Unsigned
OUT4	MODBUS TCP/IP (Zero-based Addressing)	5x	16		32-bit Float
OUT5	MODBUS TCP/IP (Zero-based Addressing)	5x	18		32-bit Unsigned
OUT6	MODBUS TCP/IP (Zero-based Addressing)	3x_Bit	20		16-bit Unsigned

Coluna A: Nome = nome definido para a Tag (coluna Diagrama no Excel não editado)

Coluna B: Driver = conforme selecionado em System Parameters > Device type no EB

Coluna C: Formato: 3x_Bit (booleano), 3x (word) e 5x (float ou I32)

Coluna D: word/float = registro da tag ou caso bit = registro da tag + 00 (DI1 = 2000000)

Coluna E: deixar em branco

Coluna F: Tipo de dado (Word = 16-bit Unsigned/16-bit Signed ou Float = 32-bit Float)

Importando Tags no EBPro:

Acesse as abas: Project > Address > Import EXCEL... > selecione o arquivo .xls editado

Utilizando uma Tag no objeto:

Em propriedades do objeto: Settings > Tag Library... > "selecione uma Tag"

Read/Write

Device : MODBUS TCP/IP (Zero-based Addressing) ⌵ 🏠 ➕

Address : 3x ⌵ 0

🏠 🔍

🏠 ✎

Name	Data type
<input checked="" type="checkbox"/> OUT1	16-bit Unsigned
<input checked="" type="checkbox"/> OUT2	32-bit Float
<input checked="" type="checkbox"/> OUT3	16-bit Unsigned
<input checked="" type="checkbox"/> OUT4	32-bit Float
<input checked="" type="checkbox"/> OUT5	32-bit Unsigned

<

Notification Enable

Notification on in Enable

Tag :

8. Tabela de endereços Modbus

É possível ler (R) / escrever (W) **diretamente** os pinos do hardware do controlador através dos endereços Modbus listado abaixo no mapa Modbus do CLP, a partir de um dispositivo mestre da rede.

Os endereços em **branco** podem ser acessados nos modelos TCP46 e TCP46A, os endereços em **vermelho** podem ser acessados somente no TCP46A.

Obs: As entradas analógicas (20100 ~ 20153) podem ser acessada em ambos os controladores, mas a leitura de temperatura se limita ao modelo TCP46A.

Endereço	Nome	Tipo	Serviço	Descrição
20000	DI_01	WORD	R	Estado da entrada digital 1 (0 = off e 1 = on)
20001	DI_02	WORD	R	Estado da entrada digital 2 (0 = off e 1 = on)
20002	DI_03	WORD	R	Estado da entrada digital 3 (0 = off e 1 = on)
20003	DI_04	WORD	R	Estado da entrada digital 4 (0 = off e 1 = on)
20004	DI_05	WORD	R	Estado da entrada digital 5 (0 = off e 1 = on)
20005	DI_06	WORD	R	Estado da entrada digital 6 (0 = off e 1 = on)
20006	DI_07	WORD	R	Estado da entrada digital 7 (0 = off e 1 = on)
20007	DI_08	WORD	R	Estado da entrada digital 8 (0 = off e 1 = on)
20008	DI_09	WORD	R	Estado da entrada digital 9 (0 = off e 1 = on)
20009	DI_10	WORD	R	Estado da entrada digital 10 (0 = off e 1 = on)
20010	DI_11	WORD	R	Estado da entrada digital 11 (0 = off e 1 = on)
20011	DI_12	WORD	R	Estado da entrada digital 12 (0 = off e 1 = on)
20012	DI_13	WORD	R	Estado da entrada digital 13 (0 = off e 1 = on)
20013	DI_14	WORD	R	Estado da entrada digital 14 (0 = off e 1 = on)
20014	DI_15	WORD	R	Estado da entrada digital 15 (0 = off e 1 = on)
20015	DI_16	WORD	R	Estado da entrada digital 16 (0 = off e 1 = on)
20016	DI_17	WORD	R	Estado da entrada digital 17 (0 = off e 1 = on)
20017	DI_18	WORD	R	Estado da entrada digital 18 (0 = off e 1 = on)
20018	DI_N1	WORD	R	Estado da entrada contato seco N1 (0 = off e 1 = on)
20019	DI_N2	WORD	R	Estado da entrada contato seco N2 (0 = off e 1 = on)
20020	DI_N3	WORD	R	Estado da entrada contato seco N3 (0 = off e 1 = on)
20021	DI_N4	WORD	R	Estado da entrada contato seco N4 (0 = off e 1 = on)
20100	NTC_01.VAL	FLOAT	R	Valor da entrada N1 (NTC 1) (-50 a 130 °C)
20102	NTC_01.STAT	WORD	R	Falha NTC1 (1 = normal e 0 = falha)
20103	NTC_02.VAL	FLOAT	R	Valor da entrada N2 (NTC 2) (-50 a 130 °C)
20105	NTC_02.STAT	WORD	R	Falha NTC2 (1 = normal e 0 = falha)
20106	NTC_03.VAL	FLOAT	R	Valor da entrada N3 (NTC 3) (-50 a 130 °C)
20108	NTC_03.STAT	WORD	R	Falha NTC3 (1 = normal e 0 = falha)
20109	NTC_04.VAL	FLOAT	R	Valor da entrada N4 (NTC 4) (-50 a 130 °C)
20111	NTC_04.STAT	WORD	R	Falha NTC4 (1 = normal e 0 = falha)
20112	NTC_05.VAL	FLOAT	R	Valor da entrada analógica 1 (0~100.0%)
20114	NTC_05.STAT	WORD	R	Falha em AI1 (1 = normal e 0 = falha)
20115	NTC_06.VAL	FLOAT	R	Valor da entrada analógica 2 (0~100%)

20117	NTC_06.STAT	WORD	R	Falha em AI2 (1 = normal e 0 = falha)
20118	NTC_07.VAL	FLOAT	R	Valor da entrada analógica 3 (0~100%)
20120	NTC_07.STAT	WORD	R	Falha em AI3 (1 = normal e 0 = falha)
20121	NTC_08.VAL	FLOAT	R	Valor da entrada analógica 4 (0~100%)
20123	NTC_08.STAT	WORD	R	Falha em AI4 (1 = normal e 0 = falha)
20124	NTC_09.VAL	FLOAT	R	Valor da entrada analógica 9 (0-100%) / NTC 9
20126	NTC_09.STAT	WORD	R	Falha NTC9 (0 = válido)
20127	NTC_10.VAL	FLOAT	R	Valor da entrada analógica 10 (0-100%) / NTC 10
20129	NTC_10.STAT	WORD	R	Falha NTC10 (0 = válido)
20130	NTC_11.VAL	FLOAT	R	Valor da entrada analógica 11 (0-100%) / NTC 11
20132	NTC_11.STAT	WORD	R	Falha NTC11 (0 = válido)
20133	NTC_12.VAL	FLOAT	R	Valor da entrada analógica 12 (0-100%) / NTC 12
20135	NTC_12.STAT	WORD	R	Falha NTC12 (0 = válido)
20136	NTC_13.VAL	FLOAT	R	Valor da entrada analógica 13 (0-100%) / NTC 13
20138	NTC_13.STAT	WORD	R	Falha NTC13 (0 = válido)
20139	NTC_14.VAL	FLOAT	R	Valor da entrada analógica 14 (0-100%) / NTC 14
20141	NTC_14.STAT	WORD	R	Falha NTC14 (0 = válido)
20142	NTC_15.VAL	FLOAT	R	Valor da entrada analógica 15 (0-100%) / NTC 15
20144	NTC_15.STAT	WORD	R	Falha NTC15 (0 = válido)
20145	NTC_16.VAL	FLOAT	R	Valor da entrada analógica 16 (0-100%) / NTC 16
20147	NTC_16.STAT	WORD	R	Falha NTC16 (0 = válido)
20148	NTC_17.VAL	FLOAT	R	Valor da entrada analógica 17 (0-100%) / NTC 17
20150	NTC_17.STAT	WORD	R	Falha NTC17 (0 = válido)
20151	NTC_18.VAL	FLOAT	R	Valor da entrada analógica 18 (0-100%) / NTC 18
20153	NTC_18.STAT	WORD	R	Falha NTC18 (0 = válido)
20154	AI_1.VAL	FLOAT	R	Valor da entrada analógica 1 (0-100%)
20156	AI_1.STAT	WORD	R	Falha da entrada analógica 1 (0 = válido)
20157	AI_2.VAL	FLOAT	R	Valor da entrada analógica 2 (0-100%)
20159	AI_2.STAT	WORD	R	Falha da entrada analógica 2 (0 = válido)
20160	AI_3.VAL	FLOAT	R	Valor da entrada analógica 3 (0-100%)
20162	AI_3.STAT	WORD	R	Falha da entrada analógica 3 (0 = válido)
20163	AI_4.VAL	FLOAT	R	Valor da entrada analógica 4 (0-100%)
20165	AI_4.STAT	WORD	R	Falha da entrada analógica 4 (0 = válido)
20166	AI_5.VAL	FLOAT	R	Valor da entrada analógica 5 (0-100%)
20168	AI_5.STAT	WORD	R	Falha da entrada analógica 5 (0 = válido)
20169	AI_6.VAL	FLOAT	R	Valor da entrada analógica 6 (0-100%)
20171	AI_6.STAT	WORD	R	Falha da entrada analógica 6 (0 = válido)
20172	AI_7.VAL	FLOAT	R	Valor da entrada analógica 7 (0-100%)
20174	AI_7.STAT	WORD	R	Falha da entrada analógica 7 (0 = válido)
20175	AI_8.VAL	FLOAT	R	Valor da entrada analógica 8 (0-100%)
20177	AI_8.STAT	WORD	R	Falha da entrada analógica 8 (0 = válido)
20200	DO_01	WORD	R/W	Saída digital 1 (0 = off e 1 = on)
20201	DO_02	WORD	R/W	Saída digital 2 (0 = off e 1 = on)

20202	DO_03	WORD	R/W	Saída digital 3 (0 = off e 1 = on)
20203	DO_04	WORD	R/W	Saída digital 4 (0 = off e 1 = on)
20204	DO_05	WORD	R/W	Saída digital 5 (0 = off e 1 = on)
20205	DO_06	WORD	R/W	Saída digital 6 (0 = off e 1 = on)
20206	DO_07	WORD	R/W	Saída digital 7 (0 = off e 1 = on)
20207	DO_08	WORD	R/W	Saída digital 8 (0 = off e 1 = on)
20208	DO_09	WORD	R/W	Saída digital 9 (0 = off e 1 = on)
20209	DO_10	WORD	R/W	Saída digital 10 (0 = off e 1 = on)
20210	DO_11	WORD	R/W	Saída digital 11 (0 = off e 1 = on)
20211	DO_12	WORD	R/W	Saída digital 12 (0 = off e 1 = on)
20212	DO_13	WORD	R/W	Saída digital 13 (0 = off e 1 = on)
20213	DO_14	WORD	R/W	Saída digital 14 (0 = off e 1 = on)
20214	DO_15	WORD	R/W	Saída digital 15 (0 = off e 1 = on)
20215	DO_16	WORD	R/W	Saída digital 16 (0 = off e 1 = on)
<hr/>				
20300	AO_1	FLOAT	R/W	Saída analógica 1 (0~100.0)
20302	AO_2	FLOAT	R/W	Saída analógica 2 (0~100.0)
20304	AO_3	FLOAT	R/W	Saída analógica 3 (0~100.0)
20306	AO_4	FLOAT	R/W	Saída analógica 4 (0~100.0)
<hr/>				
20500	RTC.DAY	WORD	R	Relógio: dia (XX)
20501	RTC.MONTH	WORD	R	Relógio: mês (XX)
20502	RTC.YEAR	WORD	R	Relógio: ano (XXXX)
20503	RTC.HOUR	WORD	R	Relógio: hora (XX)
20504	RTC.MINUTE	WORD	R	Relógio: minuto (XX)
20505	RTC.SECOND	WORD	R	Relógio: segundo (XX)
20600	AJUSTE.DIA	WORD	R/W	Ajuste do relógio: dia (XX)
20601	AJUSTE.MES	WORD	R/W	Ajuste do relógio: mês (XX)
20602	AJUSTE.ANO	WORD	R/W	Ajuste do relógio: ano (XXXX)
20603	AJUSTE.HORA	WORD	R/W	Ajuste do relógio: hora (XX)
20604	AJUSTE.MIN	WORD	R/W	Ajuste do relógio: minuto (XX)
20605	AJUSTE.SEG	WORD	R/W	Ajuste do relógio: segundo (XX)
20606	AJUSTE.SALVA	WORD	R/W	Ajuste do relógio: escrever 12345 para alterar.
20610	Horário de verão	WORD	R/W	Dia de início do horário de verão (XX)
20611		WORD	R/W	Mês de início do horário de verão (XX)
20612		WORD	R/W	Dia de fim do horário de verão. (XX)
20613		WORD	R/W	Mês de fim do horário de verão. (XX)
20650	Feriado 01	WORD	R/W	Dia (XX)
20651		WORD	R/W	Mês (XX)
20652	Feriado 02	WORD	R/W	Dia (XX)
20653		WORD	R/W	Mês (XX)
...
20688	Feriado 20	WORD	R/W	Dia (XX)
20689		WORD	R/W	Mês (XX)

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.

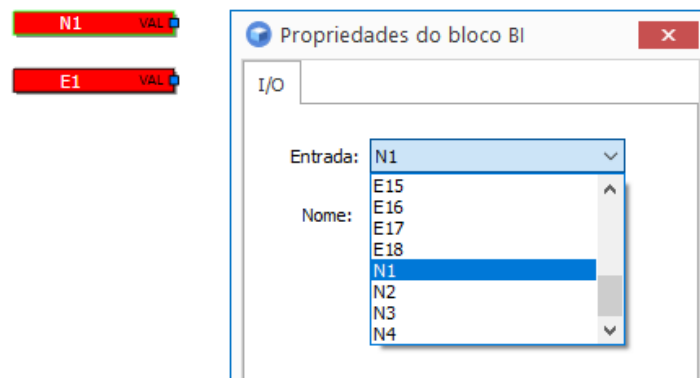
9.1. GRUPO IO – ENTRADAS E SAÍDAS DIGITAIS E ANALÓGICAS

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.

