

Nº Doc.: 0682-228
Rev.: 00
Data: 13/08/2002

Evolução dos Sistemas de Controle

1.	EVOLUÇÃO DOS SISTEMAS DE CONTROLE	1
1.1	Introdução	1
1.2	História dos Controladores de Processos	1
1.3	Aplicações de Controle	3
1.3.1	Monitoração	3
1.3.2	Sequenciamento.....	3
1.3.3	Malhas de Controle.....	4
1.4	Lógica a Relés	4
1.5	Computadores para Controle de Processos.....	5
1.5.1	Métodos de Programação	6
1.5.2	Código de Máquina e Assembler.....	6
1.5.3	Compilação e Interpretação.....	7
1.6	Controladores Programáveis.....	8
1.6.1	Unidades de Entradas e Saídas.....	9
1.6.2	Métodos de Programação	9
1.6.3	Ferramentas de Programação baseadas em Computadores.....	10
1.6.4	Execução Cíclica.....	11
1.6.5	Sistemas Distribuídos	12
1.6.6	Soft PLC	12
2.	MOTIVAÇÃO PARA OS SISTEMAS ABERTOS	13
2.1	Diferentes Dialectos de Programação	13
2.2	Qualidade de <i>Software</i>	13
2.3	Custo do <i>Software</i>	14
2.4	Portabilidade de Aplicações.....	15
2.5	Reutilização de <i>Software</i>	15
2.6	Comunicação com outros Sistemas.....	16

Figuras

Figura 1 – Componentes de um sistema de controle industrial.....	1
Figura 2 – Evolução dos sistemas de controle desde o final do século 19.....	3
Figura 3 – Diagrama de blocos de uma malha de controle.....	4
Figura 4 – Três exemplos de lógica implementada com relés.	5
Figura 5 – Na programação de baixo nível, são usados os programas para edição, montagem e depuração no processo de geração do código de máquina.	7
Figura 6 – Programas escritos em linguagens de alto nível são independentes do hardware e são traduzidos para o código de máquina pelo programa compilador.....	8
Figura 7 – Componentes de um controlador programável.....	9
Figura 8 – Exemplo de programa utilizando Diagrama Ladder e Lista de Instruções.	10
Figura 9 – Varredura típica de um Controlador Programável.....	11
Figura 10 – Custo do Hardware versus Software.....	14

1. EVOLUÇÃO DOS SISTEMAS DE CONTROLE

1.1 Introdução

Quase toda planta industrial precisa de algum tipo de controlador para garantir uma operação segura e economicamente viável. No nível mais simples, uma planta pode consistir basicamente de um motor elétrico acionando um ventilador para controlar a temperatura de um sala. No extremo oposto, uma planta pode ser reator nuclear para produção de energia para milhares de pessoas. Apesar do tamanho e complexidade, todos sistemas de controle podem ser divididos em três partes com funções bem definidas, os transdutores, os controladores e os atuadores.

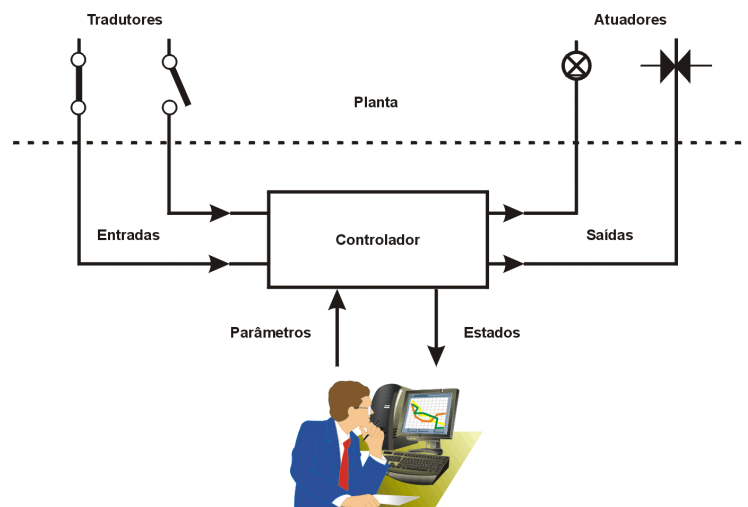


Figura 1 – Componentes de um sistema de controle industrial.

O controlador monitora o estado real do processo de uma planta através de um número de transdutores. Os transdutores convertem as grandezas físicas em sinais normalmente elétricos, os quais são conectados com as entradas dos controladores. Transdutores digitais medem variáveis com estados distintos, tais como ligado/desligado ou alto/baixo, enquanto os transdutores analógicos medem variáveis com uma faixa contínua, tais como pressão, temperatura, vazão ou nível.

