3 - Introdução ao estudo dos PLC's

3.1 - Contactor ou relé

Antes de entrar no estudo dos PLC's ("*Programmable Logic Controllers*", ou "*controladores lógicos programáveis*") é de todo o interesse abordar, de forma muito resumida, a lógica de relé.

Durante anos, esta tecnologia foi a única forma de controlar muitos processos industriais. O seu princípio de funcionamento tem por base a utilização de um dispositivo electromecânico chamado relé.

Este componente é actuado, ou accionado, fechando o circuito que contém a bobina. A este circuito dá-se a designação de circuito de comando, operado tradicionalmente a tensão reduzida.

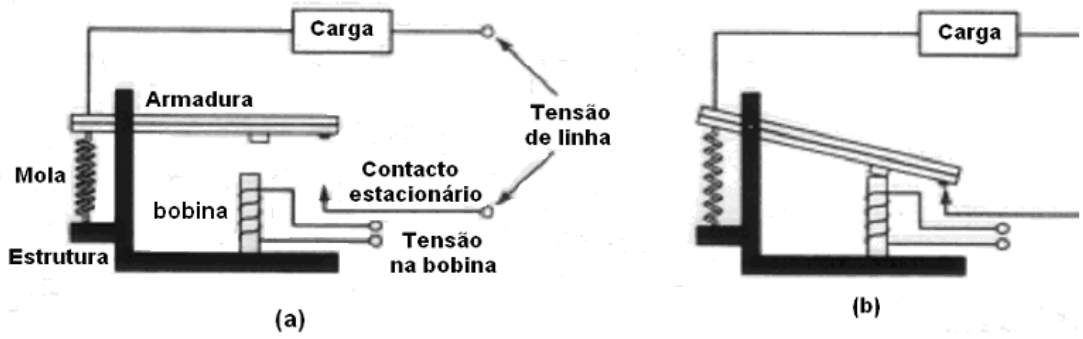


Fig. 3.1 - Esquema funcional de um relé.

A bobina ao ser percorrida por corrente vai originar um campo magnético. O contacto mecânico associado à bobina irá deixar o seu estado de repouso, proporcionado pela mola, e fechar o circuito que alimenta a carga a controlar. A este circuito dá-se a designação de circuito de comando.

Até à introdução dos PLC's, o controlo de aplicações industriais era feito com este tipo de dispositivo.

Muitas falhas que ocorriam eram inerentes do facto de se tratar de um dispositivo electromecânico, e portanto, sujeito a problemas associados à *durabilidade*; *fiabilidade*; e *versatilidade*, por exemplo.

3.2 - Evolução histórica dos PLC's

Os PLC's ("*Programmable Logic Controllers*", ou "*controladores lógicos programáveis*") são elementos fundamentais dos modernos sistemas de automação industrial.

Hoje em dia os PLC's podem desempenhar funções de controlo local de baixo nível de vários subsistemas, coordenação geral do sistema de automação industrial, aquisição e processamento de dados, gestão de comunicações, etc.

No início entretanto, os PLC's pretendiam ser uma alternativa mais flexível à lógica eléctrica e baseada em temporizadores (*timers*), que era vulgar nos paneis de controlo.

O PLC foi inicialmente concebido em 1968. O trabalho foi iniciado por um grupo de engenheiros pertencentes à divisão Hydramatic da General Motors (GM).

Os seguintes critérios foram inicialmente estabelecidos pela GM para a primeira geração de PLC's:

- A máquina deveria ser facilmente programada.
- A aplicação de *software* deveria poder ser modificada facilmente, de preferência na fábrica.
- Deveria ser constituída por módulos, de fácil substituição, com o objectivo de aumentar a fiabilidade, a manutenção e a funcionalidade.
- O espaço ocupado por um destes aparelhos deveria ser reduzido.
- O aparelho deveria poder ser capaz de comunicar com uma central remota.
- O custo final deveria ser competitivo com a tecnologia em uso na época (controlo por relé).

Os PLC's foram portanto originalmente concebidos para providenciarem uma maior flexibilidade no controlo baseado na execução de instruções lógicas.

Além disso, maiores vantagens foram possíveis adoptando a linguagem de programação Ladder, simplificando a manutenção, reduzindo os custos de concretização, e simplificando a introdução de alterações.

Existe um conjunto de vantagens, introduzidas pela utilização de PLC, que podemos resumir nos pontos seguintes:

- os PLC são fáceis de programar e instalar;
- a velocidade de funcionamento, a que normalmente operam, é em muito superior à de qualquer sistema de controlo electromecânico;
- o acesso aos PLC, com o objectivo de proceder a alterações, pode ser vedado utilizando quer uma chave quer *passwords*;
- suporte sistemas paralelo de monitorização do estado de funcionamento do controlador ou do processo;
- a capacidade para suportar ambientes agressivos, como altas temperaturas ou humidades, proporcionada por revestimentos especiais é muito superior à dos sistemas electromecânicos.

Com a sua popularidade e facilidade de programação, os PLC's evoluíram muito da década de 1970 até o momento.

Hoje existem vários tipos de PLC's, dimensionado para aplicações diferentes, sendo os mais poderosos verdadeiros computadores industriais.

3.3 - Constituição e princípio de funcionamento dos PLC's

Independentemente do fabricante, do peso, da complexidade, ou dos custos, todos os PLC's possuem algumas componentes comuns.

Alguns destes itens são exclusivamente concretizados em *hardware*, outros, podem representar características funcionais ou *software*.

Todos os PLC possuem interfaces de saída e entrada de dados, memória, uma fonte de alimentação, e alguma forma expedita de construir a aplicação de *software*. Estas funções estão representadas esquematicamente na Fig. 3.2.