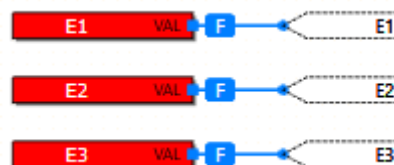
Entrada: ponto a ser monitorado / VAL: status Booleano referente a entrada (V/F).



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.



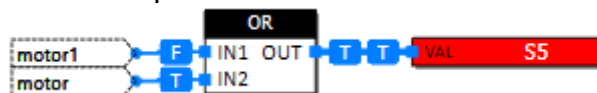
Outra vantagem da utilização dessa técnica é evitar atraso desnecessários, pois quando o diagrama identifica um bloco **BI**, a execução do programa é interrompida por **28us** para fazer a leitura física desta entrada.

BO – Saída digital

Permite acesso a uma saída digital do controlador.

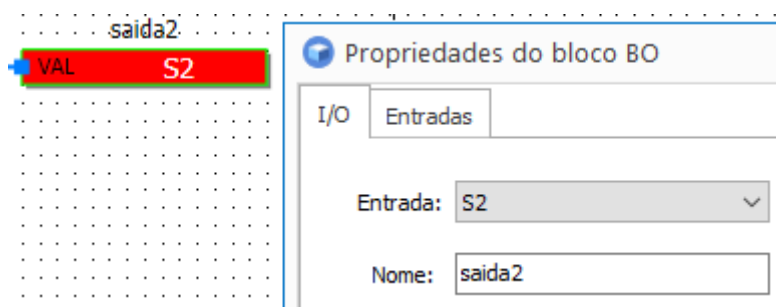
O valor das saídas é atualizado na memória do firmware (lógica -> firmware) sempre que um bloco BO/AO for executado. Esta atualização dispara uma tarefa no firmware que fará a atualização das saídas físicas. Não há sincronismo entre a execução desta tarefa e a lógica em si, mas o tempo é da ordem de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

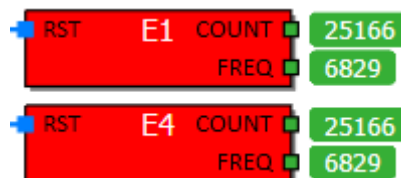
Permite acesso às 4 entradas contadoras de pulsos rápidos.

COUNT: informa o número de pulsos ocorridos.

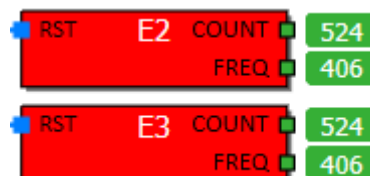
RST: zera o número de pulso de contagem.

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.

F RST E1 VAL

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.



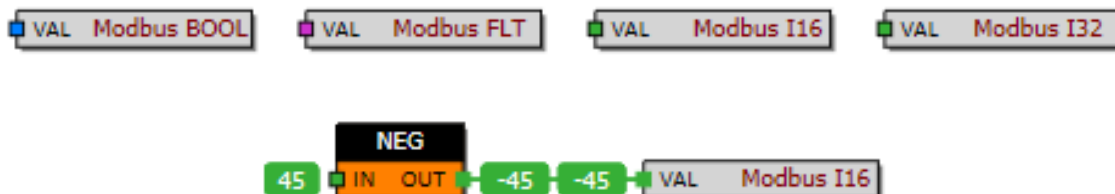
MDBS_IN_BOOL / MDBS_IN_FLOAT / MDBS_IN_I16 / MDBS_IN_I32

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.

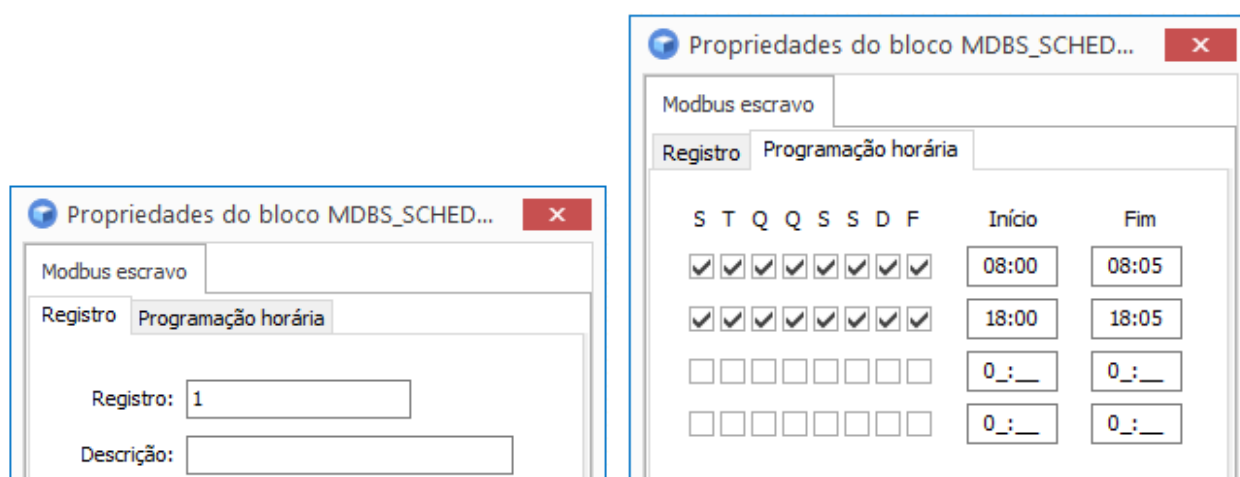


Obs: os blocos MDBS estão mais detalhados no capítulo 8.16.2.

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**.

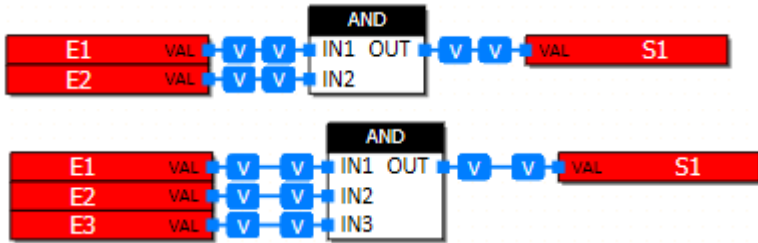
Exemplo: ativar a programação de segunda a sexta, das 08:00 às 18:00 pela IHM.
Registro 1 = **248** (seg ~ sex), registro 2 = **800** (8 h) e Registro 3 = **1800** (18 h).

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).



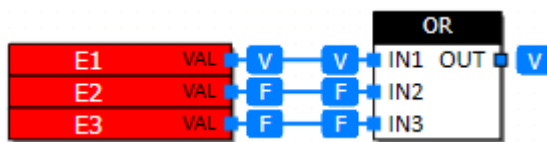
9.3. GRUPO LÓGICA - OPERAÇÕES COM AS ENTRADAS

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



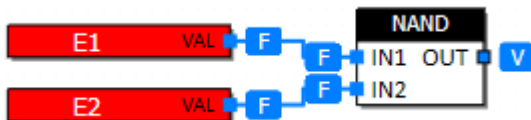
A	B	AND	AND3
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	0
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

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



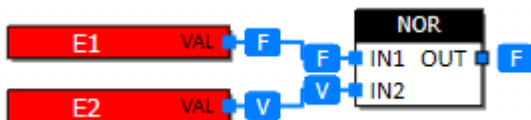
IN3	IN2	IN1	OUT
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

NAND – Operação NAND (E negado) de duas entradas



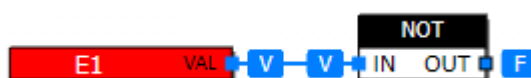
IN 2	IN 1	NAND
0	0	1
0	1	1
1	0	1
1	1	0

NOR – Operação NOR (ou negado) em duas entradas



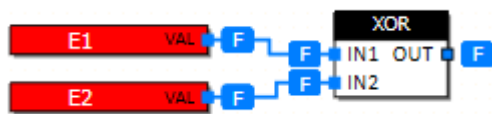
IN 2	IN 1	NOR
0	0	1
0	1	0
1	0	0
1	1	0

NOT – Operação NOT (negado)



IN 1	NOT
0	1
1	0

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

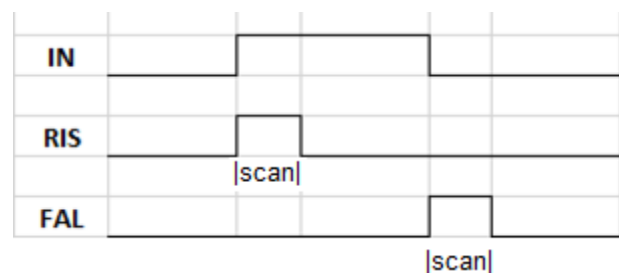


IN 2	IN 1	XOR
0	0	0
0	1	1
1	0	1
1	1	0

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

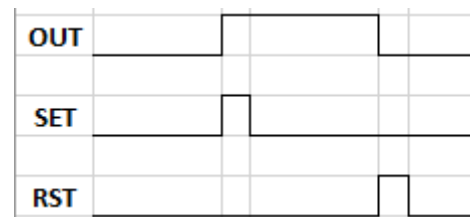
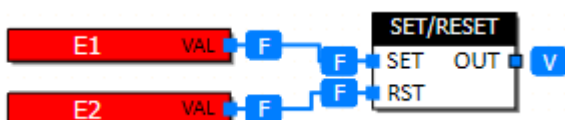
A borda de subida em IN aciona a saída RIS e a borda de descida em IN aciona a saída FAL.

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



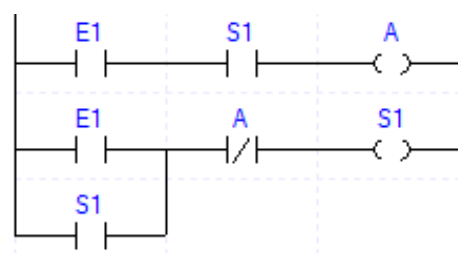
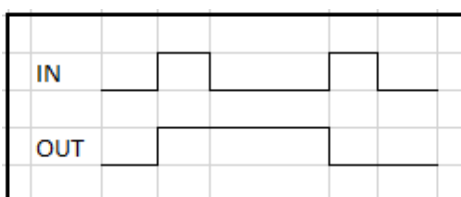
SET RESET – Controla o estado da saída por um pulso na entrada SET ou RST

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.



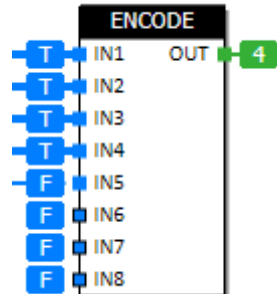
TOGGLE - Inverte o status da saída OUT a cada borda de subida em IN

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

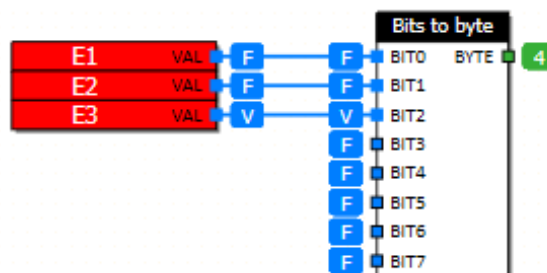
Esse bloco verifica a entrada ativa (1~8) mais significativa, sendo IN1 (menos) e IN8 (mais) e envia o número dessa entrada para OUT (INx mais significativa = OUT). Caso nenhuma entrada esteja ativa, OUT = 0.



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

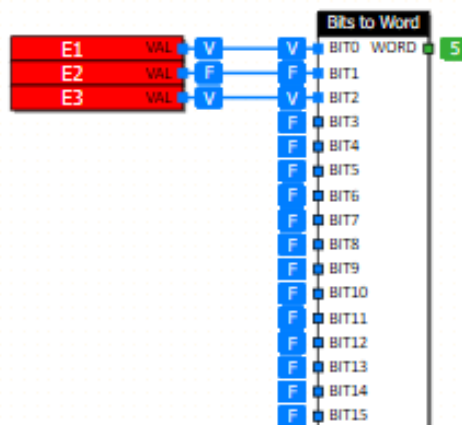
BITS_TO_BYTE – Conversor de Bits para Byte

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**.



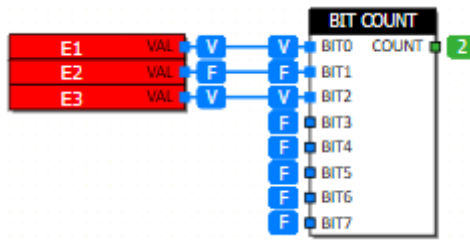
BITS_TO_WORD – Conversor de Bits para Word

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

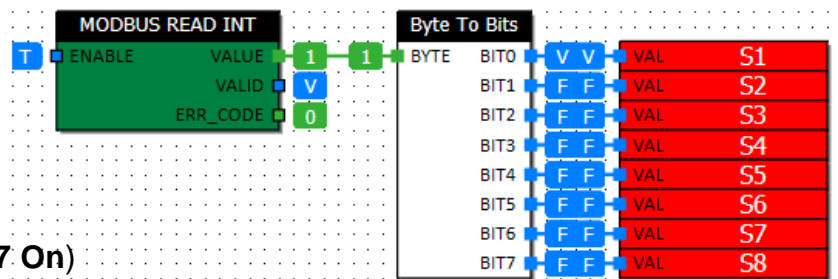
Verifica até 8 entradas ativas (BIT0 ~ BIT7) e gera o somatório em COUNT.



