

AUTOMAÇÃO INDUSTRIAL

Controladores Lógicos Programáveis - CLP

Prof. Paulo Leite
2018

1. Critério de aprovação

$$NF = ((10 - (F * 0,7) + 2 * (T1 * PO1) / 10 + 2 * (T2 * PO2) / 10) / 5$$

F = FALTAS T = TRABALHO PO = PROVA ORAL SOBRE TRABALHO

2. Referencias bibliográficas:

- Automação Industrial
 - Francesco Prudente, LTC
- Controladores Logicos Programaveis
 - Claiton M. Franchi – Valter L. A. Camargo, Erica

Estes livros foram utilizados na confecção desta apostila.

3. Histórico

Desde o início do século passado surgiu a necessidade da utilização de dispositivos para dar partida, proteger componentes e vigiar dia e noite as condições de segurança nos processos de eletrificação. Inicialmente estes dispositivos eram implementados com contadores, disjuntores, relés de proteção, chaves manuais etc.

Os sistemas desenvolvidos com relés eram grandes, complexos, de difícil manutenção e impossíveis de serem alterados.

Os CLP's surgiram na década de 60 substituindo os painéis com as seguintes vantagens:

- Possibilidade de reprogramação,
- Hardware padrão,
- Facilidade de manutenção,
- Possibilidade de ligação em rede.

O advento do microprocessador permitiu uma diminuição nos custos e tamanho dos CLP's. Atualmente existe uma forte tendência na utilização de pequenos CLP's controlando processos locais e comunicando-se com outros controladores e com sistemas supervisórios via redes industriais.

4. Definição de CLP

O CLP é um equipamento composto de componentes eletrônicos e memória programável ou não-programável que contem dados e programas com a finalidade de ler e executar instruções, interagindo com um sistema que deve ser controlado por dispositivos de input e output do tipo digital ou analógico (norma IEC 61131-1).

5. Utilização dos CLPs na automação industrial

Toda planta industrial necessita de algum tipo de controlador para garantir uma operação segura e economicamente viável. Embora existam tamanhos e complexidades diferentes, todos os sistemas de controle podem ser divididos em três partes com funções bem definidas: os **transdutores/sensores, os controladores e os atuadores**.

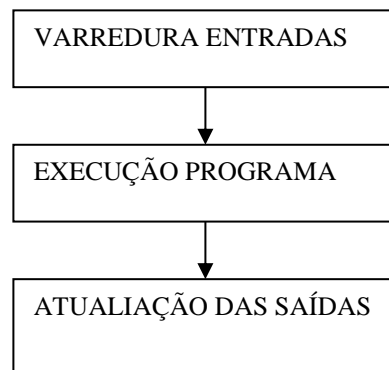
- Sensores/transdutores: transdutor é um dispositivo que converte uma condição física em um sinal elétrico para ser utilizado pelo CLP. Chaves de fim de curso e sensores de temperatura são exemplos de transdutores.
- Atuadores: Sua função é converter o sinal elétrico oriundo do CLP em uma condição física, normalmente ligando ou desligando algum elemento.
- Controladores: De acordo com os estados das suas entradas, o controlador utiliza um programa de controle para calcular os estados das suas saídas.

6. Conceitos básicos

“Programa aplicativo” é o programa que controla o funcionamento do CLP. Avalia as condições de entrada, os estados internos e aciona as saídas.

Funcionamento:

- Lê ciclicamente as entradas -> memória imagem
- Mem. Imagem das entradas+estados internos-> memória imagem saídas
- Memória imagem saídas-> saídas



7. Vantagens na utilização de CLP

- **Flexibilidade:** É possível alterar o funcionamento de uma máquina controlada por um PLC apenas alterando o software. No passado era necessário alterar o circuito elétrico.
- **Facilidade de manutenção:** É possível incluir rotinas de diagnóstico nos PLCs atuais, simplificando em muito a manutenção das máquinas.
- **Baixo custo:** CLPs simples custam poucas centenas de dólares.
- **Acompanhamento da operação:** Os CLPs podem ser ligados a computadores, via rede industrial, de forma a permitir que o operador acompanhe todas as variáveis do processo. Para isso é muito comum o uso de softwares supervisórios.

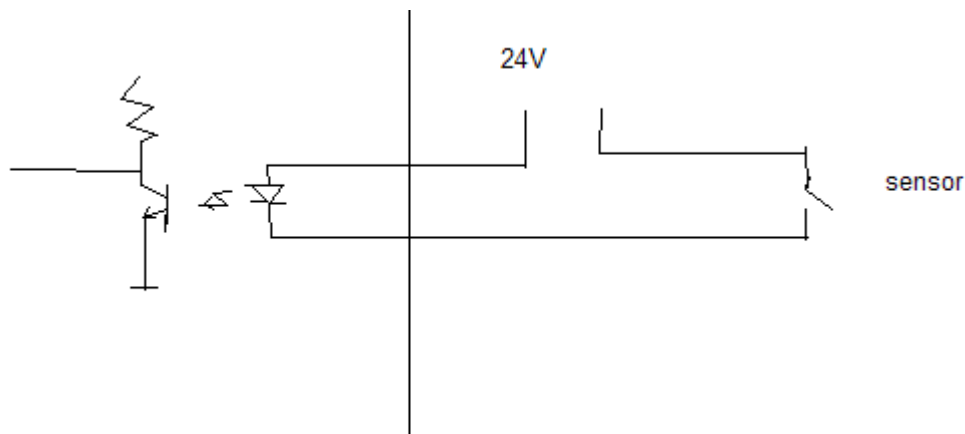
- **Facilidade de programação:** Fazer um software para PLC é muito mais fácil e rápido do que fazer um projeto elétrico com as mesmas funções.

8. Entradas e saídas

8.1. Entradas

Podem ser digitais 0 – 24V (0 ou 1)
ou analógicas 0 – 10V, 4 – 20mA (0-255 ou 0-4095 ...).