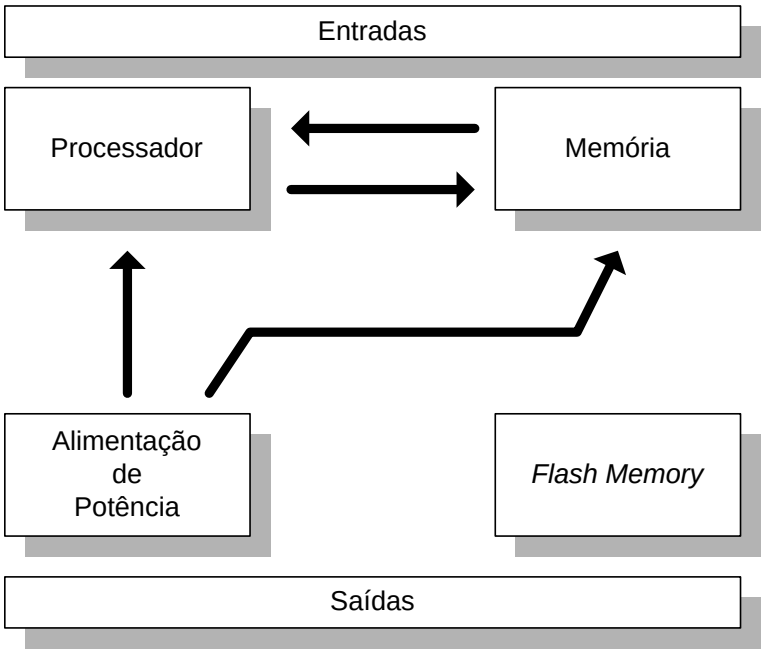


Fig. 3.2 - Blocos funcionais do PLC.

Entradas

A interface de entrada fornece uma ligação ao processo a ser controlado. A função principal deste módulo é receber e converter os sinais recebidos do processo num formato que possa ser usado pelo processador (CPU).

Esta tarefa consiste em converter sinais de diferentes tipos (corrente, tensão) e diferentes amplitudes num formato discreto único.

Este módulo é, como regra geral, expansível adicionando mais módulos para aumentar o número de entradas à medida das necessidades do processo.

O número de entradas suportadas está limitado pelo processador (CPU) e pela quantidade de memória disponível.

Saídas

A interface de saída executa a tarefa contrária da interface de entrada. Este módulo recebe sinais do processador (CPU), transformando-os num formato apropriado a executar as acções de controlo no processo.

Também aqui há a registar a característica modular do PLC, que permite que o número de saídas possa ser expandido, encontrando-se, no entanto, limitado pelas mesmas razões do módulo de entrada.

Processador (CPU)

O processador (CPU) e as memórias fornecem a inteligência central do PLC.

A informação fundamental é armazenada na memória, formando conjuntos de bits, designados por palavras (*words*).

Cada palavra armazenada na memória é um pedaço de dados, uma instrução, ou parte de uma instrução.

Os dados vêm do módulo de entrada e, baseando-se no programa armazenado, o processador (CPU) executa decisões lógicas actuando as saídas.

O processador (CPU) continuamente procura no programa armazenado a acção executar.

A memória também é utilizada para armazenar informação temporária e necessária apenas durante a execução de algumas instruções.

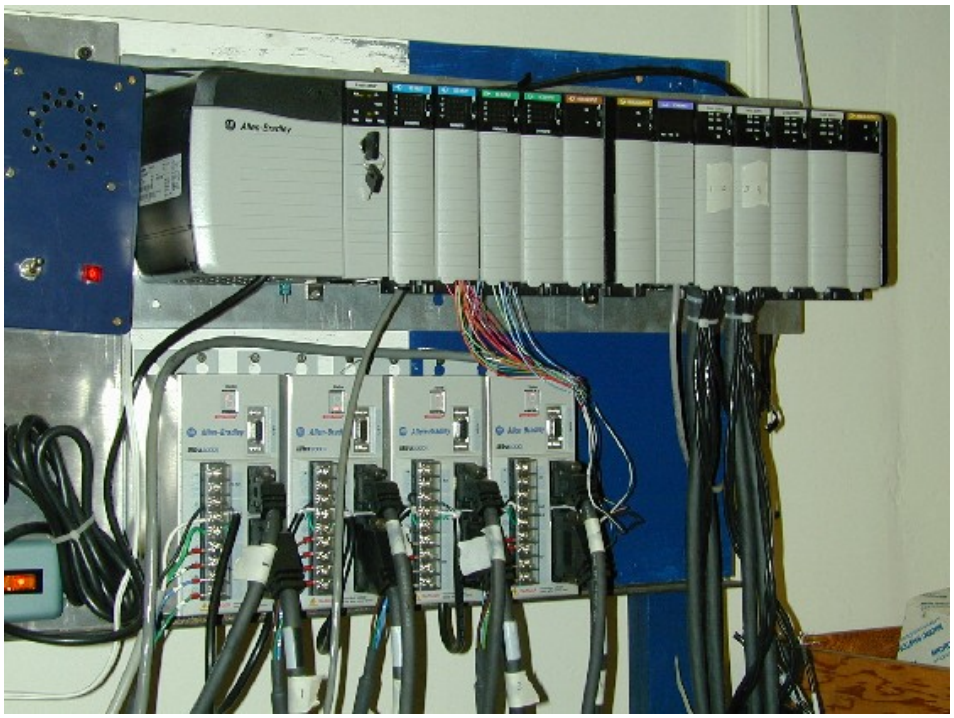


Fig. 3.3 - Controladores PLC's.