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



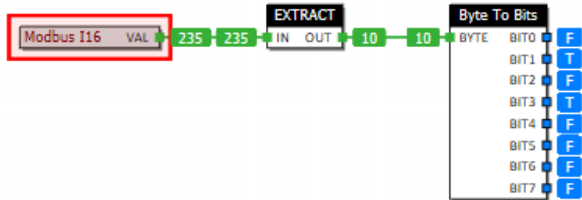
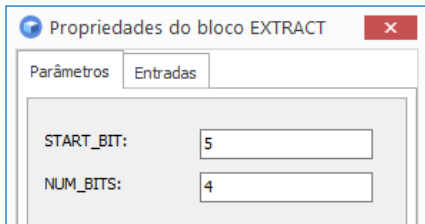
EXTRACT – Extrai bits de uma palavra

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).

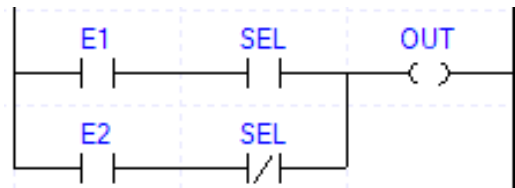
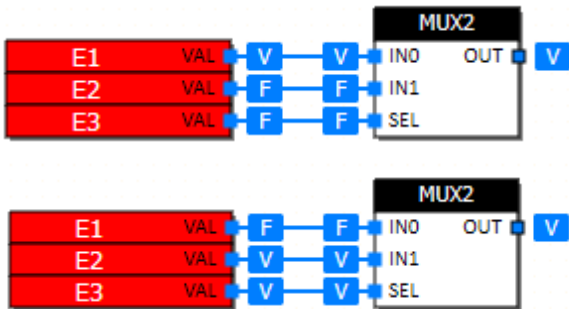
Ex: 235 = 11101011 (extrai 4 Bits a partir do Bit 5) = 10



MUX2 B - Multiplexador binário de 2 entradas

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

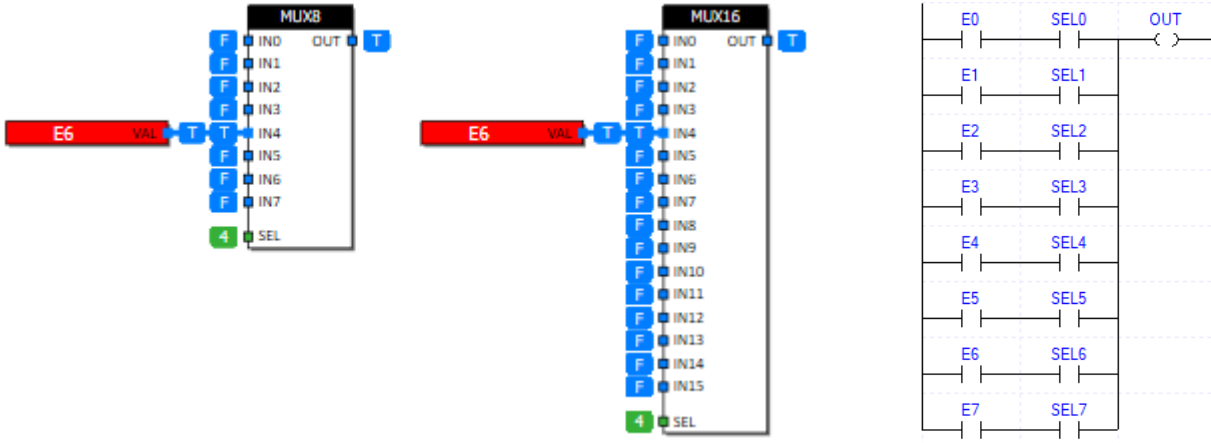
SEL = V, OUT recebe a cópia da entrada IN1.



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).

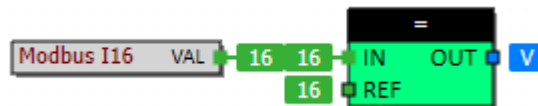


SEL	IN	.Out0	.Out1
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

9.4. GRUPO TESTE – COMPARAÇÕES

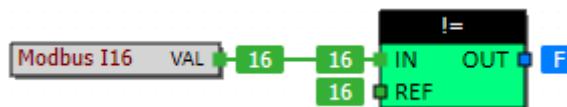
Os blocos com a terminação **_F** comparam números **reais** e os **_I** comparam números **inteiros**.

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



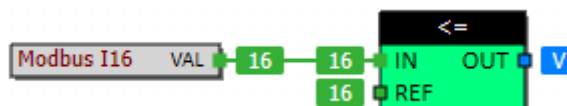
GREATER F / GREATER I - Verifica se o valor de entrada é maior que o valor de referência

Se $IN > REF$ ($OUT = V$)

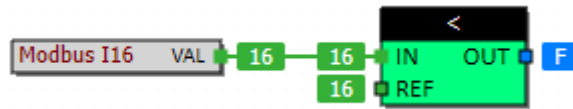


LOWEREQUAL F / LOWEREQUAL I – Verifica se o valor de entrada é menor ou igual ao valor de referência

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

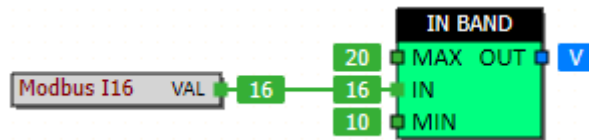


LOWER F / LOWER I – Verifica se o valor de entrada é menor que o valor de referência
Se $IN < REF$ ($OUT = V$).



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

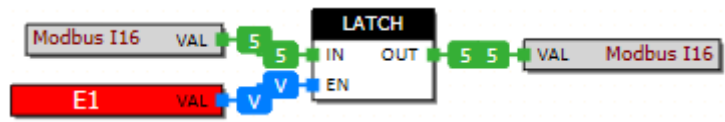
Se $IN \geq MIN$ e $IN \leq MAX$ ($OUT = V$)



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.



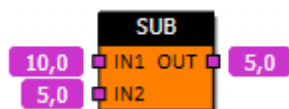
ADD2 F / ADD2 I / ADD3 F / ADD3 I / ADD4 F / ADD4 I – Soma os valores na entrada

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

$OUT = IN1 - IN2$

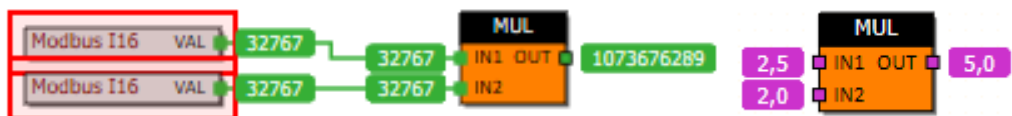


AND2 - Operação binária AND (E) em duas entradas

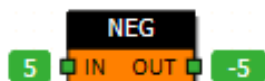


MUL2 F / MUL2 I – Multiplica dois valores de entrada

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

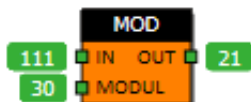


MOD_I - Calcula o resto da divisão do número inteiro IN por MOD

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.



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

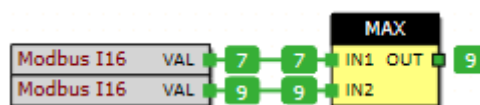
O maior valor atribuído a entrada IN é copiado para a saída MAX.

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

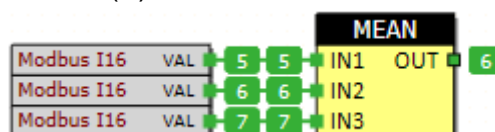
Copia para OUT o maior valor **presente** nas entradas IN1 ou IN2.



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.

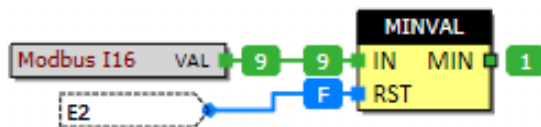
Obs: O valor de uma entrada falsa (F) não é incluso na média.



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.



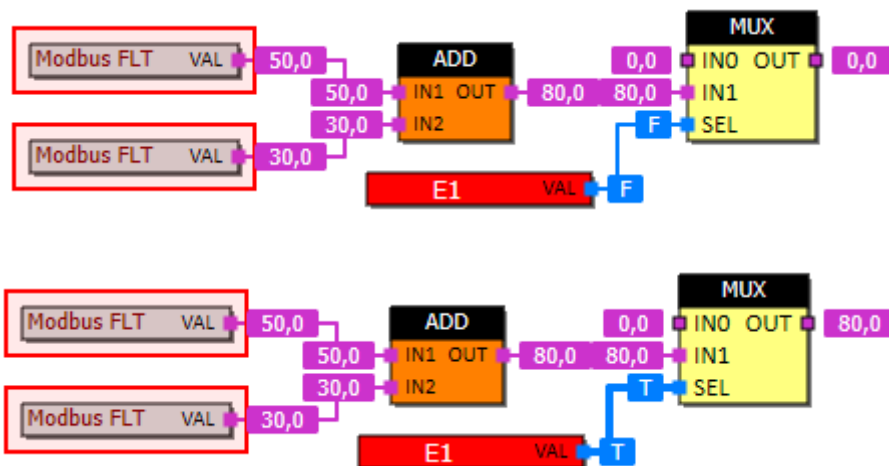
MIN F / MIN I – Copia para OUT a entrada de menor valor atual



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.

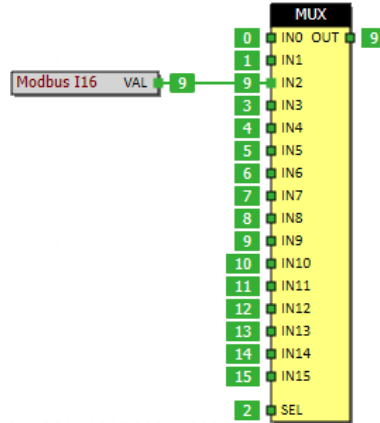


MUX3 F / MUX3 I / MUX4 F / MUX4 I / MUX8 F / MUX8 I / MUX16 F / MUX16 I

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.

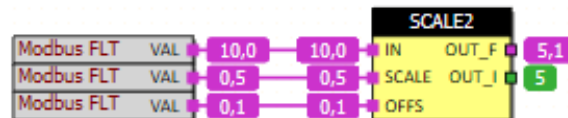
SCALE: Multiplicador a ser aplicado.

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.

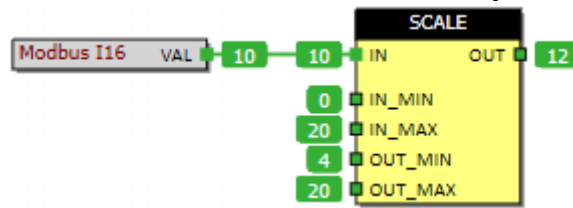
OUT_MIN: Valor de saída para o valor mínimo de entrada.

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.

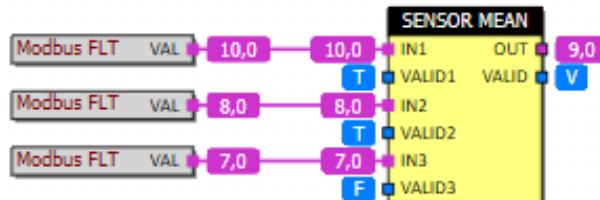


SENSOR_MEAN - Calcula a média aritmética em até 3 sensores

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

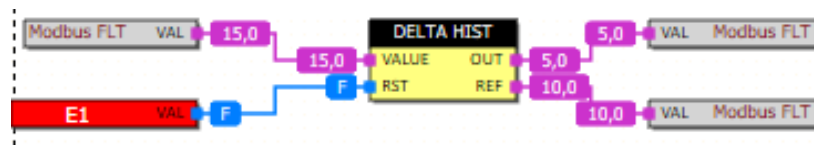
VALUE: valor para comparação (valor atual de um acumulador, por exemplo)

REF: valor atual de referência (retentivo)

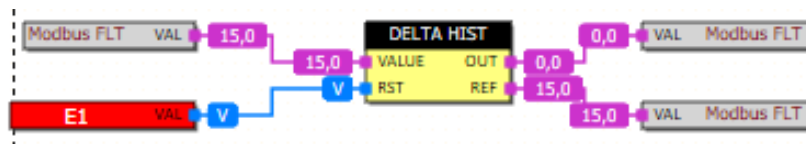
OUT: valor de saída (**OUT = VALUE - REF**)

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



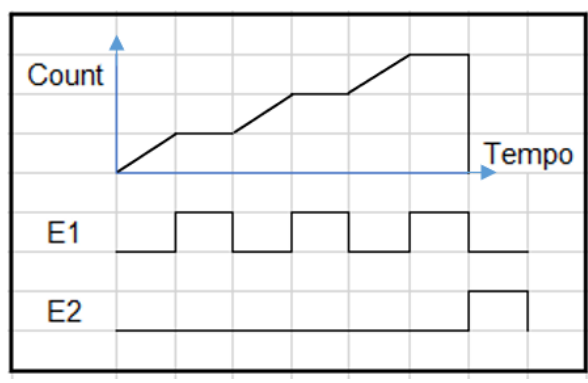
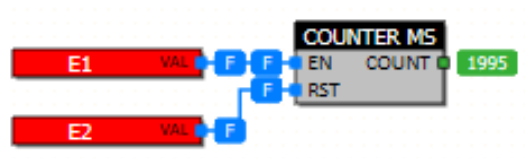
Diferença em relação a um valor atual: RST acionado



9.7. GRUPO TEMPORIZADORES

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

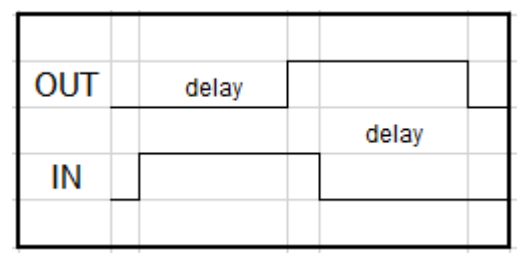
O tempo é acumulativo e armazenado na saída COUNT. Esse valor é zerado por falta de energia ou através de RST.