Exemplos de entradas digitais: sensores de fim de curso, botoeiras, contatos de reles etc. As entradas digitais são normalmente isoladas através de isoladores ópticos conforme circuito abaixo:



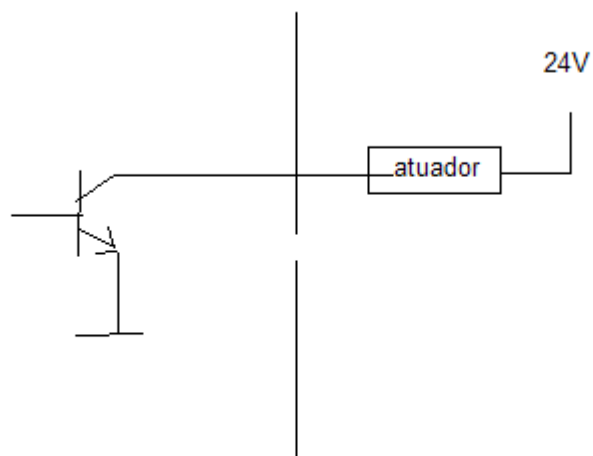
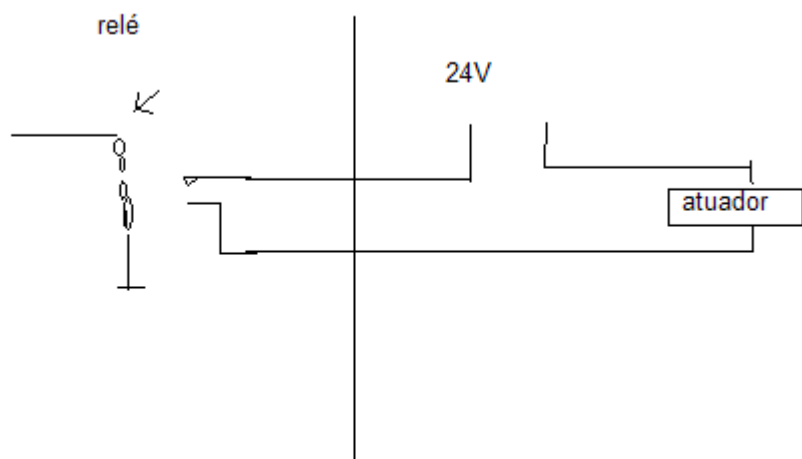
Exemplos de entradas analógicas: temperatura através de termopares ou PT100, posição através de sensores resistivos etc.

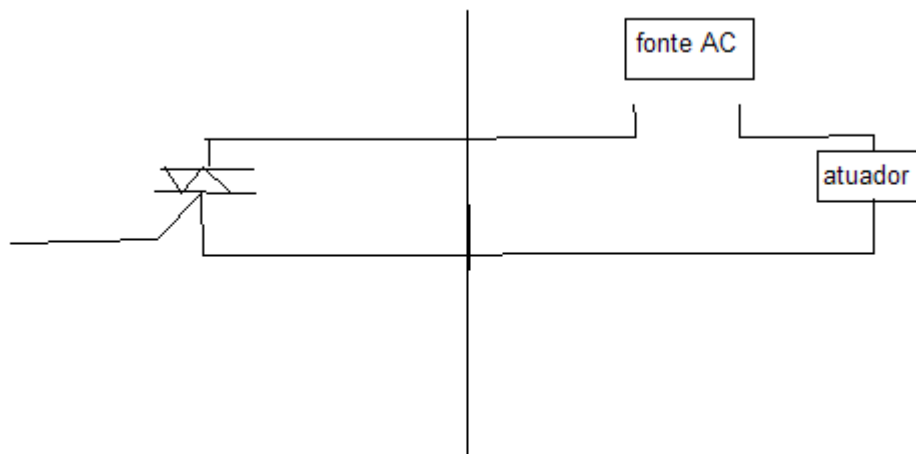
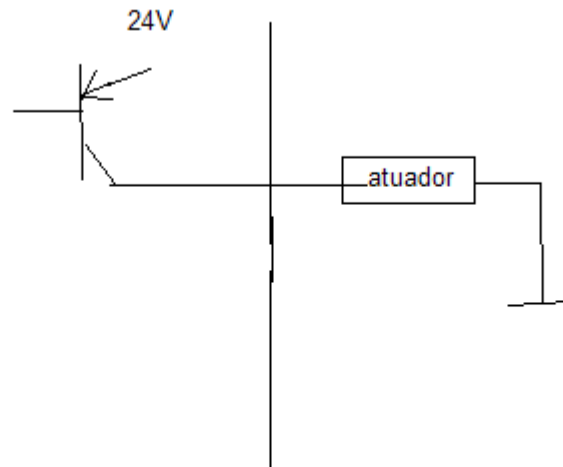
8.2. Saídas

Podem ser digitais 0 – 24V (0 ou 1 lógico)
ou analógicas 0 – 10V, 4 – 20mA (0-255 ou 0-4095 ...)

Saídas digitais são implementadas por relés, transistores ou SCR/TRIACS. São utilizadas para acionar lâmpadas, motores, válvulas etc.

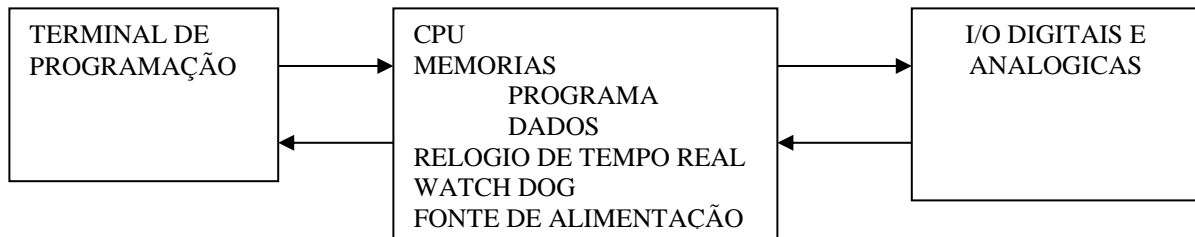
Exemplo de saídas digitais:





Saídas analógicas fornecem normalmente 0 – 10V ou 4 – 20mA e são utilizadas para acionar válvulas proporcionais, controlar velocidade de motores etc.

9. Componentes de um CLP



Terminal de programação: PC, laptop ou terminal portátil. Permite:

- Gravação ou apagamento memória
- Auto-diagnóstico
- Alterações on-line
- Programação de instruções
- Monitoração

CPU: responsável pela execução do programa do usuário, atualização da memória de dados e memória imagem das entradas e saídas. Roda em back ground um programa monitor, elaborado pelo fabricante, que faz start-up do controlador e gerencia a sequência de operações.

Memórias:

- EPROM – S.O. / sequência operações/ drives
- EEPROM ou Flash – programa do usuário
- RAM – imagem de dados E/S / dados do usuário

I/O: É por onde existe a interligação do PLC com o processo, entradas e saídas digitais e analógicas.