Memória Flash

A memória *flash* ("*Flash Memory*") é a memória não volátil onde reside o sistema operativo do PLC.

O conteúdo da memória *flash* é não volátil, logo não requer qualquer tipo de componente, ou alimentação externa.

O sistema operativo residente na memória *flash* é, na realidade, formado por um conjunto de programas supervisores que fornecem ao PLC identidade própria.

As tarefas executadas normalmente por este sistema operativo são as seguintes:

- definem a linguagem em que a aplicação é escrita;
- a memória para fins específicos;
- determinam a estrutura na qual o PLC armazena e trata toda a informação.

Programação

As linguagens de programação merecem uma referência especial. Elas podem apresentar formas diferentes. Como todas as linguagens, a linguagem de programação de um PLC possui uma gramática, uma sintaxe, e um vocabulário próprio que permite ao utilizador escrever um programa que indica ao processador (CPU) qual a tarefa a desempenhar.

Cada fabricante de PLC pode usar uma linguagem própria para realizar esta tarefa, mas como regra geral são respeitadas certas semelhanças entre elas.

Muitas das linguagens de programação de PLC são baseadas na linguagem Ladder que tem as suas raízes na lógica de relé.

Esta linguagem, para além dos contactos normais, permite a introdução de funções matemáticas, de controlo analógico, operações de contagem e temporização, etc.

Outras linguagem alternativas usam representações Booleanas como base da programação.

Temos também, de uso muito corrente, a utilização de linguagens de programação onde, através da construção de um fluxograma, se representa a interligação existente entre as variáveis do processo e a sua evolução.

A programação do PLC pode ser realizada normalmente através de uma de duas formas alternativas:

- Na primeira é utilizado um *software* específico, a ser executado no computador pessoal, que permite comunicar com o PLC por intermédio de uma ligação série.
- Na segunda forma alternativa é utilizada uma pequena consola com capacidade de comunicação e que inclui funcionalidades que permitem programar o PLC.

A primeira solução é mais agradável de utilizar, no entanto, obriga a ligar o PLC ao computador pessoal, o que nem sempre é possível, nomeadamente em instalações fabris.

Entretanto, quando se deseja programar o PLC sem o retirar da instalação é utilizada a segunda alternativa, ou seja, a consola de programação.

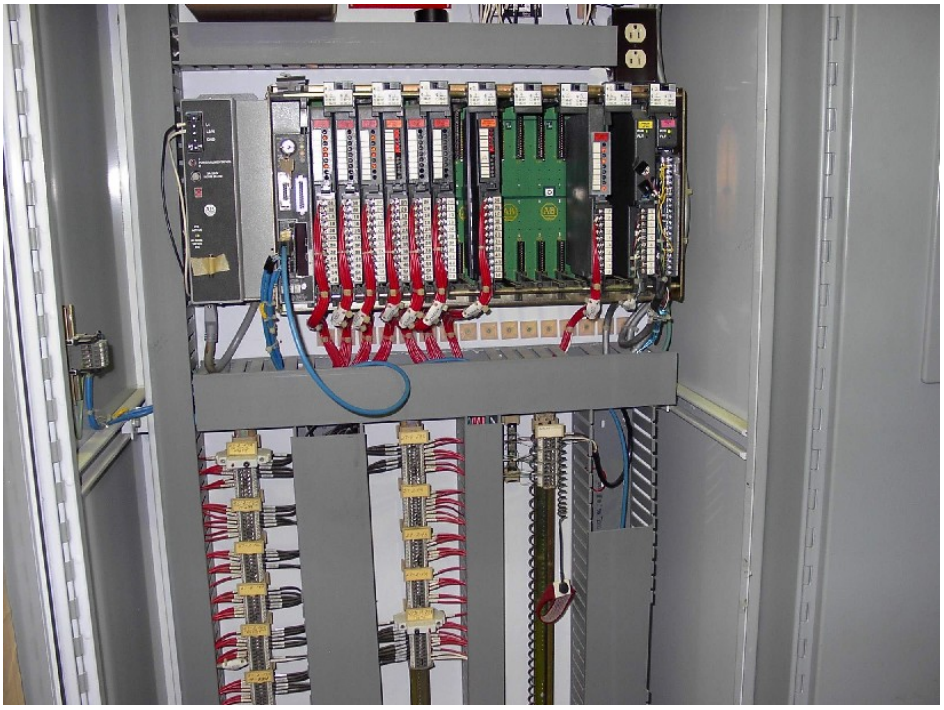


Fig. 3.4 - Controladores PLC's.

3.4 - *Modicon Micro Programmable Logic Controllers*

Tempo de varrimento (*scan time*)

Para que a CPU possa executar a aplicação de *software* desenvolvida pelo utilizador, ela deve poder examinar o que ocorre nas diversas memórias do sistema.

A este procedimento de avaliar o estado de todas as entradas e saídas, cruzando essa informação com a aplicação de *software*, dá-se a designação de função de varrimento (*scan function*) do PLC.

Este parâmetro é muito importante num PLC. Por exemplo, se uma entrada alterar o seu estado duas vezes durante um varrimento, o PLC nunca poderá detectar essa ocorrência.

Esta situação sucede sempre que a comutação de estado seja mais rápida do que a sequência de varrimento.

Se por exemplo, a CPU demorar 7 milissegundos no varrimento de um programa, e uma entrada comutar o seu estado com uma periodicidade de 3 milissegundos, a CPU não irá detectar estas ocorrências. Para resolver esta situação muitos PLC possuem tempos de varrimento ajustáveis.

Cada instrução que dá entrada num programa requer um certo tempo para que seja devidamente processada. Esta quantidade de tempo irá depender da instrução em causa. Por exemplo, o processador irá demorar menos tempo a ler o estado de uma entrada, do que a recolher o acumulado de tempo num temporizador.