Com base nos estados das suas entradas, o controlador utiliza um algoritmo de controle embutido para calcular os estados das suas saídas. Os sinais elétricos das saídas são convertidos para o processo através dos atuadores. Muitos atuadores geram movimentos como válvulas, motores, bombas e outros utilizam energia elétrica e pneumática.

O operador interage com o controlador através dos parâmetros de controle. Alguns controladores podem mostrar o estado do processo através de um *display* ou tela.

1.2 História dos Controladores de Processos

Os primeiros sistemas de controle foram desenvolvidos durante a revolução industrial, no final do século 19. As funções de controle eram implementadas através de engenhosos dispositivos mecânicos, os quais automatizavam algumas tarefas críticas e repetitivas das linhas de montagem da época. Estes dispositivos tinham de ser desenvolvidos para cada tarefa e devido à natureza mecânica dos mesmos, estes tinham uma pequena vida útil.

Nos anos 20, os dispositivos mecânicos foram substituídos pelos relés e contadores. A lógica a relés viabilizou o desenvolvimento de funções de controle mais complexas e sofisticadas. Desde então, os relés têm sido empregados em um grande número de sistemas de controle em todo o mundo. Os relés se mostram uma alternativa de custo viável, especialmente para a automação de pequenas máquinas com um número limitado de transdutores e atuadores. Na indústria moderna, a lógica a relés é raramente adotada para o desenvolvimento de novos sistemas de controle, mas ainda é usada em um grande número de sistemas antigos .

O circuito integrado, CI, proporcionou o desenvolvimento de uma nova geração de sistemas de controle. Em comparação aos relés, os CIs baseados nas tecnologias TTL ou CMOS são muito menores, mais rápidos e possuem uma vida útil muito maior. Em muitos sistemas de controle, que utilizam relés e CIs, a lógica de controle, ou algoritmo, é definida permanentemente pela interligação elétrica. Sistemas com lógica definida pela interligação elétrica são fáceis de implementar, mas é muito difícil e demorado para alterar o comportamento, ou a lógica dos mesmos.

No início dos anos 70, os primeiros computadores comerciais começaram a ser utilizados como controladores em sistemas de controle de grande porte. Devido ao fato do computador ser programável, este proporciona uma grande vantagem em comparação com a lógica por interligação elétrica, utilizada em sistemas com relés e CIs. Os primeiros computadores eram grandes, caros, difíceis de programar e muito sensíveis à utilização em ambientes “hostis” encontrados em muitas plantas industriais. A partir de uma demanda existente na indústria automobilística norte-americana, o *Programmable Logic Controller* (PLC), ou Controlador Lógico Programável (CLP), foi desenvolvido no início dos anos 70. O CLP é um computador projetado para trabalhar no ambiente industrial. Os transdutores e atuadores são conectados a robustos cartões de interface. Comparado com um computador de escritório, os primeiros CLPs tinham um conjunto de instruções reduzido, normalmente somente condições lógicas e não possuíam entradas analógicas, podendo manipular apenas aplicações de controle digital (discreto). Atualmente, as plantas industriais normalmente precisam manipular tanto controle digital quanto malhas analógicas com o mesmo sistema de controle. Estes sistemas são normalmente chamados de *Programmable Controllers*, ou Controladores Programáveis, por não serem limitados a operações com condições lógicas.

As atuais funções de controle existentes em uma planta industrial são normalmente distribuídas entre um número de controladores programáveis, os quais são montados próximos aos equipamentos a serem controlados. Os diferentes controladores são usualmente conectados via rede local (LAN) a um computador supervisor central, o qual gerencia os alarmes, receitas e relatórios.

Atualmente, o operador desempenha um papel importante na indústria moderna, sendo que a maioria das plantas industriais possui um sistema chamado Sistema SCADA (*Supervisory Control And Data Acquisition*). Os sistemas SCADA tem monitores coloridos de alta resolução, com os quais o operador pode selecionar diferentes programas e avaliar a situação do processo produtivo.

A característica da indústria moderna é o aumento do lucro, fazendo com que operador tenha que atuar frequentemente nas funções de controle.