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

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

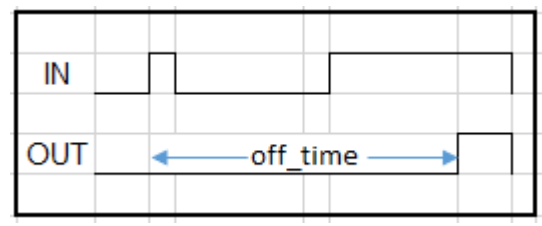
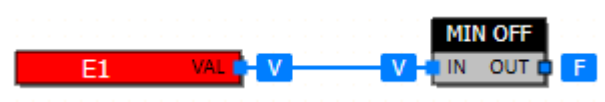
OUT: saída **temporizada para ligar e desligar**.



MIN OFF: Saída desligada por um tempo

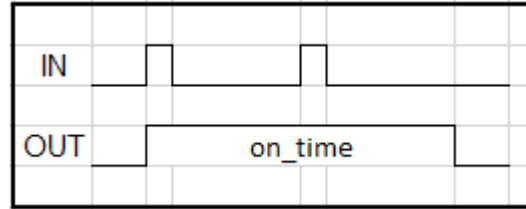
Após um pulso na entrada IN, OUT = F durante o tempo indicado em OFF-TIME (minutos), caso IN = V logo após o tempo OFF_TIME, OUT = V enquanto IN = V.

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.

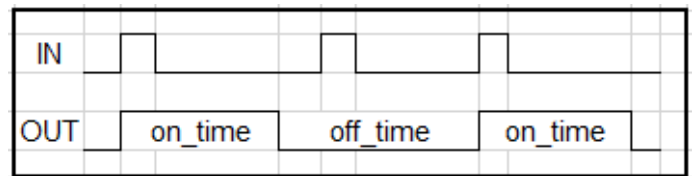


MIN ONOFF - Controla os tempos mínimos para cada estado da saída

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

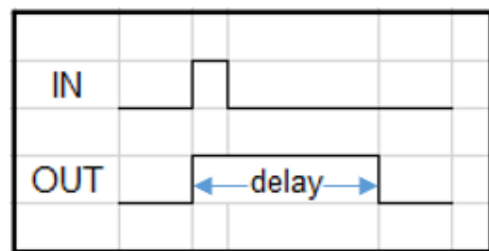
Ex: tempo ON_TIME = 1 e OFF_TIME = 2.

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.

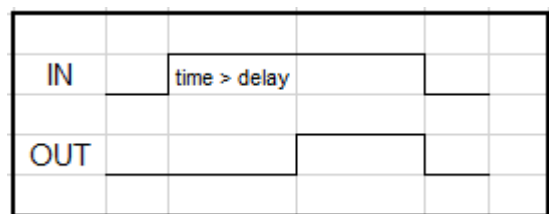


OFF DELAY / OFF_DELAY2 - Gera um atraso (décimo de segundo) para desligar a saída

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



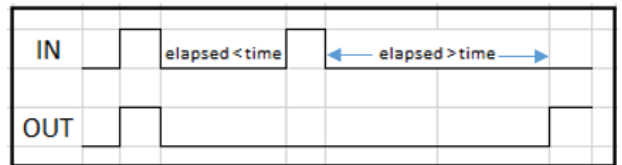
ON DELAY / ON_DELAY2 - OUT liga após IN permanecer acionado durante o tempo definido em DELAY



TOFF MS / TOFF S –Temporizador para desligar

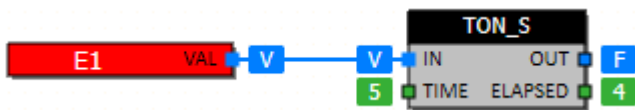
Uma borda de descida em IN, OUT = F e o temporizador conta até o tempo “TIME” (milissegundos ou segundos). Após o valor da temporização, OUT = V até receber um novo pulso em IN.
Um novo acionamento em IN reinicia o tempo acumulado.

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



TON MS / TON S - Temporizador para ligar

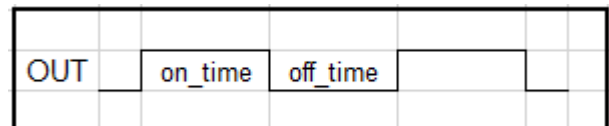
Inicia a temporização se IN = V e após o tempo TIME, OUT = V.
Caso IN seja desativada, o tempo é reiniciado.
TIME: milissegundos ou segundos.
ELAPSED: tempo decorrido.



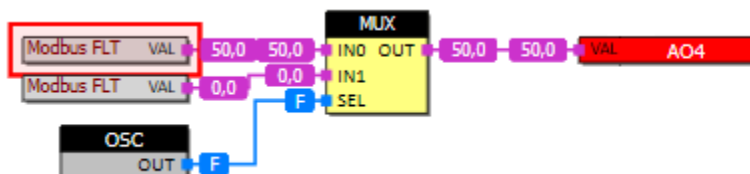
OSCILLATOR - Oscilador com tempo liga/desliga configurável (décimos de segundo)

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.

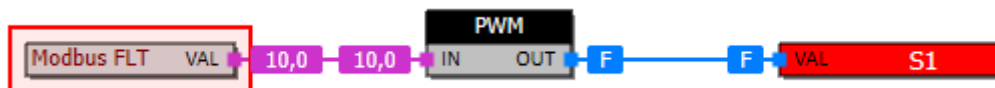
A largura do pulso é definida em PULSE_LENGTH (décimos de segundo).
Caso ocorra uma nova borda de subida em IN durante o pulso de saída, o tempo é reiniciado e o pulso é estendido.



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).

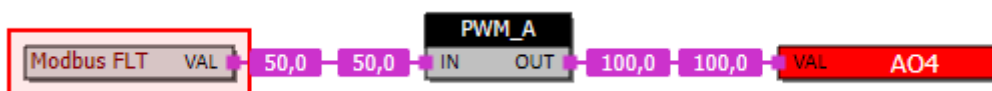


PWM_A – Saída analógica PWM

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

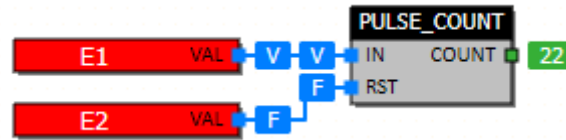
Este bloco conta as transições (bordas de subida) nas entradas UP e DOWN.

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.



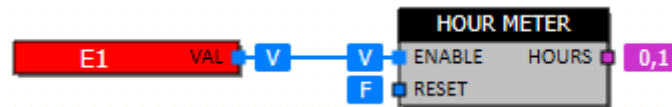
HOUR_METER1 / HOUR_METER2 – Horímetro

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**.

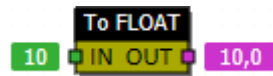
HOUR_METER2: valor de **saída inteiro**.

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

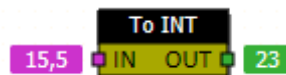


TOINT – Conversor de real para inteiro com escala

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

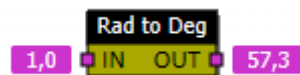
OUT = $15,5 * 1,5$ (escala) = 23,25 > eliminando a parte fracionária OUT = 23.



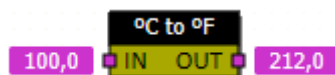
DEG TO RAD – Conversor angular (Graus para Radiano)



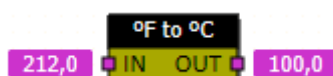
RAD TO DEG – Conversor angular (Radiano para Graus)



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



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

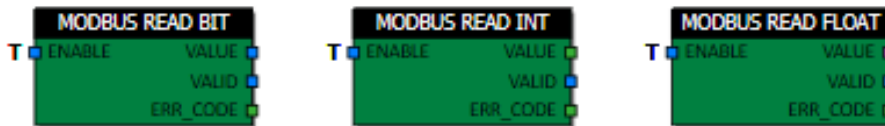


9.9. MDBM – BLOCOS DE COMUNICAÇÃO MODBUS MESTRE

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

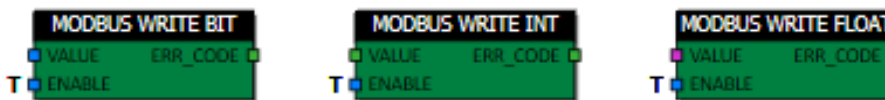
MDBM READ BIT / MDBM READ INT / MDBM READ FLOAT

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



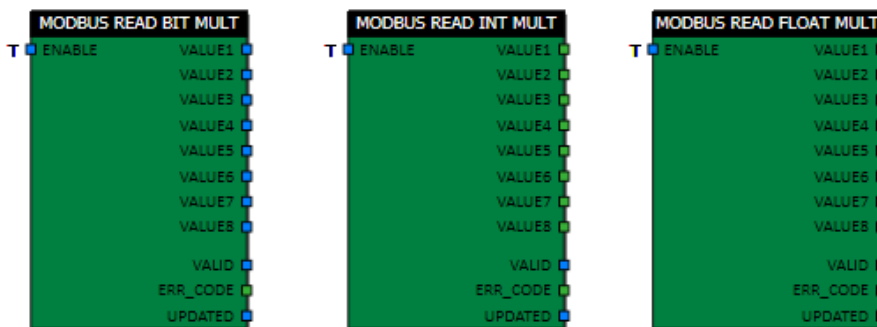
MDBM WRITE BIT / MDBM WRITE INT / MDBM WRITE FLOAT

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		

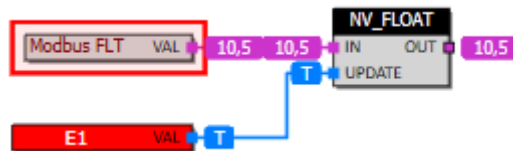
9.10. GRUPO DIVERSOS

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.

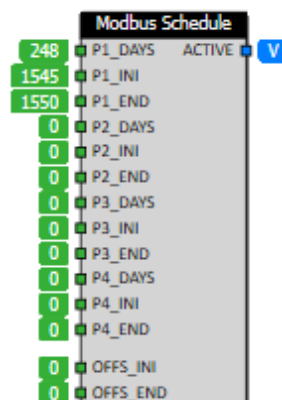
- **Px_Days:** Dias da semana (Bit 7 = segunda, Bit 6 = terça ... bit 0 = feriados)
Em binário: **11111000** (seg ~ sex)
Convertendo para decimal: **248**
- **Px_INI:** Horário de início
- **Px_END:** Horário de fim

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:

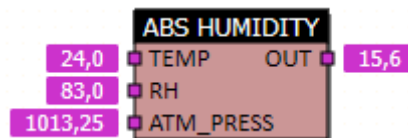
$$PS = 6.116441 \times 10^{\frac{7.591386 \times TEMP}{TEMP + 240.7263}}$$

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.

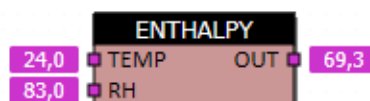
A fórmula utilizada para cálculo é:

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

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

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

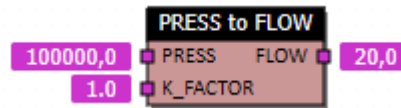
OUT: Entalpia do ar, em kJ/kg.



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.

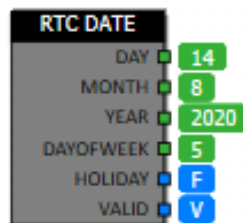
K_FACTOR: Fator K, sem unidade.

FLOW: Vazão, em m³/h.

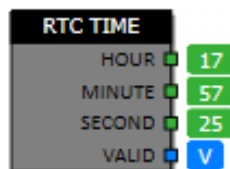
9.12. GRUPO SISTEMA

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 **Ex** pode estar ativo de cada vez e a passagem de um estado para outro é feito pelo estado lógico da transição **TRx**.

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.

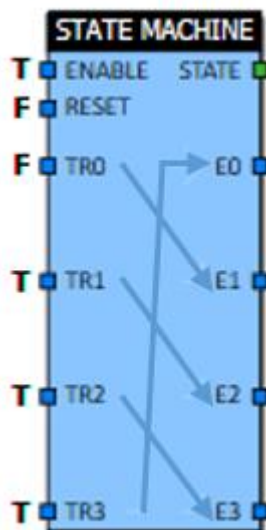
RESET: se ativada, todas as saídas **Ex** serão desligadas e o estado atual retorna ao **E0**.

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.

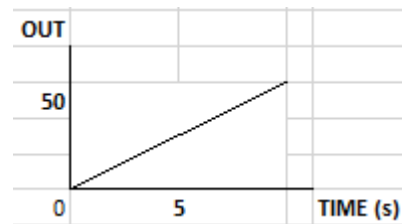
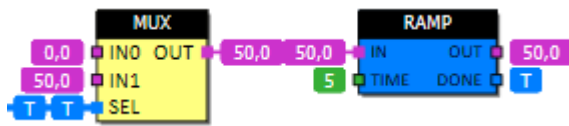
Clique [aqui](#) para ver mais exemplos do uso desse bloco no capítulo 11.

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%.

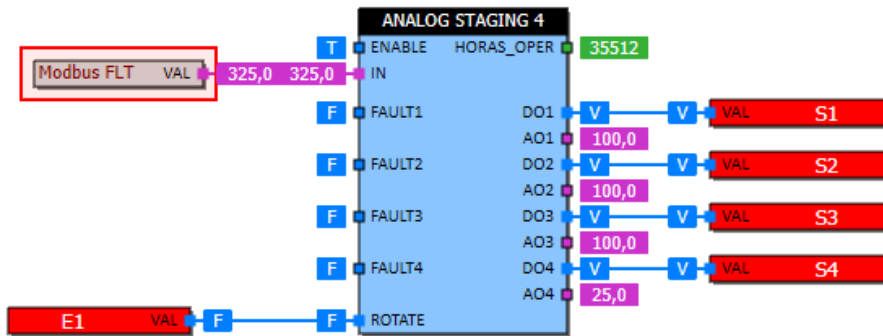
Configurações

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.

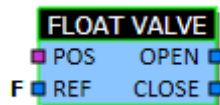
POS: Posição desejada da válvula.

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.