Um dos factores que mais condiciona o tempo de varrimento é a dimensão do programa, quanto maior este for mais tempo o PLC demora a executar a função de varrimento.

Também o facto de o PLC poder estar ligado a um terminal de programação pode aumentar o tempo de varrimento.

Nesta parcela de tempo podemos distinguir diferentes tarefas com tempos de execução próprios, tais como:

- Tempo de resolução do programa lógico (*Logic solve time*).
- Actualização das entradas e saídas do PLC (*IO servicing time*).
- Gestão do sistema operativo (*System Overhead time*).

O tempo máximo que a família de PLC's AEG-*Modicon* pode demorar a varrer um programa é de 250 milissegundos. Se isso não acontecer, um *watchdog timer* interno é activado, dando origem a um erro interno de *timeout*, sinalizado no painel de programação. Evita-se assim que o PLC entre num estado de funcionamento instável.

Por necessidade de demonstração utilizou-se aqui, a título de exemplo, os PLC's da família AEG-*Modicon*. Há entretanto opções equivalentes no mercado de outras marcas como: SIEMENS, Omron, Beck, Telemecanique, GE, Allen-Bradley, etc. e para os quais se aplicam todos os conceitos apresentados aqui.

Os editores de Ladder e os símbolos usados pelos PLC's são muito parecidos de fabricante para fabricante, com poucas diferenças em alguns pormenores que não são significativas.

Alocação de memória

O sistema operativo do PLC divide a memória disponível nas seguintes classes funcionais:

- *User data memory* – Memória disponível para as variáveis que armazenam a informação necessária à execução do programa desenvolvido pelo utilizador.
- *System configuration memory* – Memória onde são armazenados os parâmetros de configuração do sistema, tais como: os endereços de memória correspondentes às entradas e às saídas; a quantidade de memória disponível; a localização de endereços específicos; etc.
- *User programan memory* – Espaço de memória reservado à aplicação de *software* desenvolvida pelo utilizador.

O PLC usa um sistema de referência numérica para lidar com os endereços de memória. Cada endereço de memória tem um dígito que identifica o tipo a que diz respeito, seguido por mais quatro dígitos que definem a sua localização. Esta organização pode ser observada na Tab. 3.1.

Referência	Descrição	bits
0xxxx	saída discreta	1
1xxxx	entrada discreta	2
3xxxx	registo de entrada	16
4xxxx	registo de saída	16

Tab. 3.1 – Referência das entradas e saídas.

A relação entre as entradas físicas e os respectivos endereços de memórias são as seguintes:

Entrada	01	02	03	....	12
Endereço	10001	10002	10003	....	10012

A relação entre as saídas físicas e os respectivos endereços de memórias são as seguintes:

Saída	01	02	03	....	16
Endereço	10001	10002	10003	....	10016

A memória de dados do utilizador, a memória de programa do utilizador, e a configuração do sistema podem ser mantidas de três formas diferentes sempre que ocorrer uma falha de energia, a saber:

- Pilhas de lítio – duração de um ano aproximadamente.
- Bateria capacitiva – Setenta e duas horas aproximadamente.
- Gravação prévia na área reservada da memória *flash* do PLC.

Sequência de configuração da memória

No arrancar, o PLC necessita de possuir informação válida sobre a sua configuração. Existem diferentes alternativas para a origem desses dados, tal como está representado no diagrama da Fig. 3.5.

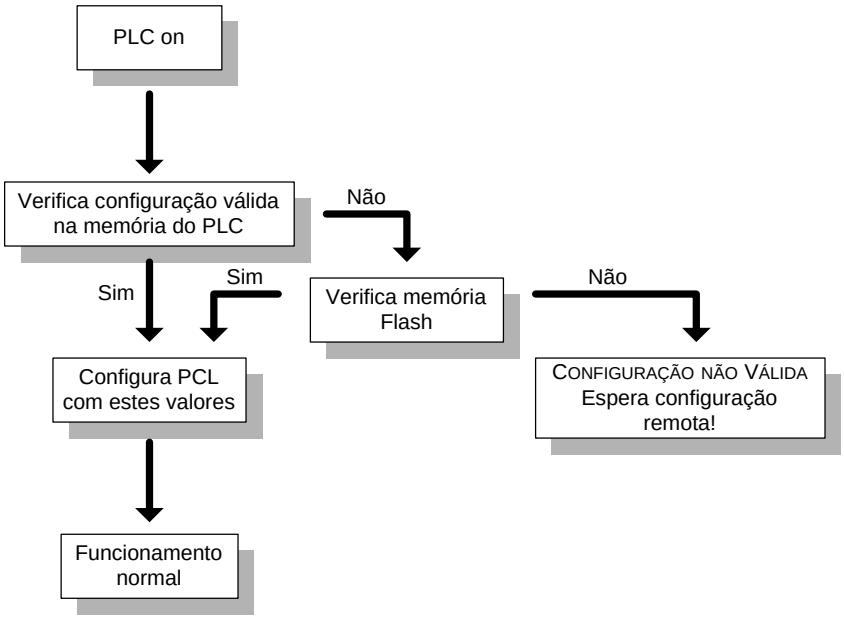


Fig. 3.5 - Configuração da memória.

Processo de arranque e configuração

A sequência de arranque de um PLC é em tudo semelhante à de um computador pessoal. O diagrama da Fig. 3.6 demonstra os passos percorridos na configuração do dispositivo.