10. Linguagens de programação

O IEC padroniza as linguagens através da norma IEC1131-3. Principais linguagens:

Tabulares	Tabela de decisão	Obsoleta
Textuais	Instruction List – IL Structured Text – ST	
Gráficas	Diagrama de Relés – LD Function Block Diagram – FBD Sequential Flow Chart – SFC	

IL – assemelha-se a linguagem assembler. É de difícil aprendizado para engenheiros de manutenção.

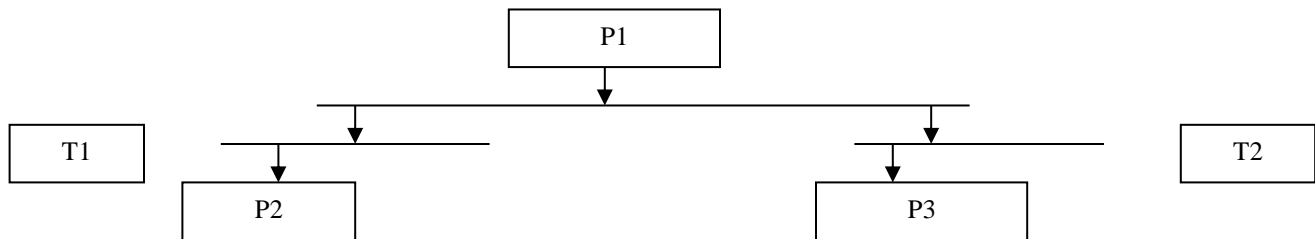
Ex: $(O5) = (I1) \cdot (I2 \text{ NEGADO}) \cdot (I3) + (I4)$

LD	I1
ANDN	I2
AND	I3
OR	I4
ST	O5

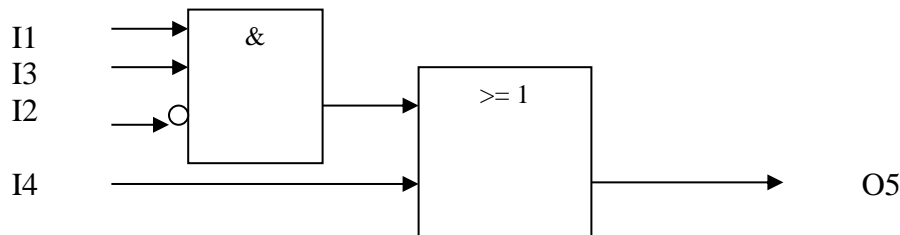
ST – linguagem de alto nível

Ex: O5 = I1 AND NOT I2 AND I3 OR I4

SFC – representa-se em sequência, graficamente, as etapas do programa. É uma linguagem gráfica que se originou das redes de petri.



Function Block Diagram - FBD – linguagem de programação gráfica utilizando blocos da lógica Booleana como comandos padronizados.



11. Linguagem LADDER

LD – Diagrama de contatos ou Ladder Diagram permite programar desde funções binárias até funções matemáticas complexas. A linguagem Ladder permite transladar com relativa facilidade os diagramas elétricos baseados em relés para o CLP.

1	4	7	10	13
2	5	8	11	14
3	6	9	12	15

Barra esquerda

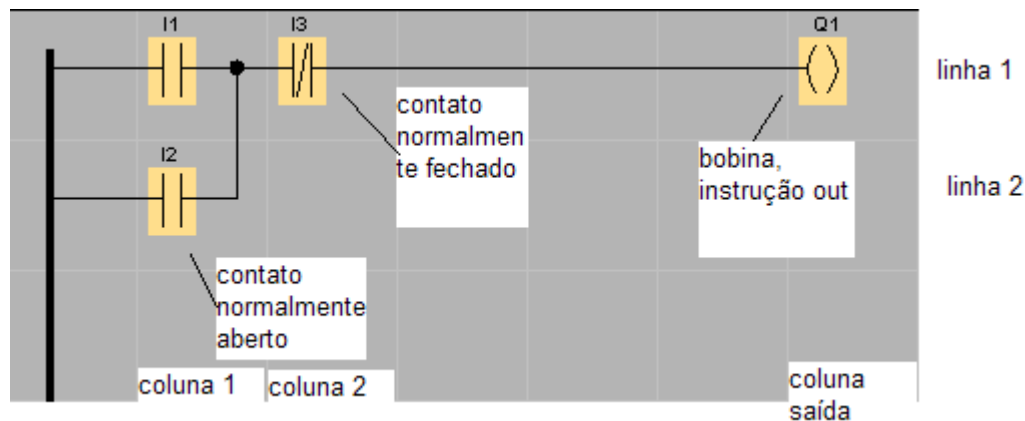
barra direita

Entre as duas barras são colocadas as instruções a serem executadas. As instruções podem ser contatos, bobinas, temporizadores, contadores etc.

Cada célula pode ser ocupada por uma conexão (fio), por um bloco, ou por um contato. Os números representam a ordem de execução das células.

Mesmo tendo sido a primeira linguagem destinada especificamente à programação de CLP's, a linguagem ladder continua sendo a mais utilizada.

Exemplo de programa:



Mesmo programa em lista de instruções:

```
LD      I1
OR      I2
ANDN    I3
ST      Q1      ( ou OUT Q1)
```

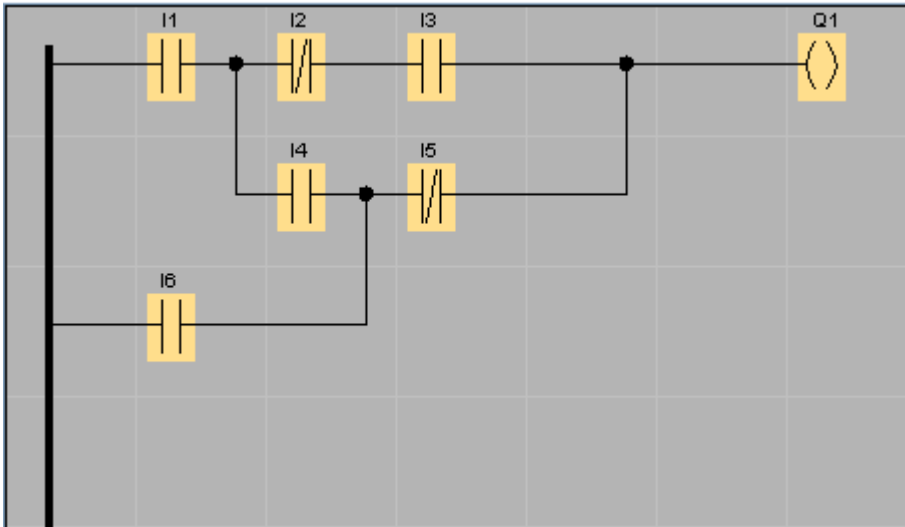
Ex: Qual a lógica de contatos em paralelo? E em série?