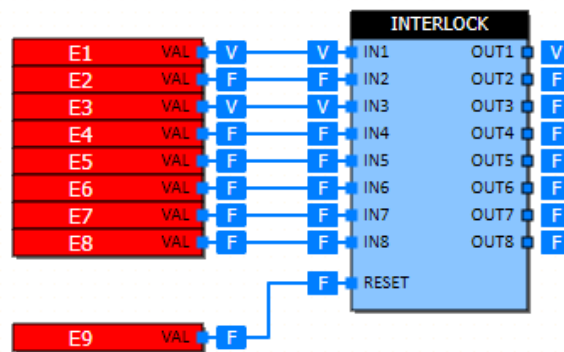
INx: Cada entrada controla o estado da saída correspondente.

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.

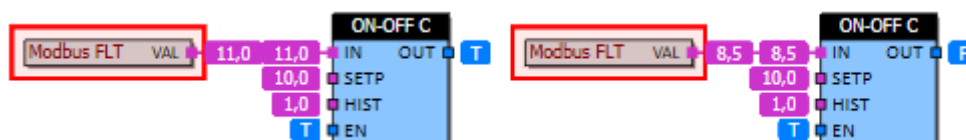
SETP: *Set point* para a variável de controle.

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.

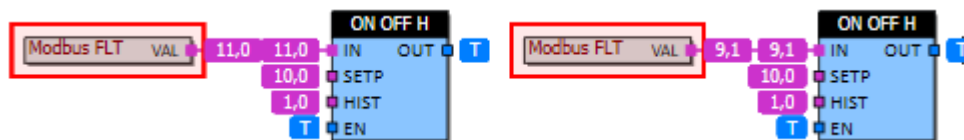
SETP: *Set point* para a variável de controle.

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 + HISTERESE1

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).

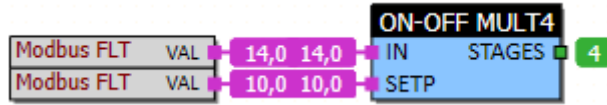
Ex: Com os 4 valores de histerese = 1 e MODE = 0 (refrigeração), os valores de entrada acima de SP teremos os seguintes estágios acionados:

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.

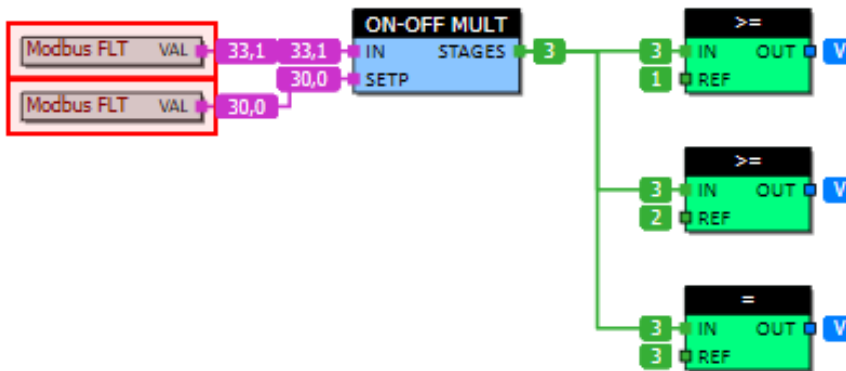
Saídas:

STAGES: Número de estágios que devem ser acionados.

Configurações:

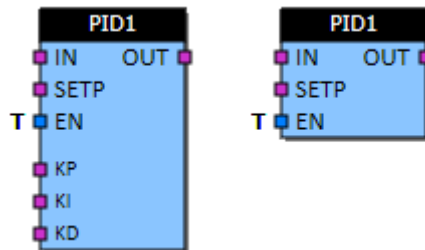
HISTERESE1: Ponto de acionamento do primeiro estágio. É ativado acima de $SETP + HISTERESE1$ e desativado abaixo de SETP.
HISTERESE2: Ponto de acionamento do segundo estágio. É acionado acima de $SETP + HISTERESE1 + HISTERESE2$ e desativado abaixo de $SETP + HISTERESE1$.
HISTERESE3: Ponto de acionamento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

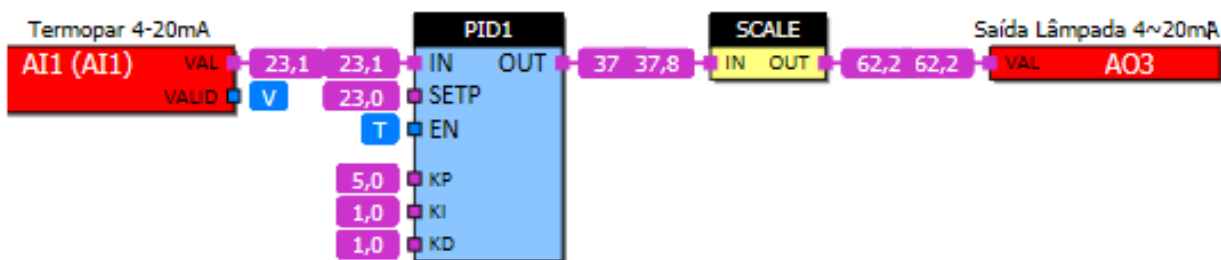
OUT_MIN: Valor mínimo da saída do PID.

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.

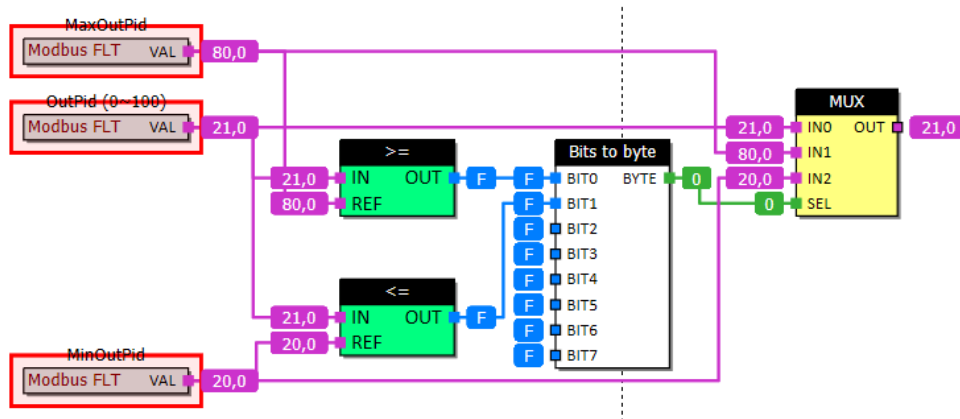
KP: Ganho proporcional do controlador PID.

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).

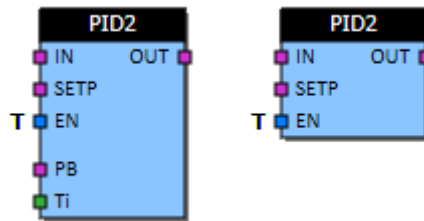
Ti: Tempo de integração (x 100ms). Este parâmetro define o tempo que o controlador repete a parcela proporcional na saída do PID. Tempos maiores tornam a contribuição integral mais lenta.

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.

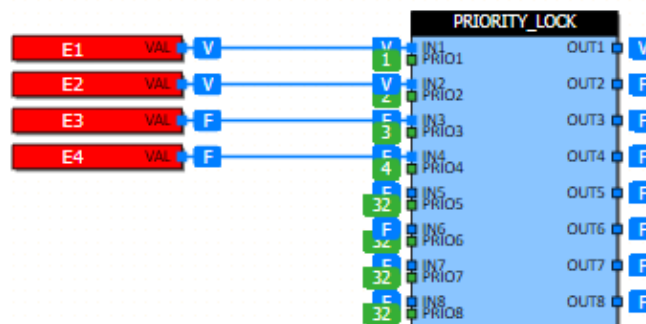
A entrada com maior prioridade (menor valor de PRIOx) define qual a saída que será acionada.

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.

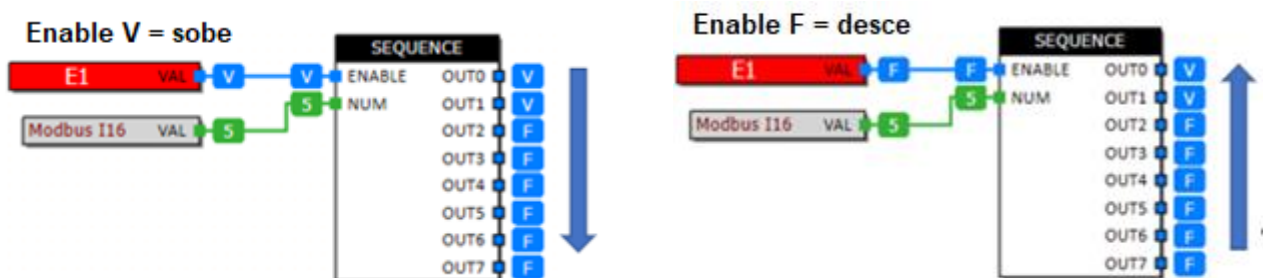
NUM: Define o número de saídas que devem ser ligadas.

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)

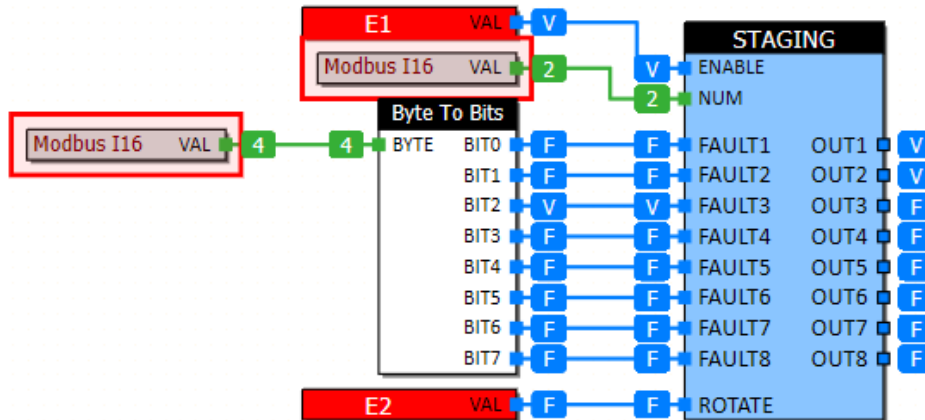
Configurações:

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.

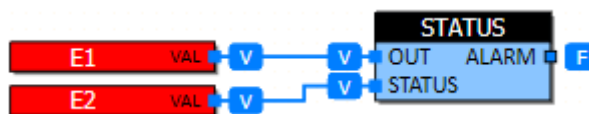
STATUS: Estado da entrada de monitoração.

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) ↩ ➡

Address : 3x

New Numeric Object

General Data Entry Format Security Shape Font

Display

Device data format : 16-bit Unsigned Mask

Display format

Type : Numeric

Left of decimal Pt. : Right of decimal Pt. :

Modbus I16 VAL

Propriedades do bloco MDBS_IN_I16

Modbus escravo

Registro

Registro:

Descrição:

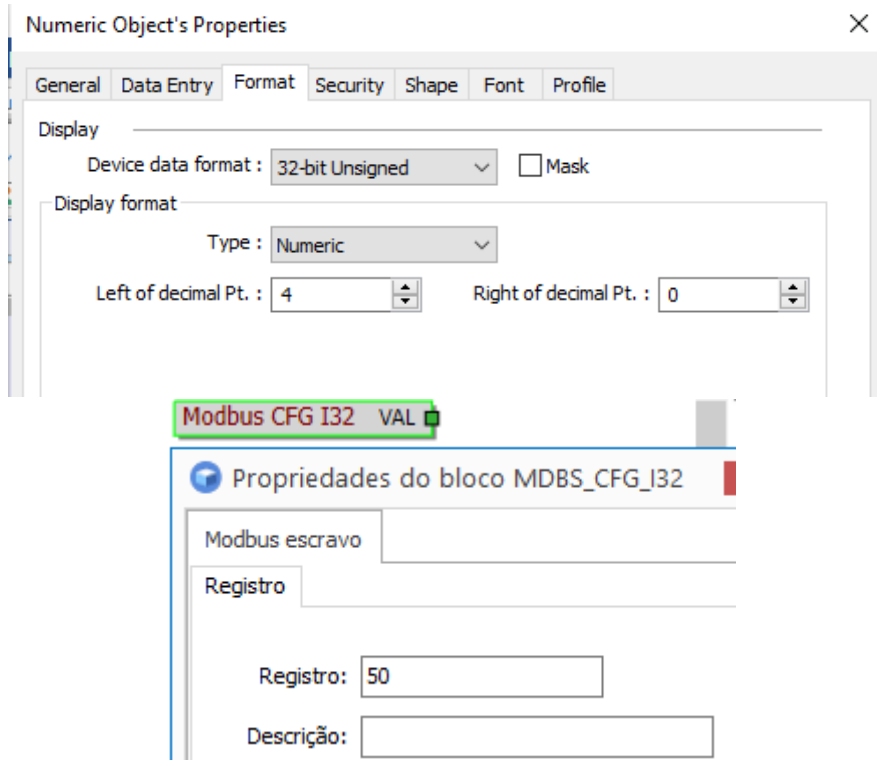
Ler/escrever em I32 (-2147483647 a 2147483647) (função = 5x)

Em formato, altere para 32-bit signed.