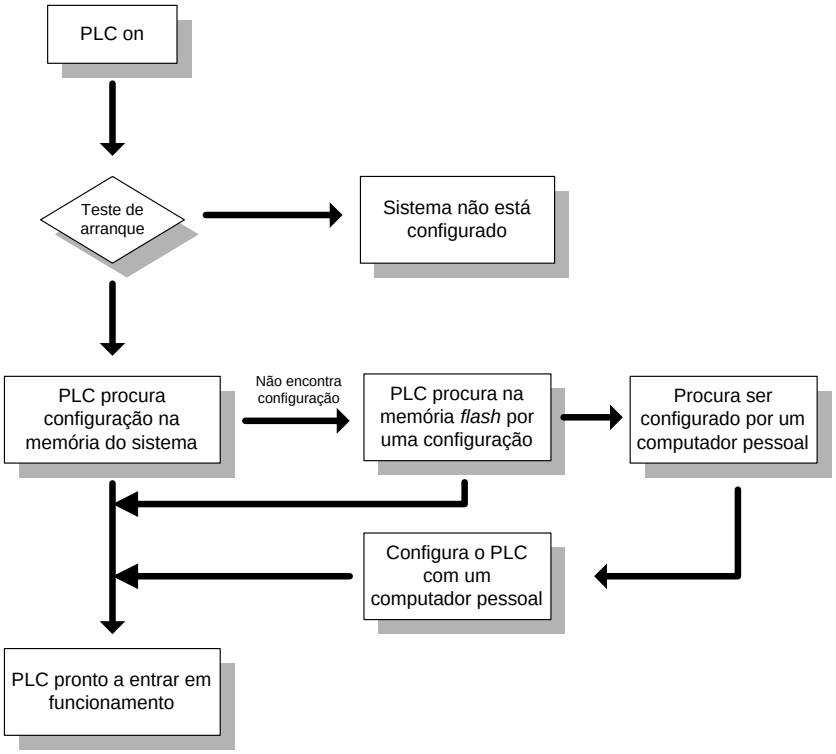


Fig. 3.6 - Processo de arranque do PLC.

PLC status display

O PLC possui um conjunto de LED que permite a um utilizador avaliar localmente o estado de funcionamento da máquina, sem necessidade de qualquer tipo de equipamento extra. A Tabela 3.2 refere a função de cada um deles.

LED	Função
power ok	Uma luz verde acende quando alimentação ok
ready	Ambar está *on* quando o PLC arranca com sucesso os seus procedimentos de *power-up* e permanece *on* enquanto o PLC está ok.
run	Um LED verde está *on* quando o PLC está a correr um programa lógico. Fica intermitente quando o PLC está alimentado mas não tem uma configuração correcta.
Baterry low	Um LED verde fica *on* quando a bateria necessita ser mudada.
exp. link	Um LED verde fica *on* quando uma comunicação válida ocorre no link de exp I/O. fica intermitente quando erros ocorrem na *link*.
comm 1	Um LED verde está intermitente quando o porto série RS232 está a ser utilizado com uma comunicação.
comm 2	Um LED verde está intermitente quando o porto série RS232 está a ser utilizado com uma comunicação.

Tab. 3.2 – Display de estado do PLC.

3.5 - Programação em linguagem Ladder

Introdução à linguagem Ladder

A programação em linguagem Ladder é uma ferramenta usada para descrever o formato de diagramas esquemáticos introduzidos num PLC. A linguagem usa dois elementos básicos: instruções lógicas de relé e instruções para transferência de dados.

Nesta secção será examinada a utilização de instruções lógicas de relé discretas. Este conjunto de instruções lógicas permite que a linguagem Ladder possa substituir, de uma forma eficaz, o controlo realizado exclusivamente com relés.

Um circuito de lógica Ladder consiste numa rede formada por linhas, nas quais deve existir continuidade para que se possa activar a respectiva saída. Estas saídas são controladas pela combinação de estados das entradas e saídas.

As condições podem ser ligadas em série, paralelo, ou série-paralelo, a fim de construir a lógica necessária.

Desta forma, uma rede Ladder define a estratégia de comando que controla as saídas do PLC. Ao contrário da lógica de relé, uma rede Ladder não representa ligações físicas.

Como exemplo, o circuito representado na Fig. 3.7 é um diagrama que utiliza relés para concretizar uma sequência de paragem e arranque de um relé CR1, a que corresponde o diagrama da Ladder da Fig. 3.8. Neste novo esquema cada uma das entradas e saídas está identificada por um endereço de memória próprio.

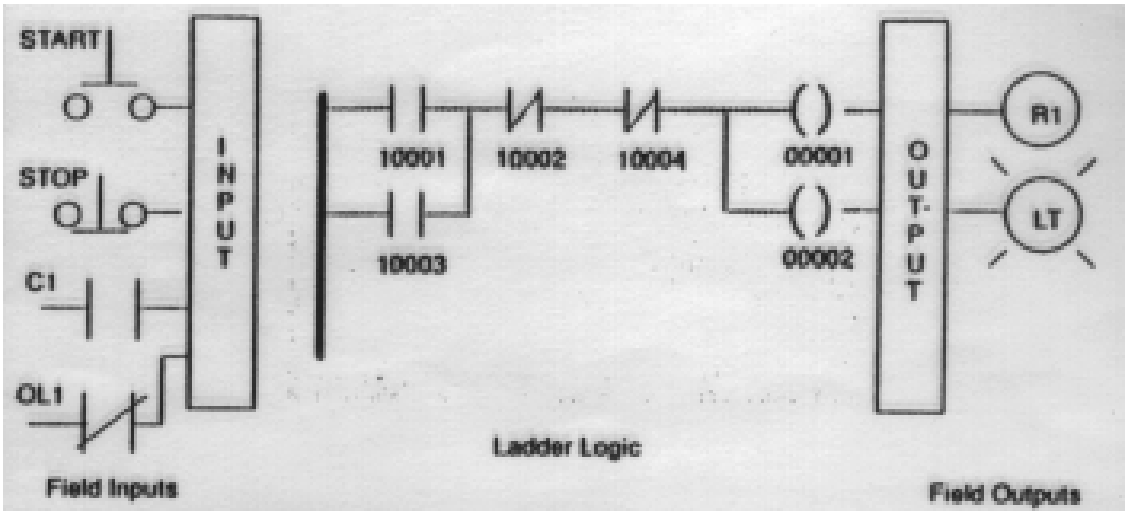


Fig. 3.7 - Circuito de paragem e arranque com relé.