Corrente fictícia:

No exemplo anterior, quando o contato I1 ou I2 for acionado, a corrente fluirá passando por I3 (caso desenergizado, pois é normalmente fechado) até energizar a bobina Q1.

O sentido da corrente fictícia é sempre da esquerda para a direita.

No exemplo abaixo, mesmo que acionados I6, I4, I2 e I3, se I1 ou I5 não estiverem acionados a saída não será acionada.



Ex: escreva a expressão lógica do diagrama Ladder acima.

$$I1 \cdot ((I2 \cdot I3) + (I4 \cdot I5)) + I6 \cdot I5$$

12. – Primeiro contato com o programa COMFORT V5.0

Os capítulos seguintes têm por objetivo aprofundar o conhecimento do aluno na linguagem Ladder através de uma série de exercícios utilizando o CLP LOGO da Siemens.

12.1. Passo a passo para 1º programa

- Carregar: Iniciar, siemens LOGO soft, LOGO! Software confort V5.0
- Abrir novo projeto: Clicar na página branca (primeiro ícone à esq.) e preencher propriedades
- Passar para linguagem LADDER: Ficheiro, converter para LADDER
- Ajuda: Em “ajuda, conteúdo” existem vários itens explicativos incluindo um tutorial bastante útil.

Os itens abaixo apresentam os passos para a criação de um projeto, desenvolvimento de um programa e gravação no PLC.

- Clicar em contato de trabalho, entradas digitais (primeiro ícone à esq.)
- Clicar na parte superior esq. Do área de trabalho – usar I1,
- Clicar novamente abaixo do anterior – usar I4
- Clicar novamente ao lado de I4 – usar I5

- Clicar agora em bobina de relé – saídas digitais
- Clicar ao lado de I1 – usar Q1
- Clicar ao lado de I5 – usar Q4 (clicar duas vezes botão direito para entrar em propriedades)
- Clicar na ferramenta de ligar (logo abaixo da tesoura no lado esq) e ligar saída de I1 em Q1, I4 em I5 em Q4.
- Salvar – clicar em ficheiro, guardar como, criar subdiretorio e salvar como exemplo 1
- Transferir para PLC – clicar em extras, transferir, PC->Logo!
- Testar funcionamento do PLC:
- Acionando I1 -> Q1 on, acionando I4 e I5 -> Q4 on.

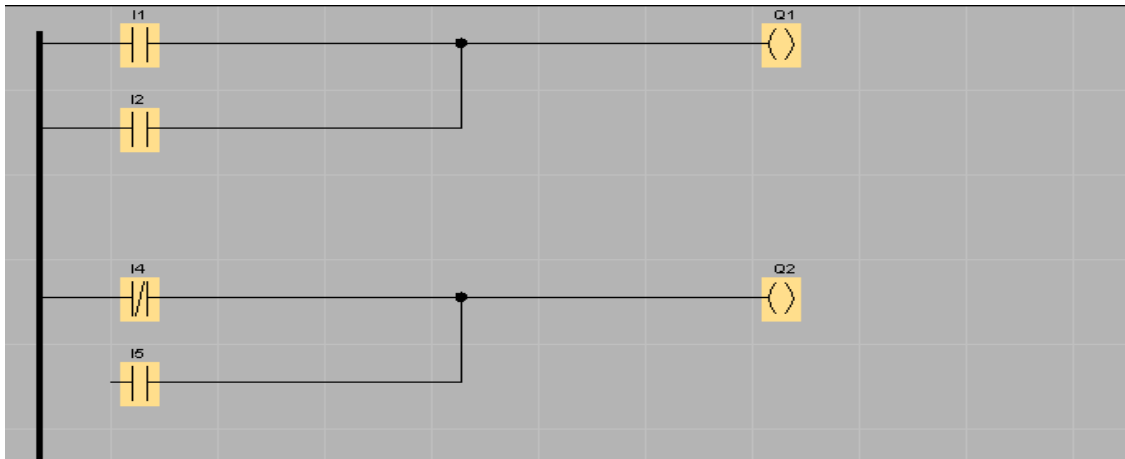
O seu programa deverá ficar com as ligações iguais as da figura abaixo.
 Salve seu programa como **exemplo1**



13. OUTROS EXEMPLOS

Nos próximos exemplos utilizaremos diversos blocos disponíveis no LOGO e na maioria dos CLPs.

13.1. Exemplo 2 – OR e contato normalmente fechado.



Ex: Em que condições Q1 é acionado? E Q2?

Escreva a função lógica deste programa e teste seu funcionamento no PLC.

$Q1 = I1 + I2$ $Q2 = I4 + I5$

13.2. Usando o simulador

Após a escrever um programa é possível testar a sua lógica através do simulador. Para tanto, basta pressionar a tecla F3 ou clicar no icone mais abaixo da barra de ferramentas à esquerda.

Teste o simulador com os programas que você já fez.

13.3. Exemplo 3 – timer, retardamento de ligação



Funcionamento: Ao acionar I1, a saída do timer só vai a 1 após o tempo programado (10s). Desligando I1 o timer vai a zero instantaneamente.

Conexão	Descrição
Entrada Trg	Por meio da entrada Trg (Trg indica trigger) o Sr. dará a partida para o tempo de retardamento de ligação
Parâmetros	T é o tempo de retardamento segundo o qual a saída será ligada (sinal de saída comuta de 0 para 1). Remanência seleccionada (on) = o estado é guardado de forma remanente.
Saída Q	Q liga após o decurso do tempo parametrizado T, se o Trg ainda estiver regulado na posição.

Parâmetro T

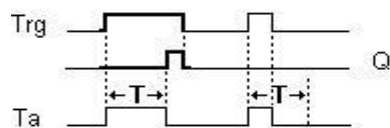
O tempo especificado para o parâmetro T também pode ser um valor actual de uma outra função já programar os valores actuais das seguintes funções:

- Comparador analógico
- Interruptor de valor limiar analógico
- Amplificador analógico e
- Contador crescente e decrescente.