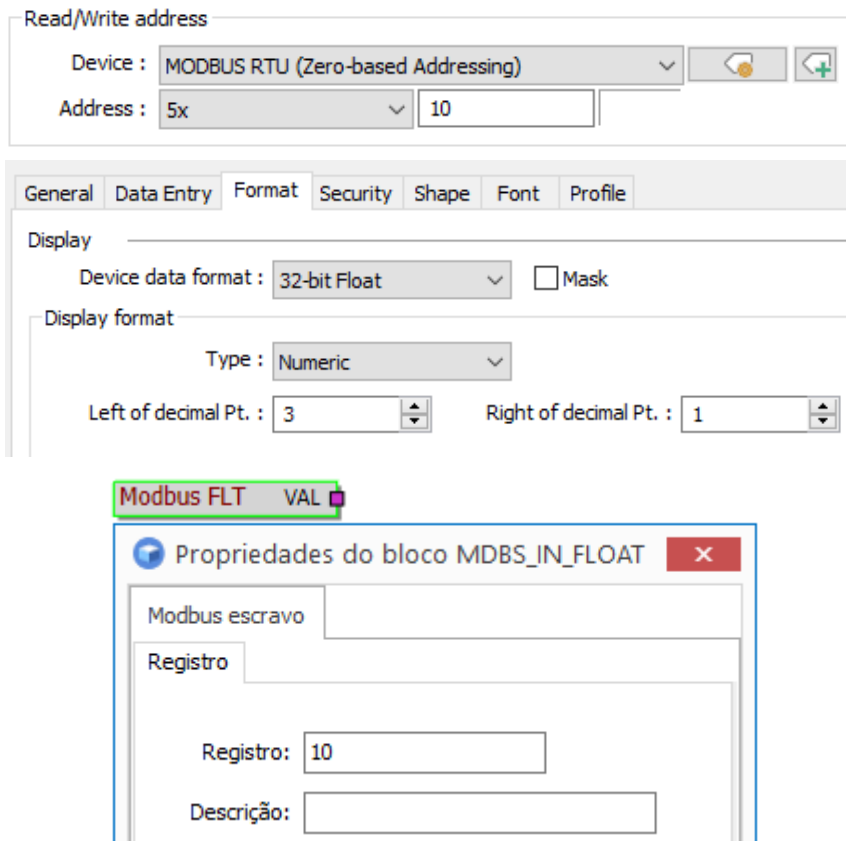
Read/Write address

Device : MODBUS TCP/IP (Zero-based Addressing) ↩ ➡

Address : 5x



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

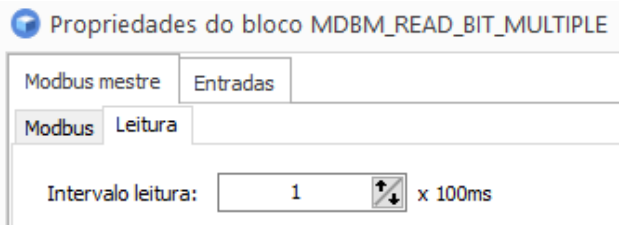
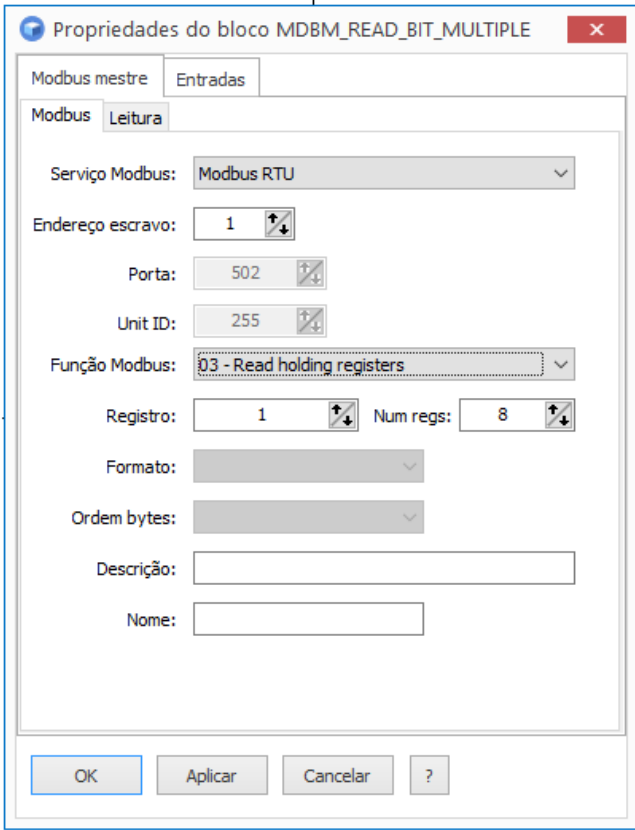
Porta RS485:

Velocidade:

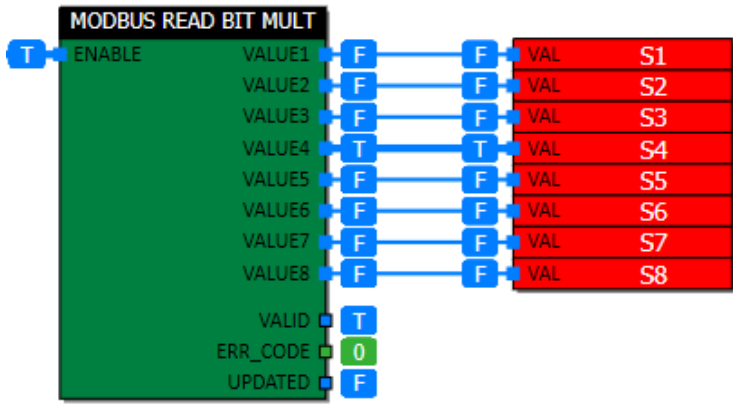
Paridade:

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



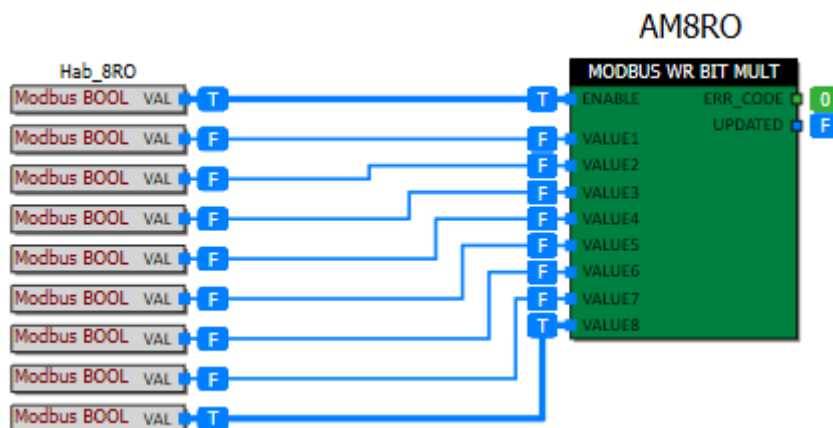
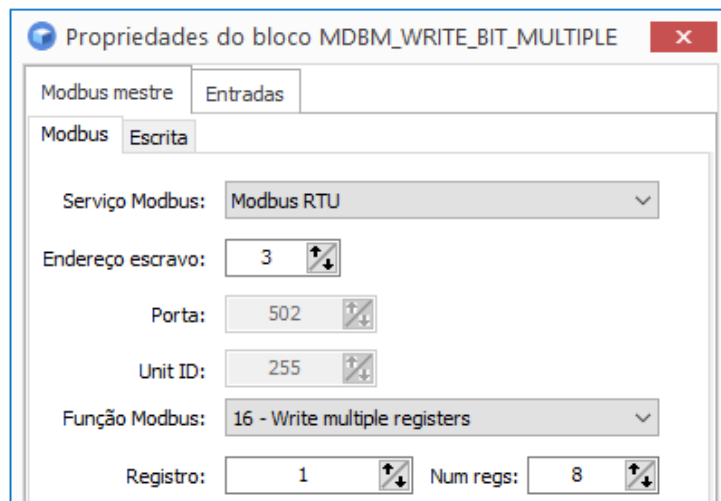
Exemplo utilizando TCP46 + AM8RO:

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

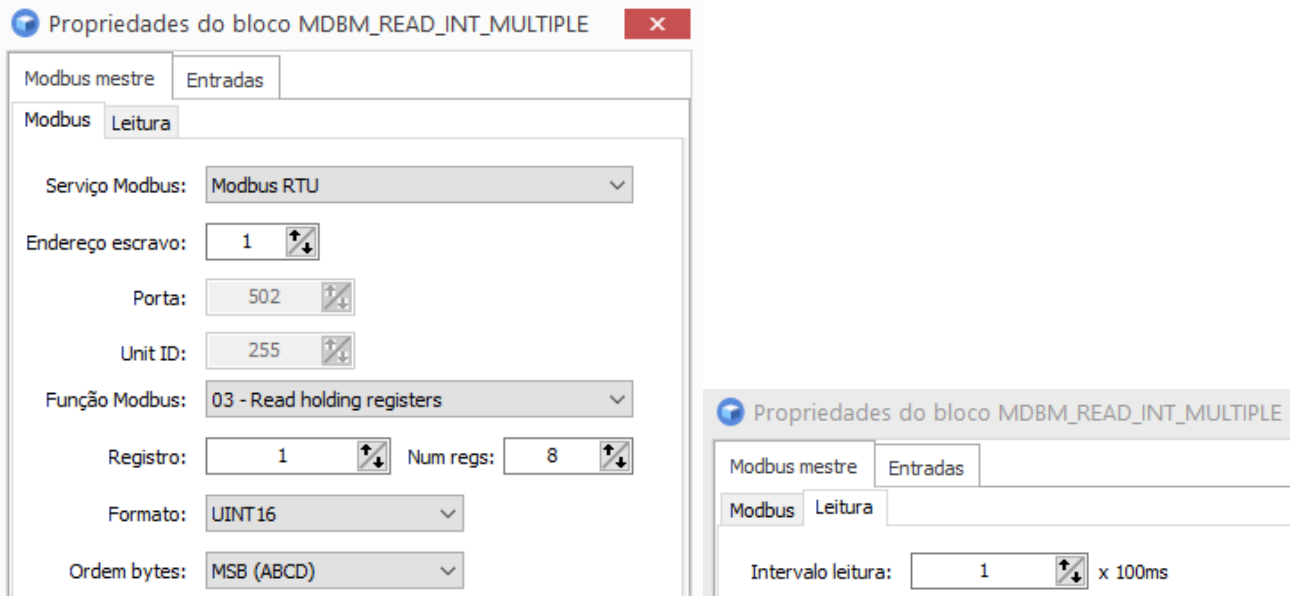
Posição	Protocolo
0	LG Inverter
1	Modbus RTU

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

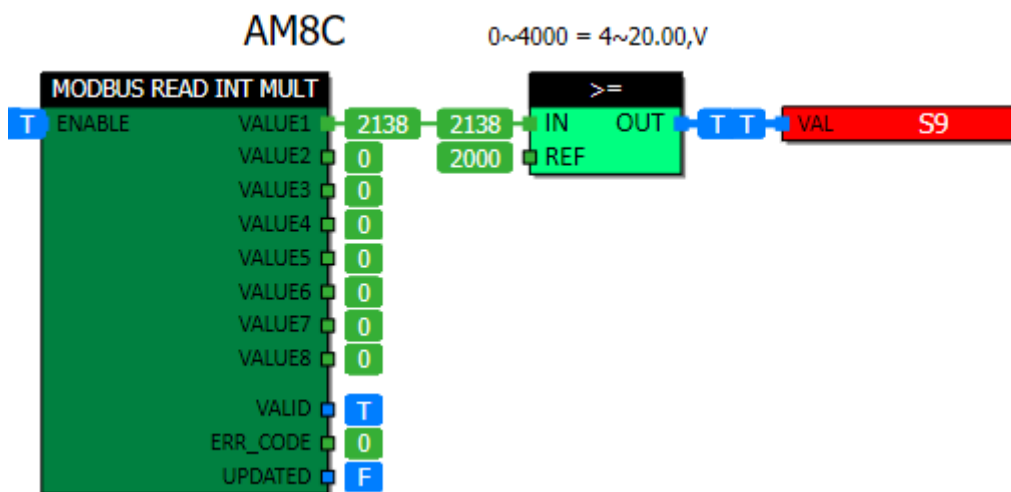
Posição	Endereço
0000	0*
0001	1
0010	2
...	...
1111	15

*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.

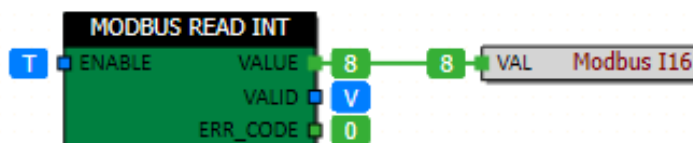
Conexão de rede:

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.

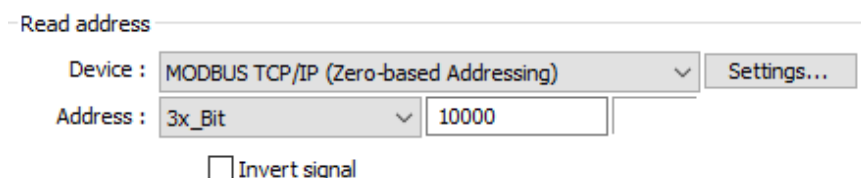


Convertendo esse valor decimal para binário é possível ver qual entrada do módulo está acionada:

- 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
<input type="text" value="30.00"/>	<input type="text" value="5.00"/>
Comando	DEC
<input type="text" value="3"/>	<input type="text" value="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.

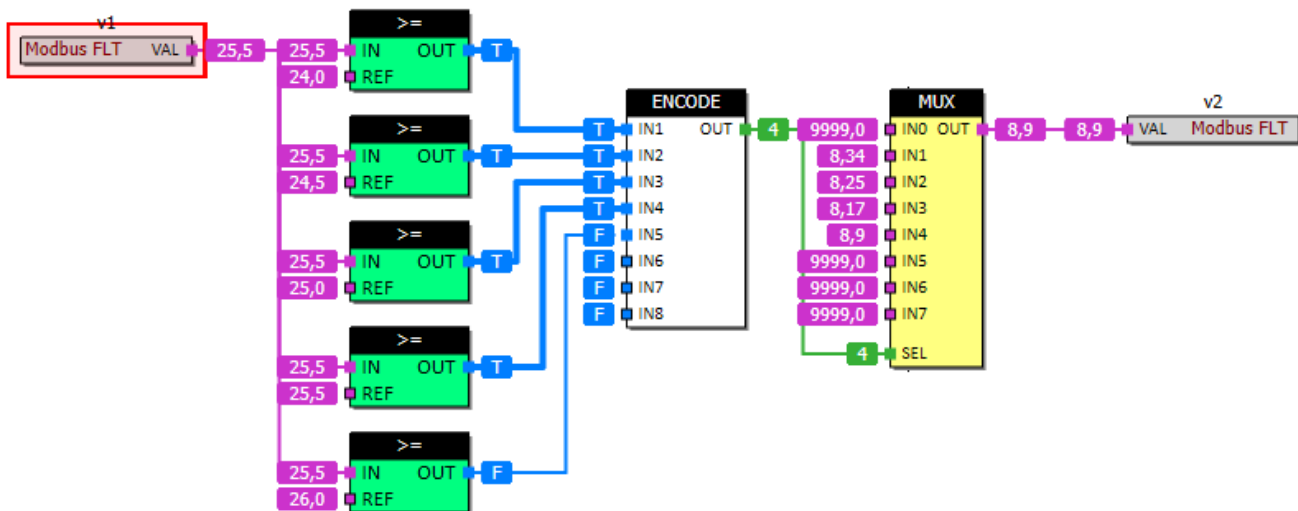
10.6. Bloco ENCODE para comparação de faixa de valores

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.