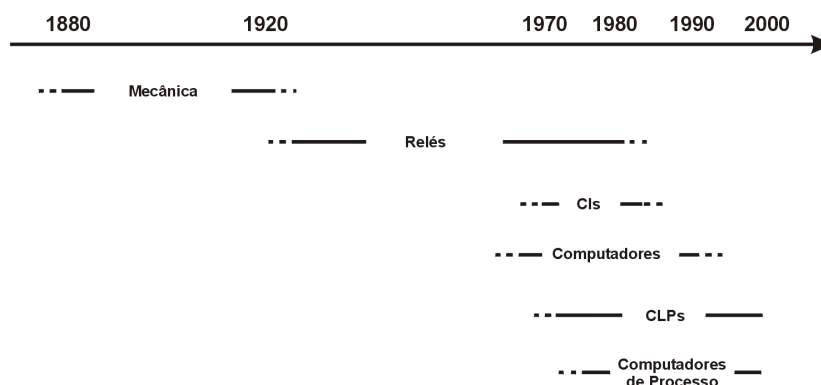


Figura 2 – Evolução dos sistemas de controle desde o final do século 19.

Como o preço dos computadores caiu dramaticamente nos últimos anos, o custo de desenvolvimento e manutenção de *software* tem se tornado o fator predominante dos sistemas de automação.

Com o objetivo de melhorar a qualidade e viabilizar a reutilização de programas, existe cada vez mais pessoas trabalhando com sistemas orientados a objetos. Nestes sistemas, os elementos reais de processo como motores, válvulas e controladores PID são programados através de objetos de *software* armazenados em bibliotecas. Estes objetos são devidamente testados e possuem interfaces de dados padronizadas.

1.3 Aplicações de Controle

Os processos industriais podem facilmente impressionar pela sua complexidade. Entretanto, muitos processos podem ser simplificados através da divisão em sub-processos menores. Estes sub-processos podem ser classificados em 3 categorias que são os subsistemas de monitoração, sequenciamento e malha de controle, os quais são descritos a seguir.

1.3.1 Monitoração

O subsistema de monitoração mostra os estados do processo para o operador e destaca as condições anormais que necessitam que seja tomada alguma ação por parte dele. Os valores medidos de temperatura, pressão, vazão, e etc., são mostrados para o operador através de indicadores, medidores, gráficos de barras ou nas telas do computador. Os sinais também podem ser tratados para as condições de alarme do processo. O sistema faz a indicação de alarmes através de lâmpadas, sinais sonoros e muitas vezes com impressão. Muitos sistemas de monitoração também registram o consumo de matérias-primas e energia com a finalidade de contábeis. O sistema também pode criar advertências automáticas quando os componentes críticos precisam ser substituídos.

1.3.2 Sequenciamento

A grande maioria de todos os sub-processos podem ser descritos através de uma seqüência pré-definida de ações que devem ser executadas em uma determinada ordem. Nestes sistemas, não é possível definir uma combinação de entradas, resultando em determinadas saídas (lógica combinatória). As saídas são dependentes de uma seqüência de entradas já ocorridas. A monitoração da seqüência de ações é possível através de funções do tipo memória.

Os subsistemas de sequenciamento têm uma série de vantagens sobre os sistemas baseados em condições momentâneas das entradas, como ocorre na lógica combinatória. É mais fácil diagnosticar a falha em um transdutor quando este interrompe o progresso de uma seqüência.

1.3.3 Malhas de Controle

Muitos subsistemas manipulam variáveis analógicas como temperatura, vazão ou pressão, as quais devem ser mantidas automaticamente em um determinado valor desejado ou seguir o valor de outro sinal. A Figura 3 apresenta um diagrama de blocos deste sistema.

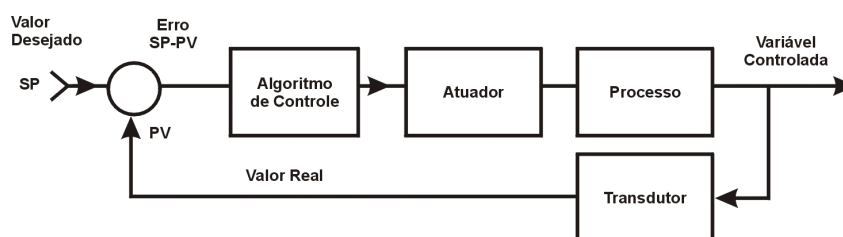


Figura 3 – Diagrama de blocos de uma malha de controle.

A variável de processo chamada de PV (*Process Value*) deve ser mantida no valor desejado denominado de SP (*Setpoint*).