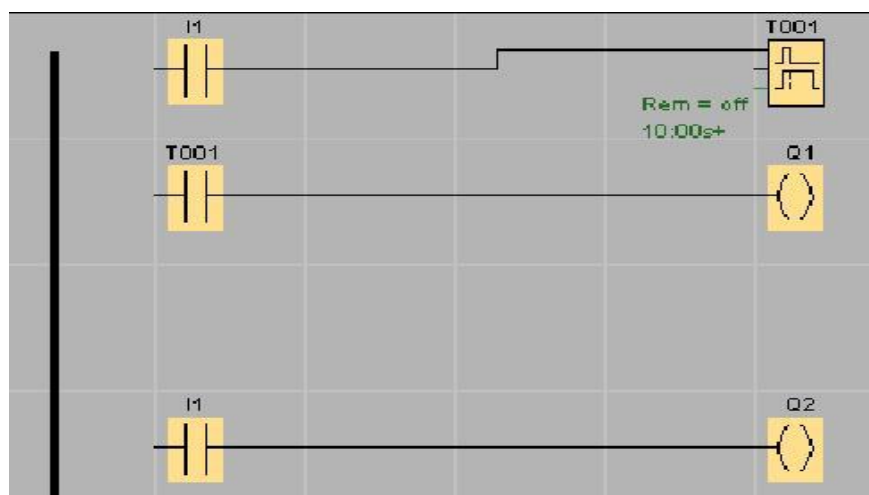
Selecione a função desejada através do número de bloco.

Para garantir a validade e a precisão da base do tempo, consulte o manual do LOGO!

Diagrama do timing



13.4. Exemplo 4 – timer, retardamento de desligamento

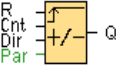


Funcionamento: Ao acionar I1, a saída do timer vai a 1 instantaneamente. Desligando I1 o timer vai continuar em 1 pelo tempo programado (10s).

14. MAIS EXEMPLOS

14.1. Exemplo 5 – contador crescente e decrescente.
Use o simulador para testar: I1 -> reset I2-> contagem I3 -> direção

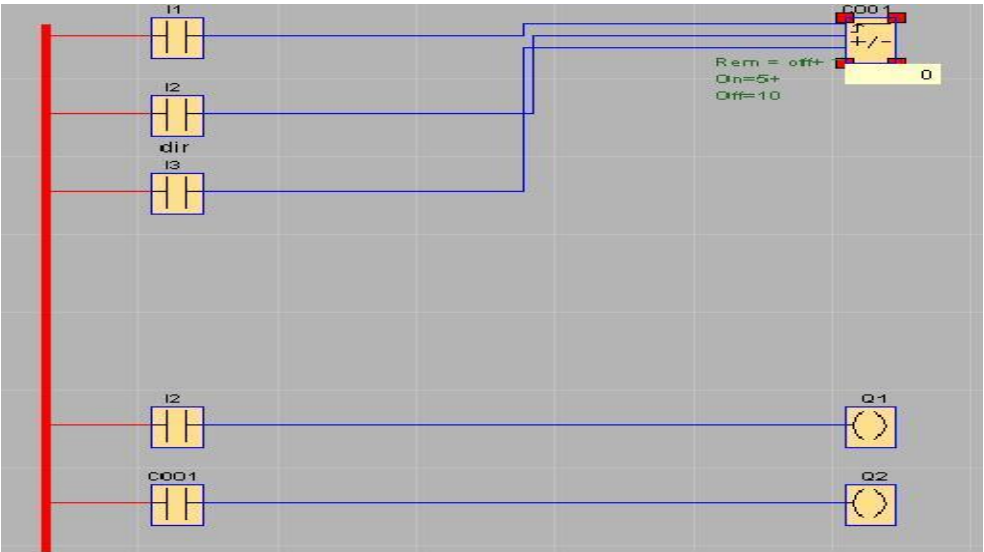
Up/Down counter



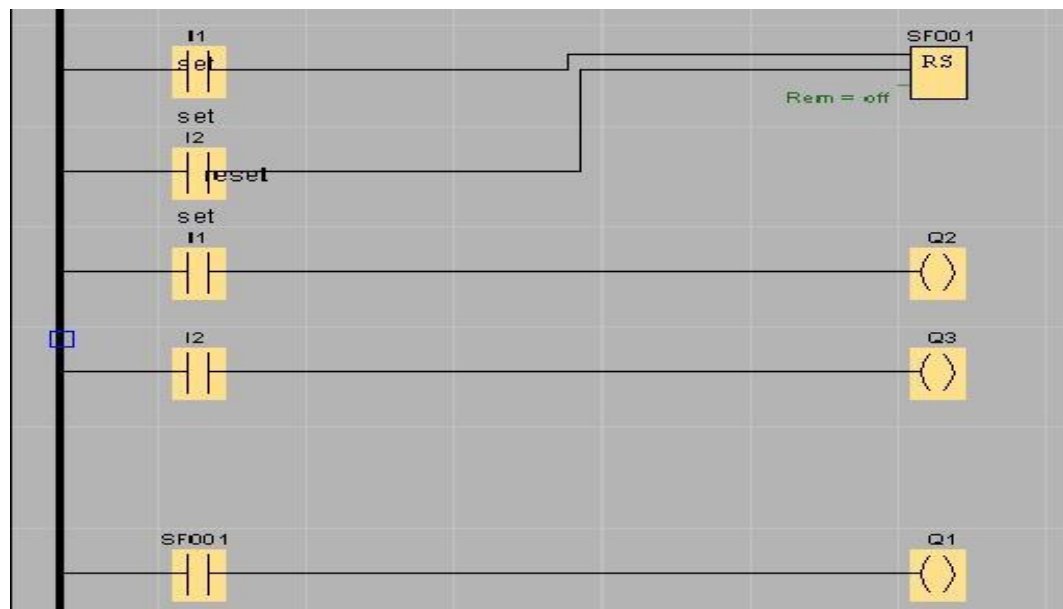
Short description

An input pulse increments or decrements an internal value, depending on the parameter setting. The output is set or reset when a configured threshold is reached. The direction of count can be changed with a signal at input Dir.

Connection	Description
Input R	You reset the output and the internal counter value to the start value (StartVal) with a signal at input R (Reset)
Input Cnt	This function counts the 0 to 1 transitions at input Cnt . It does not count 1 to 0 transitions. <ul style="list-style-type: none">● Use the inputs I3, I4, I5, and I6 for high-frequency counts (LOGO! 12/24RCE/RCEo, LOGO! 24CE/24CEo): max. 5 kHz, if the fast input is directly connected to the Up/Down counter function block● Use any other input or circuit element for low-frequency counts (typically 4 Hz).
Input Dir	Input Dir (Direction) determines the direction of count: Dir = 0: Up Dir = 1: Down
Parameter	On: On threshold / Value range: 0 to 999999 Off: Off threshold / Value range 0 to 999999 Start Value: Initial value from which to begin counting either down or up. Retentivity on = the status is retentive in memory.
Output Q	Q is set and reset according to the actual value at Cnt and the set thresholds.