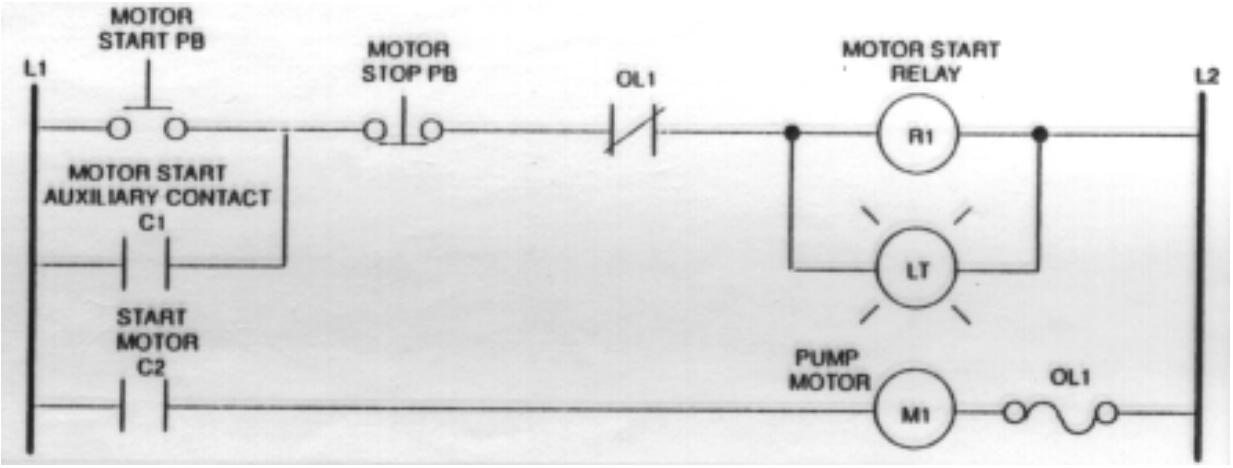


Fig. 3.8 - Circuito de paragem e arranque em lógica Ladder.

Alguns princípios básicos relativos à lógica Ladder devem ser referidos:

- Todos os símbolos que denotem saídas devem estar localizados o mais à direita possível.
- Todos os símbolos que denotem contactos devem estar localizados do lado esquerdo.
- É possível ligar por intermédio de caminhos horizontais e caminhos verticais os diversos componentes.
- Todos os símbolos são representados no seu estado normal. Ou seja, os contactos normalmente abertos encontram-se abertos; e os contactos normalmente fechados encontram-se fechados. Só ocorrem comutações de estado quando o contacto for alimentado.

Conjunto de Instruções em Ladder

Lógica de relé

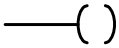
Estas funções permitem manipular a informação.


Contacto normalmente aberto


Contacto normalmente fechado

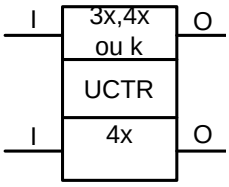

Contacto de transição positiva


Contacto de transição negativa

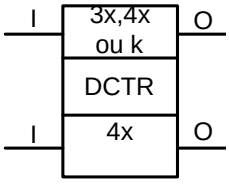

Relé de saída

Contadores

Os contadores são elementos básicos de qualquer PLC. Contar eventos lógicos, de IO, etc. é uma das actividades de rotina de um programa de PLC.



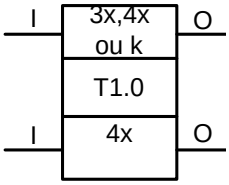
Contador incremental



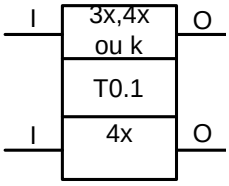
Contador decremental

Temporizadores (*timers*)

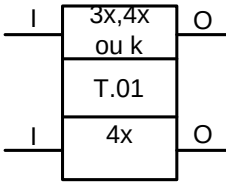
Todos os PLC's possuem vários tipos de temporizadores (*timers*) diferentes, adaptados avarias situações de utilização e que permitem rotinas temporizadas fundamentais em qualquer aplicação industrial.



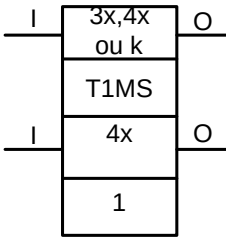
Temporizador com base temporal de 1 seg.



Temporizador com base temporal de 0.1 seg.



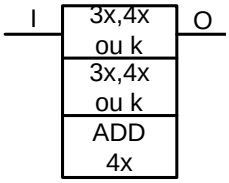
Temporizador com base temporal de 0.01 seg.



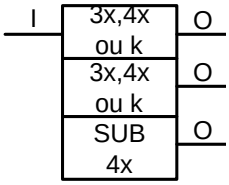
Temporizador com base temporal de 1 milissegundo.

Matemática Inteira

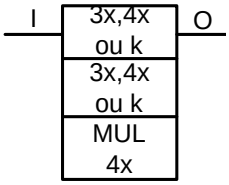
Funções matemáticas para álgebra de inteiros.