A variável PV é medida por um transdutor e comparado com o valor do SP, gerando um sinal de erro. Este sinal de erro é passado para um algoritmo de controle que calcula o sinal de saída para o atuador com o objetivo corrigir a variável de processo. O algoritmo de controle tentará ajustar o atuador até eliminar o sinal de erro.

Existem vários algoritmos de controle disponíveis, mas o mais usado é o Proporcional-Integral-Derivativo (PID). Desde que a função de controle é executada continuamente, a PV pode então acompanhar o valor do SP.

1.4 Lógica a Relés

Os relés eletromecânicos têm sido um dos componentes mais importantes na evolução dos sistemas de controle. A lógica a relés consiste na associação de diversos relés que são acionados por contatos de transdutores digitais. A função de controle é definida pela forma como os contatos são associados para comandar a bobina do relé.

Todas as bobinas de relés são usadas para ativar um ou mais contatos de saída. Estes contatos são interligados com os atuadores do processo. Se um dos contatos do relé é utilizado com uma entrada da mesma lógica é possível construir um circuito de selo, ou uma função do tipo memória.

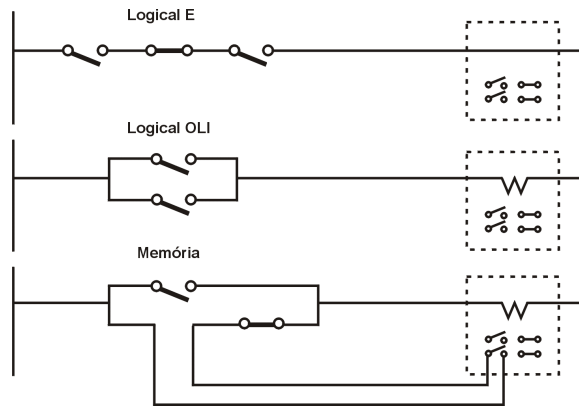


Figura 4 – Três exemplos de lógica implementada com relés.

Um sistema de controle baseado em relés pode conter facilmente dezenas a milhares de relés facilmente. Os relés e os cabos necessários para interligação dos mesmos são acondicionados em armários.

A função lógica de um sistema de controle baseado em relés é descrita no diagrama chamado de *Ladder*, devido à sua semelhança com uma escada, mostrando como os contatos dos transdutores e os atuadores são interligados eletricamente. O Diagrama *Ladder* não somente descreve a função lógica como é usado como desenho para a montagem dos armários. Desde que os relés têm custo considerável e é necessário um tempo elevado para a interligação elétrica, o custo total de um sistema de controle baseado em relés é determinado pelo número de relés utilizados. Em plantas muito grandes, o número limitado de contatos disponíveis para os transdutores e relés normalmente representa uma dificuldade a mais do projeto de engenharia.

A experiência mostra que é fácil implementar um sistema com poucos relés, sendo quando a complexidade aumenta, torna-se necessário engenheiros bem experientes.

Uma característica positiva dos sistemas baseados em relés está na descentralização do controle em um grande número de relés discretos. Como os relés são dispositivos eletromagnéticos, eles têm uma vida útil limitada. Portanto, os sistemas baseados em relés necessitam de uma manutenção contínua. Outra desvantagem destes sistemas é o tempo gasto para alterações na lógica de um sistema existente. Atualmente, os sistemas baseados em relés só são viáveis em sistemas com poucas entradas e saídas e em plantas com um elevado nível de interferência elétrica, onde computadores e CLPs não podem ser utilizados.

1.5 Computadores para Controle de Processos

Os primeiros computadores foram desenvolvidos nos anos 50, os quais eram muito grandes e de custo elevado. Estes computadores foram utilizados principalmente em tarefas administrativas como folha salarial, contabilidade e operações bancárias. As operações desempenhadas eram muitas processos por batelada.

Os microprocessadores foram desenvolvidos nos anos 70, iniciando uma dramática revolução que resultou em computadores menores e mais baratos. Nos anos 70, muitos sistemas de controle foram desenvolvidos utilizando microprocessadores como controladores. A maior vantagem dos computadores em comparação com a lógica por interligação elétrica, é que a função de controle programada pode ser modificada facilmente. Computadores também podem facilmente manipular grandes volumes de dados e executar inúmeros cálculos. Um computador padrão no entanto não é preparado para comunicar com os equipamentos industriais. Outra desvantagem é o nível de conhecimento necessário para o desenvolvimento de programas. Os primeiros sistemas de controle baseados em computadores necessitavam de interfaces especiais para manipular os sinais provenientes dos transdutores e atuadores. Estas interfaces normalmente tinham de ser desenvolvidas especialmente para cada planta industrial. Desde então, diversos fabricantes desenvolveram módulos de interfaces padrões para sinais digitais e analógicos.