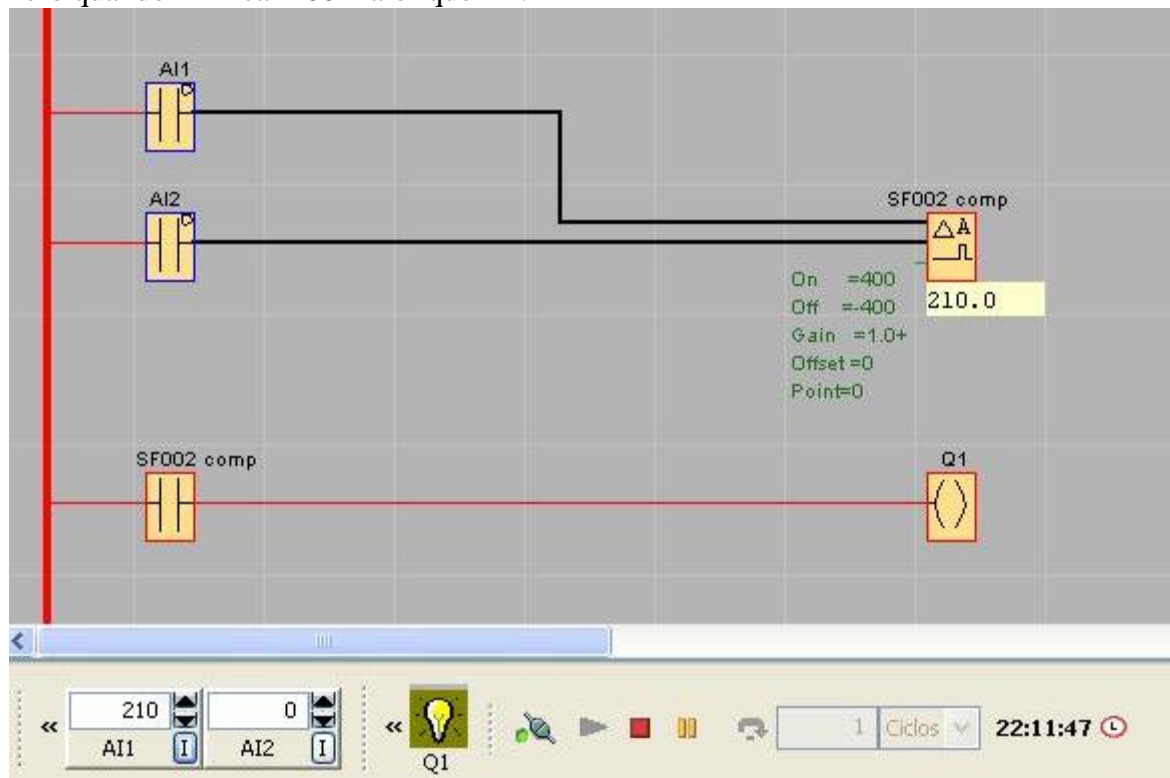


14.2. Exemplo 6 – FilpFlop S/R



14.3. Exemplo 7 – comparador analógico

Quando a entrada A1 ficar 400 (0 – 1000) maior que A2, a saída vai a 1 e só volta a zero quando A2 ficar 400 maior que A1.



15. PROJETOS

15.1. Principais fases de um projeto de automação

- Detalhamento, junto com o cliente, das características técnicas do produto. Gerar o documento Especificação Técnica,
- Definir todas as entradas e saídas do CLP,
- Fazer o projeto elétrico, incluindo sensores, chaves, atuadores,
- Elaboração do software, incluindo testes por simulação,
- Montagem do painel,
- Comissionamento incluindo:
 - Testes de continuidade e isolamento em todas as ligações,
 - Verificação de que todos os componentes são os que foram especificados no projeto,
 - Teste da fonte de alimentação,
 - Testes funcionais,

15.2. Projeto Portão automático 1

Implementar um controlador para um portão automático de garagem.

Entradas: I1 – Botoeira abrir
I2 – Botoeira fechar
I3 – chave de fim de curso de porta fechada
I4 – chave de fim de curso de porta aberta

Saídas: Q1 – motor para abrir portão
Q2 – motor para fechar portão

Funcionamento:

- Se botoeira abrir for acionada, o portão deve abrir até a chave de fim de curso de porta aberta seja acionada.
- Se botoeira fechar for acionada, o portão deve fechar até a chave de fim de curso de porta fechada seja acionada.
- O controlador deverá ter um intertravamento que impeça a ligação simultânea do motor de abrir e de fechar.
- O controlador deverá ter um intertravamento para não ligar o motor de abrir se o portão já estiver aberto, e não ligar o motor de fechar se o portão já estiver fechado.

Salve o projeto e teste no simulador.

15.3. Portão automático 2

Altere o programa acima de forma que ele tenha um sensor tipo barreira ótica que pare o motor caso haja algo na frente do portão. O portão tem que continuar fechando quando o obstáculo sair da frente do portão. A barreira ótica deverá ser conectada na entrada I5

15.4. Portão automático 3

Altere o programa acima de forma que:

- Se o portão estiver fechando e for acionada a chave de abrir, o motor de fechar é desligado e o motor de abrir é ligado,
- Se o portão estiver abrindo e for acionada a chave de fechar, o motor de abrir é desligado e o motor de fechar é ligado.

15.5. Portão automático 4

Implementar um controlador para um portão automático de garagem.

Entradas: I1 – sensor de carro fora da garagem
I2 – sensor de carro dentro da garagem
I3 – chave de fim de curso de porta fechada
I4 – chave de fim de curso de porta aberta
Saídas: Q1 – motor para abrir portão
Q2 – motor para fechar portão

Funcionamento:

- Ao chegar um carro próximo ao portão, acionando um dos sensores, o portão abre até a chave de fim de curso de porta aberta ser acionada.
- Após 10Seg que o portão chegar ao sensor de portão aberto, o motor de descida é acionado até que o sensor de fim de curso de porta fechada seja acionado.
- O controlador deverá ter um intertravamento que impeça a ligação simultânea do motor de subir e de descer.
- O controlador deverá ter um intertravamento para não ligar o motor de abrir se o portão já estiver aberto, e não ligar o motor de fechar se o portão já estiver fechado.

15.6. Portão automático 5

Altere o programa acima de forma que ele não use mais o temporizador e só feche a porta após o carro acionar o sensor do outro lado da porta.

16. O CLP LOGO 12/24 RC

Os itens abaixo apresentam partes do manual da LOGO, com as principais informações de entradas, saídas e memórias.