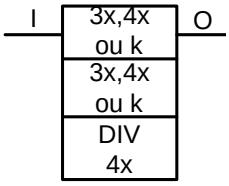
Adição Inteira



Subtracção Inteira



Multiplicação Inteira

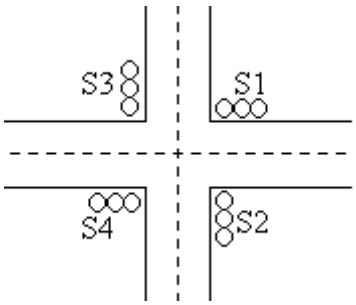


Divisão Inteira

3.6 - Exemplos de aplicação

Exemplo 1 – Pretende-se implementar, num cruzamento rodoviário, um conjunto de semáforos que controlem o trânsito.

A disposição e a identificação dos semáforos é a apresentada no esquema. O objectivo deste conjunto de semáforos é o de apenas permitir o trânsito, à vez, de veículos quer na direcção horizontal quer na direcção vertical. Os tempos mínimos a respeitar, para a implementação dos semáforos, são fornecidos na tabela



Cor	Semáforos S3 e S2	Semáforos S1 e S4
Verde	2 min	2 min
Amarelo	5 seg	5 seg
Vermelho	2 min	2 min

O trabalho deve contemplar os seguintes aspectos:

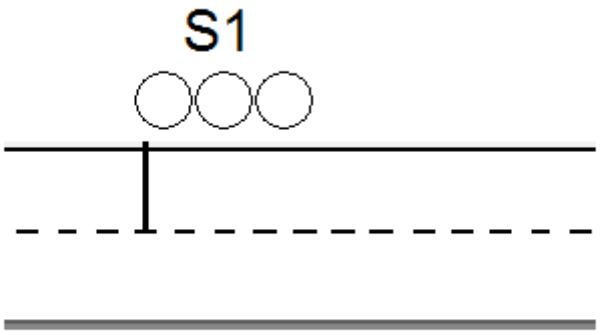
- Esquema funcional de todo o sistema.
- Entradas do sistema.
- Saídas do sistema.
- Implementação do programa em LADDER.

Exemplo 2 – No troço de estrada apresentado existe a necessidade de controlar a velocidade dos veículos que nela circulam.

Para isso, é pedida a implementação de um sistema que controle a velocidade dos veículos, através do accionamento de semáforos.

Quando a velocidade de um veículo é superior ao limite fixado, o semáforo deve passar de verde a vermelho, passando antes pelo amarelo.

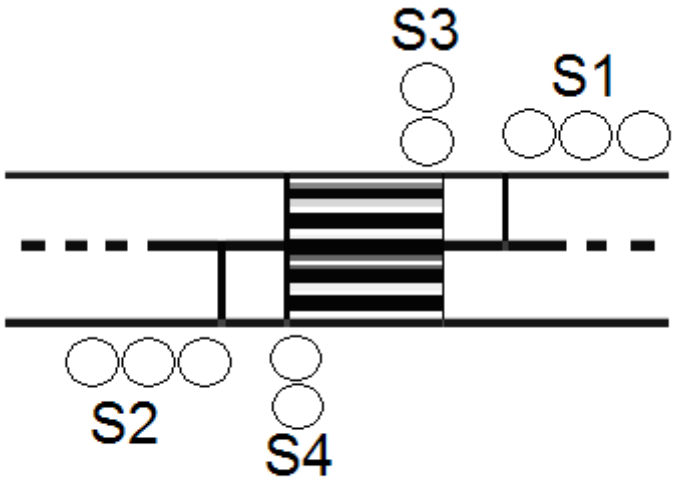
O condutor deste veículo deve então ser obrigado a efectuar uma paragem de 30 segundos, só podendo arrancar após efectuar esse compasso de espera.



Ainda no mesmo troço de estrada vai ser colocada uma passadeira para peões.

Quando estes pretenderem atravessar a faixa de rodagem devem accionar um botão de pressão, será então desencadeado o processo de paragem dos veículos.

Estes devem então permanecer parados durante um período igual a 30 segundos.



O trabalho deve contemplar os seguintes aspectos:

- Esquema funcional de todo o sistema.
- Entradas do sistema.
- Saídas do sistema.
- Implementação do programa em LADDER

Exemplo 3 – O processo de fabrico de um determinado produto passa pela sua cozedura durante 2 minutos.

Para retirar um melhor rendimento do forno, são cozido ao mesmo tempo 4 unidades.

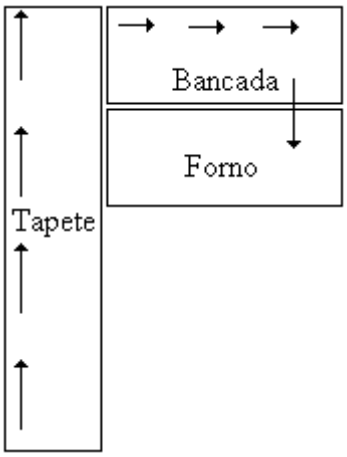
Cada uma das unidades é colocada à vez num tapete rolante, que a transporta até a uma bancada em frente ao forno.

Quando existem 4 unidades em frente ao forno a sua porta é aberta, as 4 unidades são empurradas para dentro do forno. Sendo a porta fechada de seguida.

No esquema está representado todo o processo de transporte.

- tapete rolante é accionado por um motor de indução.
- forno é eléctrico, o que permite controlar o seu estado de funcionamento.