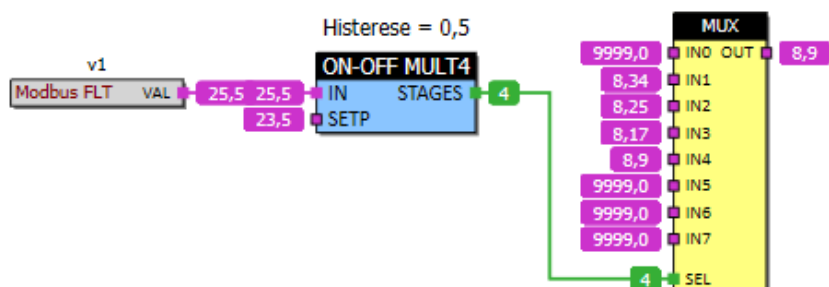
No exemplo são feitas 4 comparações dentro dos limites de 24 a 26 Vdc (faixas de 0,5 Vdc) e caso o valor de entrada esteja fora dos limites, saturamos a saída (Out = 9999).



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.



10.8. 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 conectado** 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 conectado 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

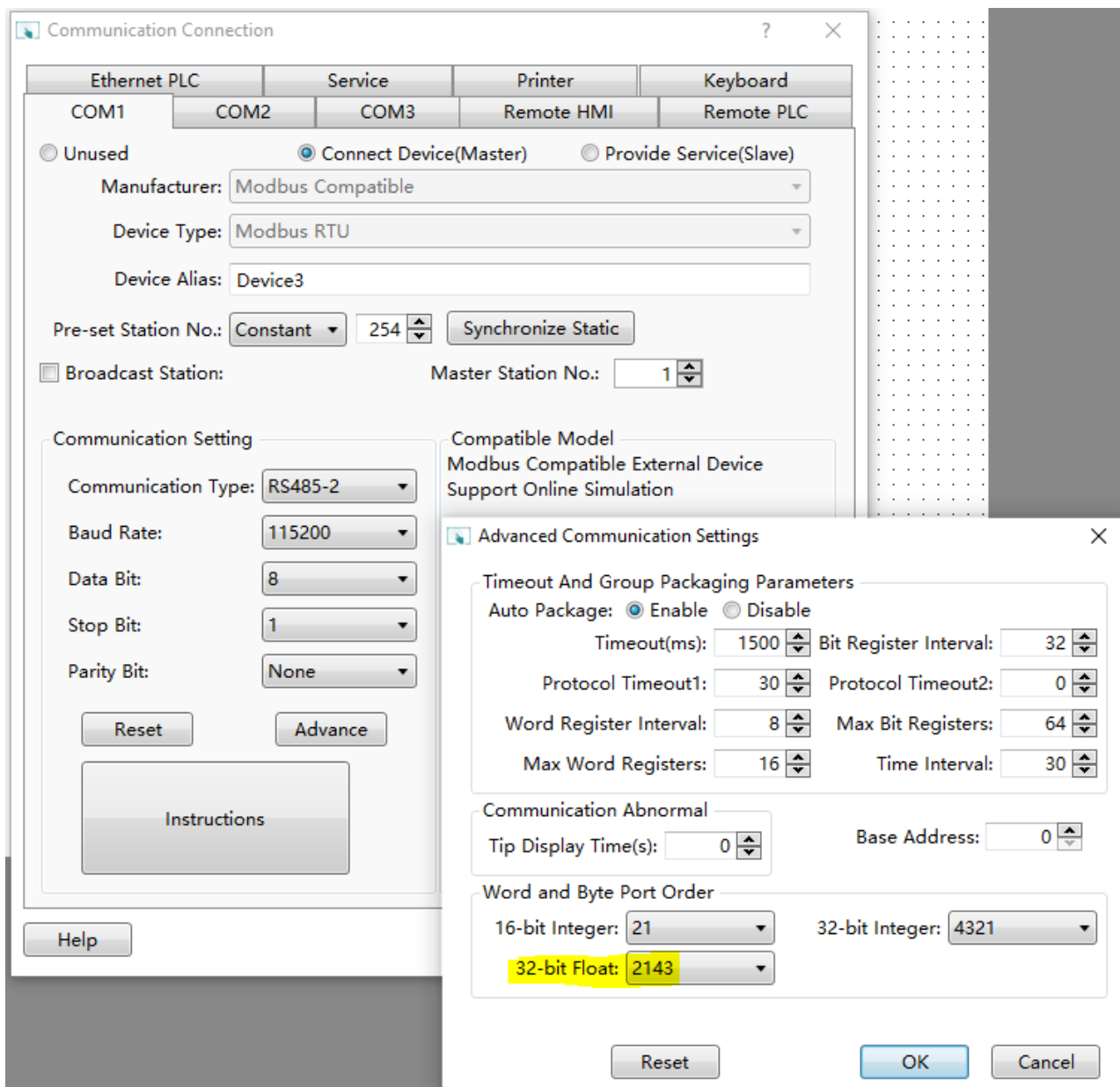
A comunicação entre a IHM Veichi Vi20 com o CLP Technolog TCP46 é possível usando o protocolo Modbus RTU e meio físico RS485.

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:

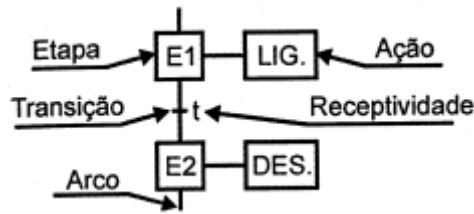
Atraso transmissão: ms

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.



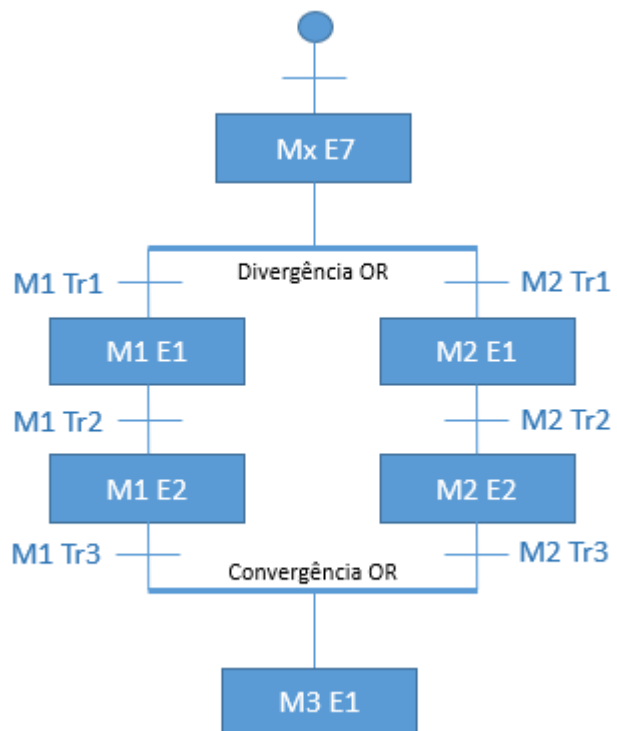
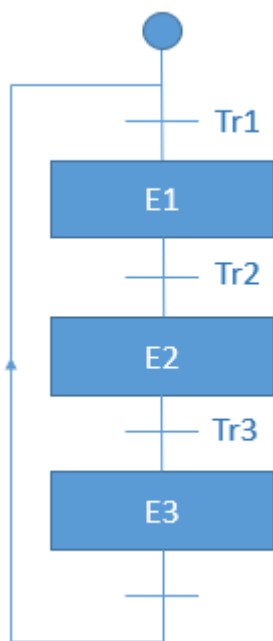
- Sequências

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.



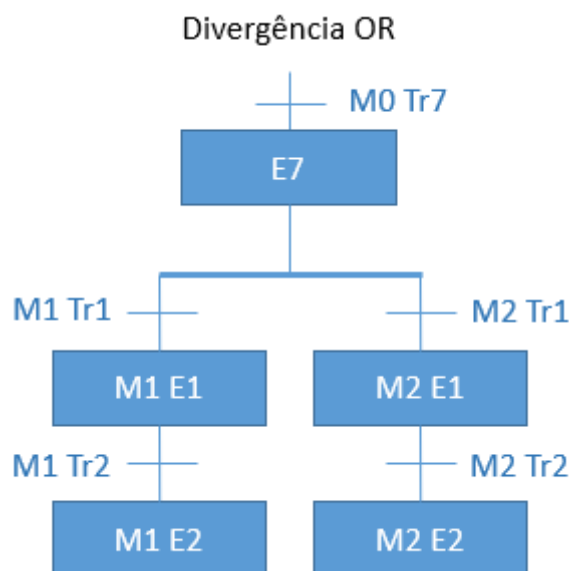
Estruturas lógicas

Divergência OR

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

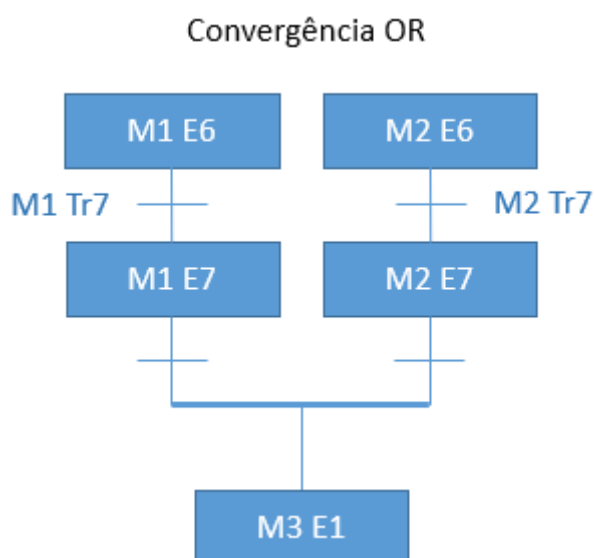
A máquina M1 E1 ativa se: Mx E7 ativa, e satisfaça a receptividade da transição M1 Tr1.

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.



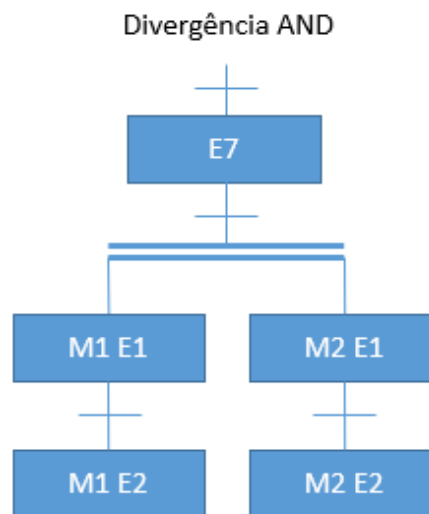
Paralelismo e sincronicidade

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

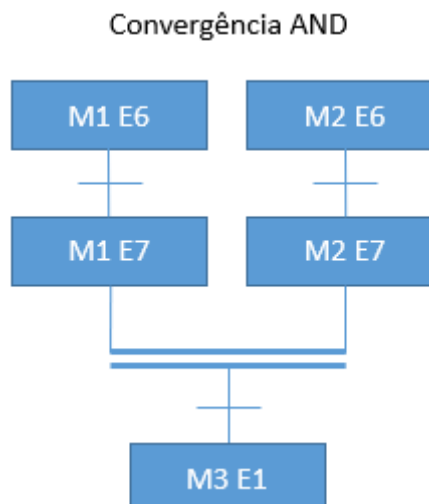
Uma divergência simultânea é precedida por uma transição e sucedida por sequências iniciadas por etapas.