16.1. Entradas

Digital inputs	
Number	8
Electrical isolation	No
Number of high speed inputs	4 (I3, I4, I5, I6)
Input frequency <ul style="list-style-type: none"> Normal input High speed input 	<ul style="list-style-type: none"> max. 4 Hz max. 5 kHz
Max. continuous permissible voltage	28.8 V DC
Input voltage Signal 0 Signal 1	L+ < 5 V DC > 12 V DC
Input current at Signal 0 Signal 1	< 0.85 mA (I3...I6) < 0.05 mA (I1, I2, I7, I8) > 2 mA (I3... I6) > 0.15 mA (I1, I2, I7, I8)
Delay time at 0 to 1 1 to 0	typ. 1.5 ms <1.0 ms (I3 ... I6) typ. 1.5 ms <1.0 ms (I3 ... I6)
Line length (unshielded)	max. 100 m
Analog inputs	
Number	4 (I1=AI3, I2=AI4, I7=AI1, I8=AI2)
Range	0 ... 10 V DC input impedance 72 k Ω

16.2. Saídas

Digital outputs	
Number	4
Output type	Transistor, current-sourcing ¹⁾
Electrical isolation	No
In groups of	- -
Control of a digital input	Yes
Output voltage	\leq Supply voltage
Output current	max. 0.3 A
Short circuit-proof and overload-proof	Yes
Short circuit current limitation	Approx. 1 A
Derating	none; across the entire temperature range

16.3. Memória

Bytes	Blocos	REM
2000	130	60

Existem as memórias disponíveis ao programador chamadas de FLAG, ou seja, armazenam apenas um bit. São 27 memórias sendo que a M8 é acionada apenas no primeiro ciclo e M25,26,27 são reservadas para controle do display e de mensagens. Existem também as memórias analógicas AM1 a AM6, todas capaz de armazenar números de 0 a 1000.

17. Projeto de uma plataforma de elevação

Fazer um programa para controle de uma plataforma de elevação com as seguintes características:

17.1. Funcionamento:

- Saída do PLC aciona um inversor de frequência com rampa de subida programável,
- Com o motor parado o freio deve ser acionado,
- Botão subir aciona motor que só para no FC superior ou botão de emergência,
- Botão descer aciona motor que só para no FC inferior ou botão de emergência,
- Se botão de emergência for acionado, motor para, freio é ligado, e saída de alarme é ativada,
- Botão reset desliga alarme,
- Sensor de velocidade gera pulso proporcional a velocidade. Caso os pulsos aconteçam em tempo inferior a 1 segundo, alarme de velocidade. Motor para, freio é ligado e saída de alarme é ativada com frequência de 1Hz. Botão reset desliga alarme de velocidade,
- Ao chegar ao sensor FC superior ou inferior, motor para, o freio é ligado e alarme é ligado por 2 segundos,
- Com o motor de subir ligado, a botão de descer fica inoperante. O mesmo deve ocorrer com o motor de descer ligado.

17.2. Entradas:

I1	Botão subir - BS
I2	Botão descer - BD
I3	Botão emergência - BE
I4	Reset
I5	Sensor de velocidade - SV
I6	FC sup.
I7	FC inf.

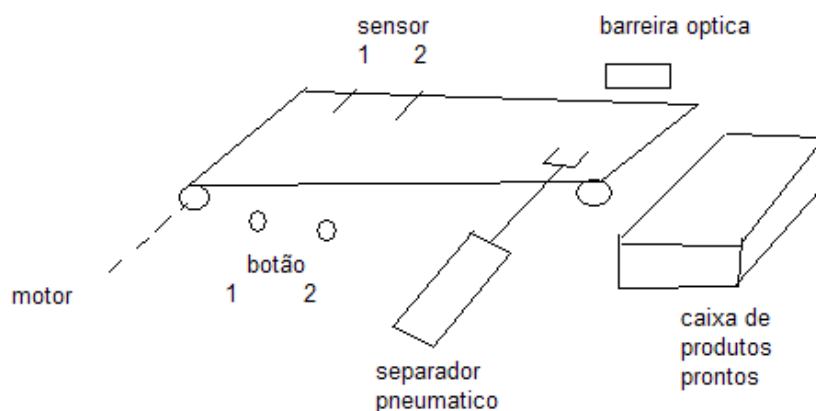
17.3. Saídas:

Q1	Motor subir - MS
Q2	Motor descer - MD
Q3	Freio motor - FM
Q4	Alarme - AL

18. Projeto de uma esteira de empacotamento.

Este capítulo consiste em um projeto orientado e completo de uma esteira de empacotamento. As principais funções existentes em CLPs serão praticadas a partir dos diversos exercícios que compõe o projeto.

18.1. O desenho abaixo apresenta a esteira:



Entradas e saídas do CLP:

I1	S1	SENSOR 1
I2	BT1	BOTOEIRA 1
I3	BO	BARREIRA OPTICA
I4	S2	SENSOR 2
I5	BT2	BOTOEIRA 2
I6	TC	TROCA DE CAIXA OK
I7	ON	BOTOEIRA ON
I8	OFF	BOTOIERA OFF
Q1	CM	CONTROLE DO MOTOR
Q2	SP	SEPARADOR PNEUM.
Q3	CC	LUZ CAIXACHEIA
Q4	ON	LUZ ON (SISTEMA LIGADO)

18.2. Funcionamento:

Inicialmente os operadores devem pressionar o botão on para ligar o sistema.

Em operação normal, a esteira fica parada esperando que os operadores coloquem uma peça pronta na posição 1 ou 2. O operador deve então pressionar a botoeira correspondente que fará com que o motor ligue.

Quando a peça passar pela barreira óptica e cair na caixa, o motor deve ser desligado liberando para que o operador coloque outra peça.

A cada 10 peças, o acionador pneumático deve ser acionado por 4 segundos para separar uma peça para testes de CQ. Neste caso, o motor deve ser desligado quando a peça entra na barreira óptica, iniciando o acionamento do SP.

A cada 5 peças o sinalizador CC caixa cheia deve acender para o operador trocar a caixa cheia por uma vazia. Ao final da troca o operador deve pressionar a botoeira TC para liberar o transporte de uma nova peça..

18.3. Diagrama temporal



18.4. Exercício 1 – Controle do estado ON e OFF

Fazer um programa que, ao acionar momentaneamente a entrada *on*, (I7) a variável operação vai a 1 (acendendo a luz *on* = Q4) só voltando a zero quando a chave *off* (I8) for pressionada momentaneamente.