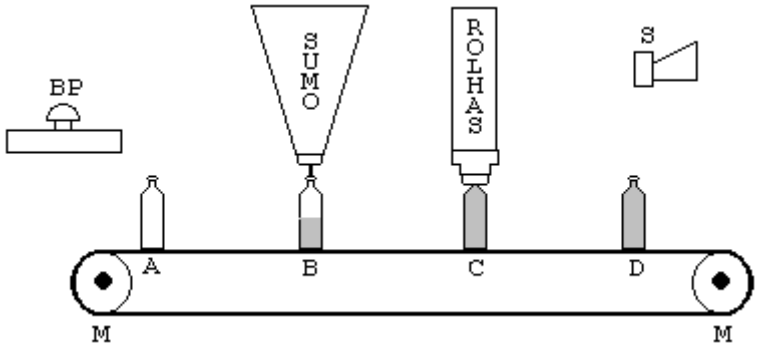
A passagem dos objectos do tapete para a bancada, e desta para o forno, é feito por accionamento de êmbolos pneumáticos.



O trabalho deve contemplar os seguintes aspectos:

- Esquema funcional de todo o sistema.
- Entradas do sistema.
- Saídas do sistema.
- Implementação do programa em LADDER

Exemplo 4 – Pretende-se controlar a cadeia de engarrafamento de sumos apresentada na figura seguinte.



Um operador ao colocar uma garrafa vazia no ponto (A) prima no botão de pressão (BP).

Os dois motores (M) só devem começar a rodar se existir uma garrafa em (A).

Quando a garrafa chegar a (B) os motores (M) devem parar durante 30 segundos, de forma a permitir o enchimento da garrafa.

Quando a garrafa chega a (C) os motores (M) devem parar e a máquina de arrolhar deve ser accionada para colocar uma rolha na garrafa.

Ao chegar a (D) os motores (M) são parados e deve ser accionada uma sirene (S).

O trabalho deve contemplar os seguintes aspectos:

- Esquema funcional de todo o sistema.
- Entradas do sistema.
- Saídas do sistema.
- Implementação do programa em LADDER

Exemplo 5 – Considere um parque de estacionamento automóvel instalado num silo de dois pisos.

Por motivos de segurança cada nível não pode conter mais que 100 automóveis.

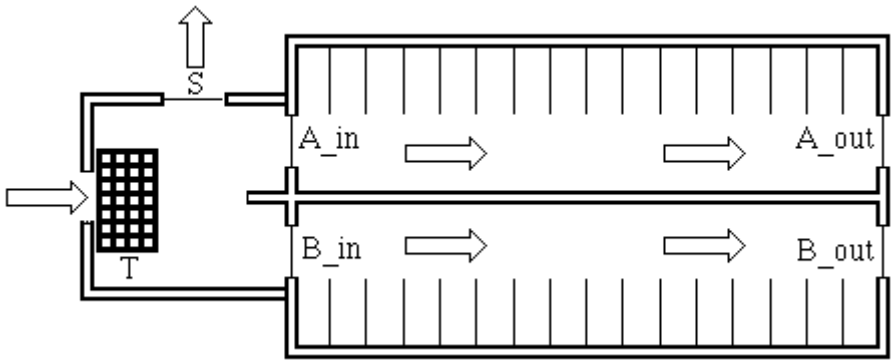
Quando a lotação do nível superior chega a 70 % do limite o acesso a este nível é vedado, só voltando a reabrir quando a lotação do nível inferior estiver completa.

Quando a lotação dos dois pisos estiver esgotada, é vedado o acesso dos automóveis ao silo, só voltando a reabrir quando houver vaga para mais automóveis.

O trabalho deve contemplar os seguintes aspectos:

- Esquema funcional de todo o sistema.
- Entradas do sistema.
- Saídas do sistema.
- Implementação do programa em LADDER

Exemplo 6 – Na figura seguinte está representado um estacionamento de automóveis.



O piso A está reservado a veículos ligeiros e o piso B está reservado a veículos pesados.

A categoria do veículo é atribuída consoante o peso medido pela balança (T). Após a pesagem, se o peso do veiculo for inferior ou igual a 3000 kg é aberto o acesso (A_in) ao piso A, se o peso for superior ao limite anterior é aberto o acesso (B_in) ao piso B.

Quando o estacionamento esgotar em qualquer um dos pisos este é vedado e o veículo deve seguir, sem poder estacionar, pela porta (S). As saídas dos estacionamentos (A_out e B_out) devem ser accionadas automaticamente.

O trabalho deve contemplar os seguintes aspectos:

- Esquema funcional de todo o sistema.
- Entradas do sistema.
- Saídas do sistema.
- Implementação do programa em LADDER

Exemplo 7 – Pretende-se construir uma caixa automática de venda de bebidas refrigerantes em lata.

Os requisitos básicos da máquina são os seguintes:

- pagamento é feito em moedas, e não dá troco;
- utilizador deve escolher, após a introdução da quantia correcta e autorização da máquina, a bebida que deseja;
- deve sinalizar quais as bebidas esgotadas.

O trabalho deve contemplar os seguintes aspectos:

- Esquema funcional de todo o sistema.
- Entradas do sistema.
- Saídas do sistema.
- Implementação do programa em LADDER

Exemplo 8 - Pretende-se construir uma máquina de café.

O utilizador deve inicialmente introduzir uma moeda, após o que deve escolher num teclado, constituído por 3 teclas: se deseja

- Tecla 1 - café cheio;
- Tecla 2 - café curto; ou
- Tecla 3 - água quente para fazer chá.

O funcionamento da máquina deve respeitar estes requisitos.

O trabalho deve contemplar os seguintes aspectos:

- Esquema funcional de todo o sistema.
- Entradas do sistema.
- Saídas do sistema.
- Implementação do programa em LADDER