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)

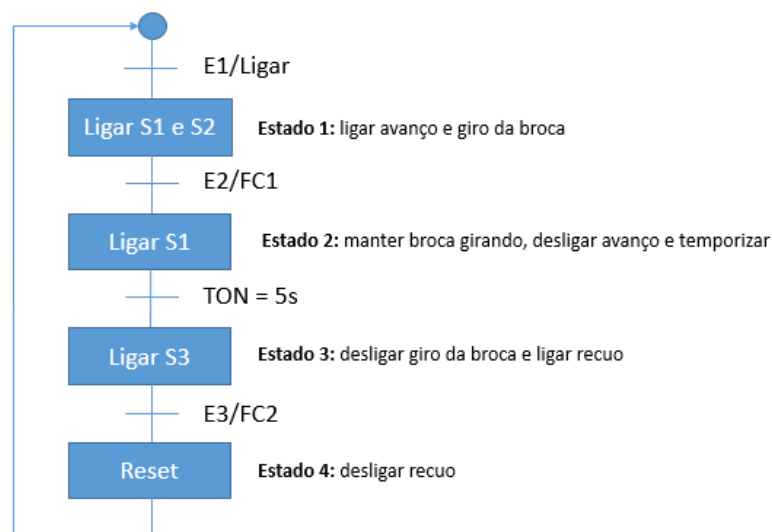
Saídas:

S1: Giro da broca

S2: Desce o conjunto mecânico

S3: Sobe o conjunto mecânico

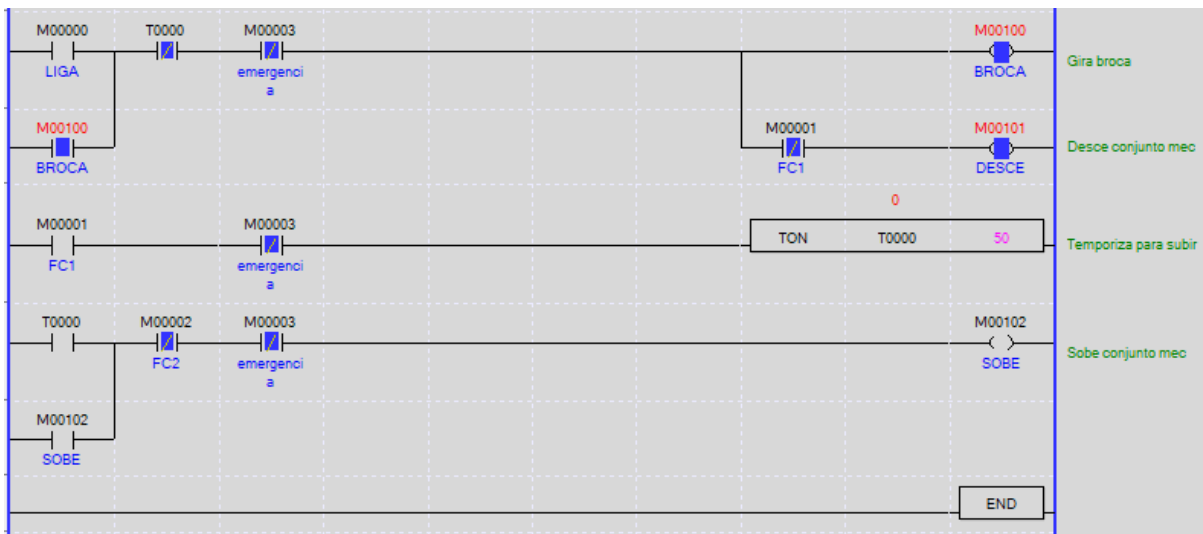
Máquina de estados da furadeira automática:



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.

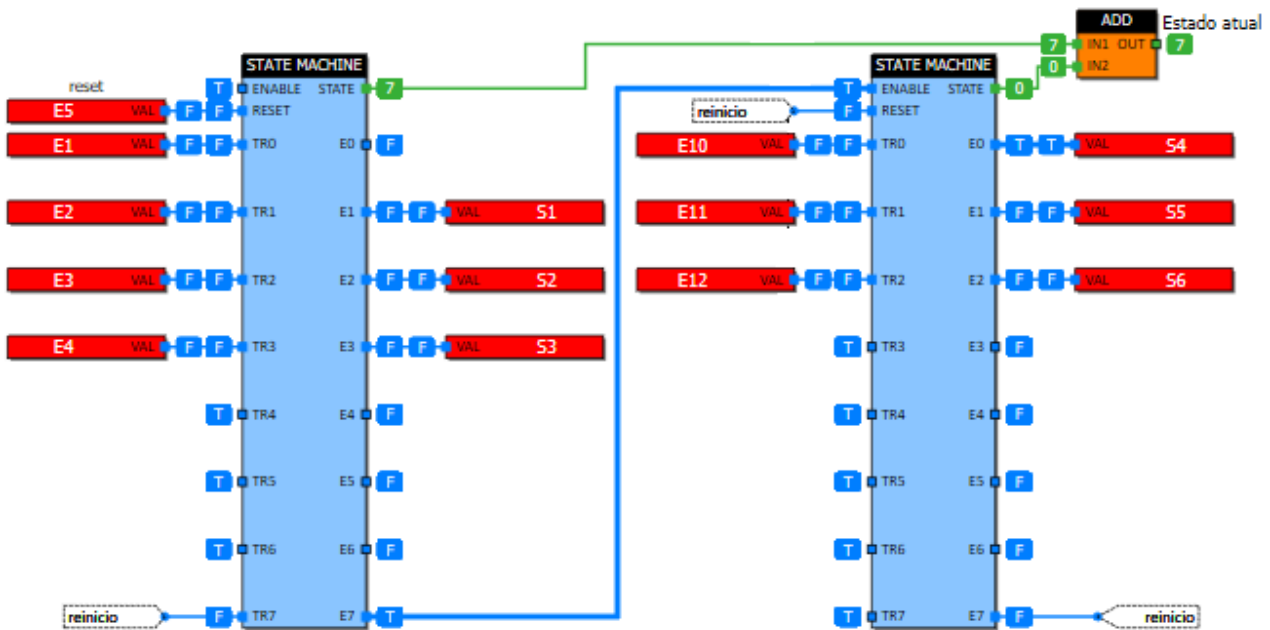
Máquina de Estados em Série

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.

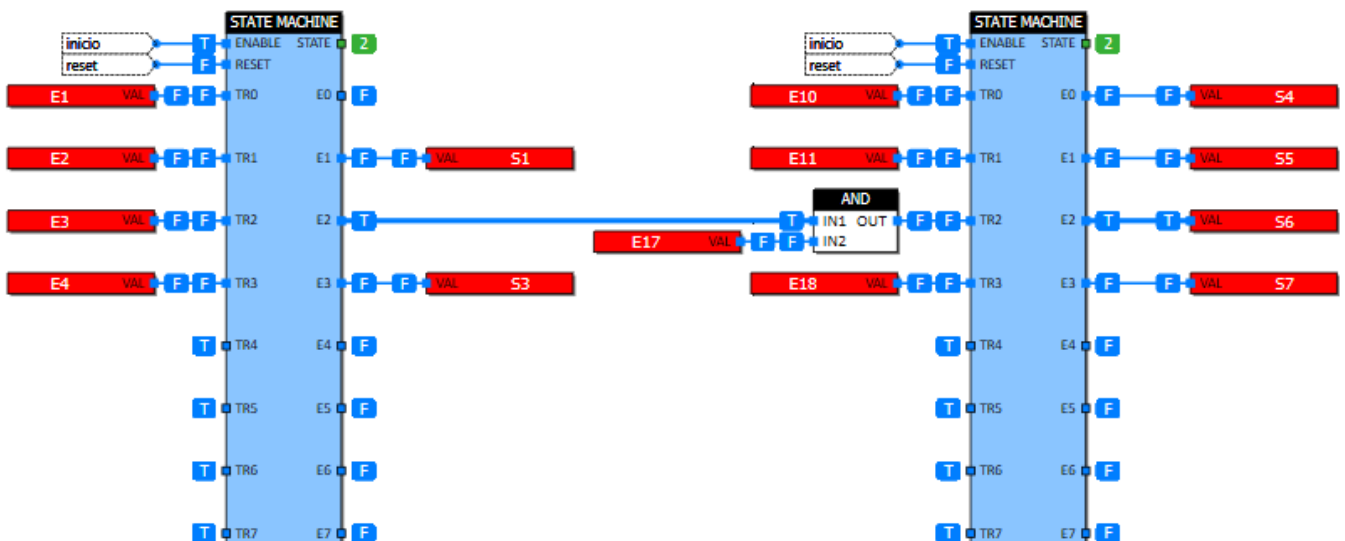
O último estado do último bloco precisa resetar todos os blocos, desabilitando todas as saídas, colocando a máquina inicial no estado 0 e permitindo condições para o início de um novo ciclo.

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.



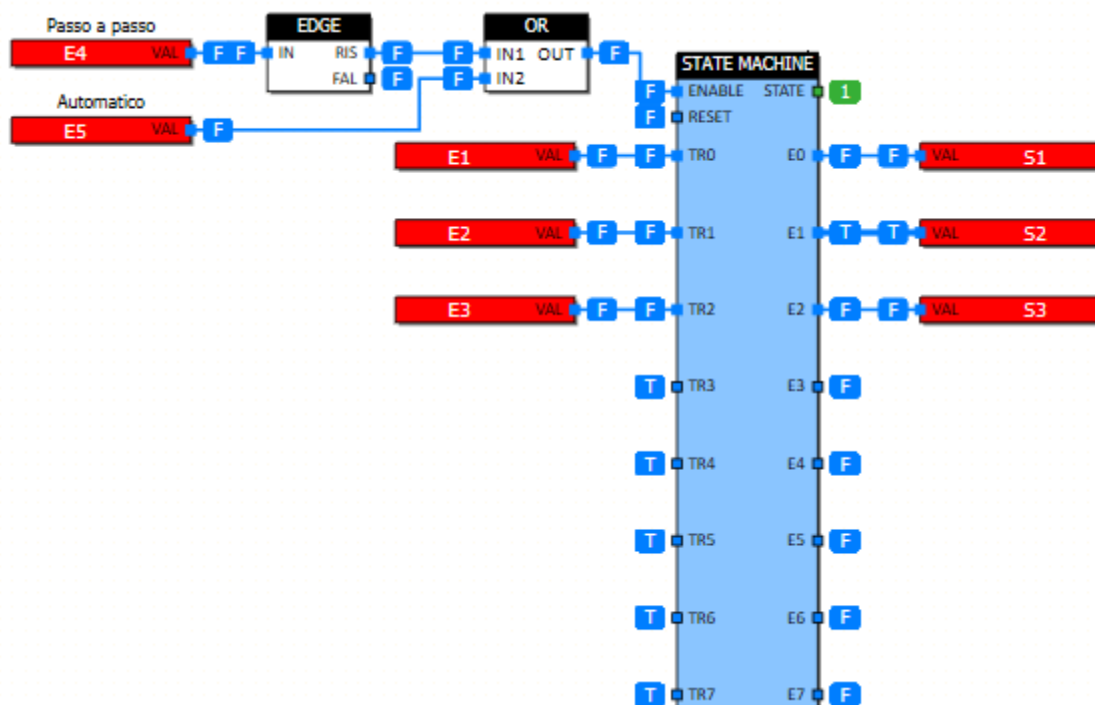
Máquina de estados em modo passo a passo

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.



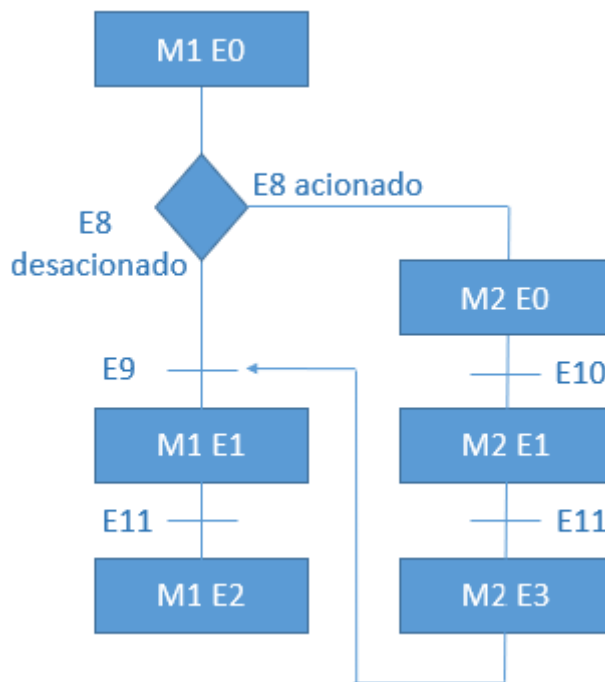
Função sub-rotina

Essa configuração permite pausar a sequência principal, executar uma rotina auxiliar e retornar a sequência principal (após a conclusão da sub-rotina) até finalizá-la.

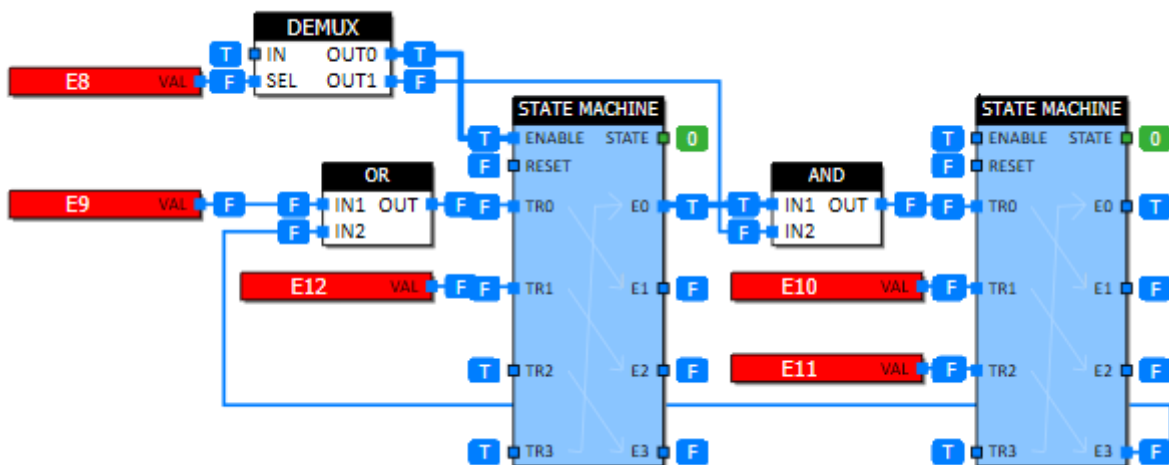
Exemplo:

A M1 (principal) e M2 (auxiliar) iniciam no estado 0 e E8 controla a ativação da sub-rotina. Se está em M1 E0 e E8 está acionada, a M2 (sub-rotina) é executada e no último estado (E3) transiciona Tr1 da M1 para seguir a rotina principal, e se E8 não é acionada, o programa executa apenas a rotina principal. Abaixo estão demonstrados o fluxograma e a lógica de programação no TProg desse exemplo.

Fluxograma:



Lógica no TProg:




Elaborado por Eng. Kelvin Soares (suporte@tecnolog.ind.br) e Eng. Cláudio Zardo.

Última edição: 23/06/2022

TECNOLOG

www.tecnolog.com.br

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

 Telefone: (51) 3076.7800

 E-mail: vendas@tecnolog.ind.br