18.5. Exercício 2 – controle do motor

Implementar um código para controle do motor da seguinte forma:

Ligar o motor se o sistema estiver “on”, existir uma peça na posição 1 e a botoeira 1 for apertada ou o sistema estiver “on”, existir uma peça na posição 2 e a botoeira 2 for apertada.

Se existirem duas peças, uma em cada posição, o motor não deverá ligar.

Desligar o motor quando ele terminar de passar pela barreira óptica.

18.6. Exercício 3 – controle da lâmpada operação

Alterar o código de acendimento da lâmpada on de forma que ela fique piscando em uma frequência de 4 Hz sempre que uma peça estiver sendo transportada.

18.7. Exercício 4 – controle da falha

Criar subrotina alarme que monitora o tempo para a peça chegar na barreira óptica. Se o tempo for maior que 10seg, o motor deve ser desligado, e a lâmpada caixa cheia deverá começar a piscar com uma frequência de 4Hz e o sistema deverá sair de on (apagando a lâmpada on). O sistema deverá ser levado para off para que retorne ao funcionamento normal. A chave on limpa a condição de alarme.

18.8. Exercício 5 – controle da produção

O numero de peças produzidas por cada operador e total deverá ser monitorado e guardado na memória de programa.

A cada 10 peças, o acionador pneumático deve ser acionado por 4 segundos para separar uma peça para testes de CQ.

A cada 5 peças o sinalizador CC caixa cheia deve acender para o operador trocar a caixa e pressionar a botoeira TC.

19. Variáveis analógicas

O LOGO é um PLC para pequenas aplicações, porem possui entradas e disponibiliza uma série de funções para variáveis analógicas. Seguem algumas funções analógicas disponíveis no LOGO:

- Analog contact: por onde o programa recebe o valor digitalizado de uma entrada analógica. Sempre de 0 – 1000 para entradas de 0 – 10Volts.
- Analog amplifier: multiplica a entrada por A e soma por B.
- Analog comparator: A saída é setada ou resetada se Ax for maior ou menor que Ay.
- PI controler: É um controlador com as ações proporcional e integral.
- PWM: É um modulador de largura de pulso que utiliza a entrada analógica para definir o tamanho do pulso ON.

20. PROJETOS COM VARIÁVEIS ANALÓGICAS

20.1. Controlador ON-OFF

Implementar um código em Ladder para fazer o controle automático de temperatura de um processo de aquecimento com as seguintes partes:

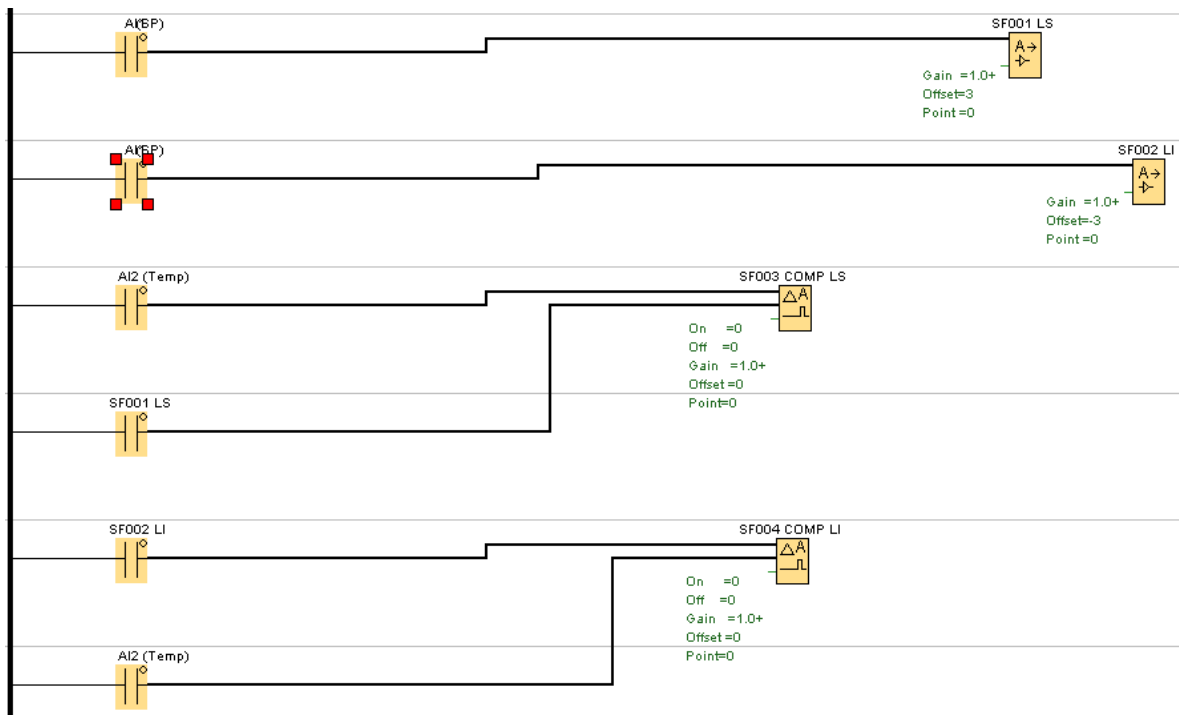
- Sensor de temperatura: LM35, produz 10mV por grau centigrado. Ex: 25°C -> 250mV ou 0,25V. Ou seja, cada step vale 1 grau.
- EFC, resistência de 30W controlada por relé de estado solido,
- Ventoinha para auxiliar na refrigeração.

Entradas e saídas do PLC:

- AI1 – I7 : Set Point vindo de potenciômetro,
- AI2 – I8 – Sensor de temperatura,
- Q1 – saída ON-OFF para acionar o relé,

Funcionamento:

- Se a temperatura for maior que o limite superior LS, desliga resistência,
- Se a temperatura for menor que o limite superior LI, liga resistência,





Fazer as seguintes alterações no programa acima:

- Alterar SP de 0 a 1.000 para 0 a 100graus,
- Incluir entrada I1 – para Ligar/desligar a saída digital,
- Utilizar I2 – para aumentar em 10 graus o Set Point,
- Q2 – saída caso a temperatura esteja 20% maior que o Set Point.

20.2. Acionamento com PWM

PWM é uma técnica que permite fornecer um valor médio linear a partir de um elemento de controle do tipo on-off. O período, e como consequência a frequência, de operação de um PWM é sempre o mesmo. O controle linear é realizado pelo tempo em que a saída fica em 1 e em 0.

Ex: PWM com período de 10 segundos e potência de 20%. A saída fica 2 seg em 1 e 8 seg em zero.

O LOGO tem um bloco PWM

20.3. Controlador PI