1.5.1 Métodos de Programação

Todo programa de computador consiste em um número de instruções as quais dizem para o computador o que é para ser feito quando o programa é executado. Como os computadores entendem apenas informações binárias, as instruções dos computadores são muito diferentes da linguagem verbal utilizada por nós para descrever as ações que queremos que sejam executadas. Em programação, entretanto, vários artifícios são empregados para transformar nossa descrição verbal na linguagem própria dos computadores. Estes artifícios são implementados em linguagens de programação, as quais podem ser adquiridas a um custo bem acessível.

1.5.2 Código de Máquina e Assembler

Muitos computadores têm um limitado conjunto de instruções que desempenham operações simples tais como coletar um dado, armazenar o dado, somar números, etc. Através da combinação de um grande número destes códigos de máquina em extensos programas, o programador pode fazer com que o computador desempenhe funções complexas. Contudo, para que o programa possa funcionar, é muito importante que sejam seguidas as regras de como as instruções devem ser usadas e combinadas, normalmente chamado de sintaxe do programa.

Como os códigos de máquina são binários ou hexadecimais, a tarefa de programação é simplificada através do uso de instruções assembler. Cada uma destas instruções tem um nome composto por três letras (código mnemônico), tal como LDA, para coleta de dados e ADD para soma de dois números. Os programas assembler são escritos através de um editor e, antes que os mesmos sejam executados, os códigos mnemônicos devem primeiramente ser traduzidos para os códigos de máquina em hexadecimal. Esta tradução é feita por um outro programa chamado de assembler. Além da tradução do programa, o assembler também verifica a sintaxe dos programas e calcula os saltos lógicos dentro de um programa. A assemblagem, processo realizado pelo assembler, de um programa é feito para um determinado tipo de computador ou sistema operacional, sendo que existem programas capazes de fazer a assemblagem para outros sistemas operacionais.

Algumas desvantagens da programação em assembler consistem basicamente em:

- Faz-se necessário o conhecimento do funcionamento de computadores para facilitar a programação;
- O problema deve ser devidamente estruturado para facilitar a utilização do conjunto de instruções de um determinado computador;
- O programa final é destinado a um tipo de computador ou sistema operacional, não sendo facilmente portátil para outros.

Apesar destes pontos contrários, a linguagem assembler permite um ótimo desempenho e uma melhor utilização da memória do computador. Estes fatores podem ser determinantes em determinados sistemas de controle. O Assembler é chamado de linguagem de baixo nível, em função do mesmo ser próximo à forma como os computadores processam os dados.

Existem programas chamados de depuradores (*debuggers*), os quais permitem executar o programa passo-a-passo, de forma que possam ser simulador os dados do sistema real sem a necessidade de conexão com a planta industrial, facilitando a depuração de erros.

1.5.3 Compilação e Interpretação

A programação é facilitada significativamente quando a mesma é feita utilizando-se uma linguagem de alto-nível, a qual é traduzida para o código de máquina através de um programa chamado de interpretador ou compilador.

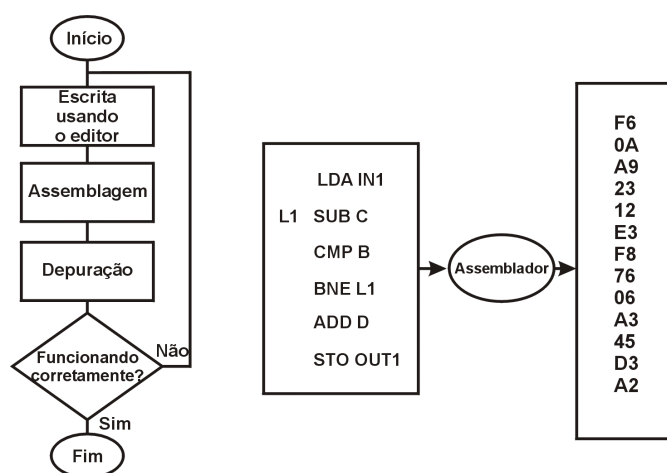


Figura 5 – Na programação de baixo nível, são usados os programas para edição, assemblagem e depuração no processo de geração do código de máquina.

A diferença entre compiladores e interpretadores é que o compilador primeiramente traduz todo o programa para o código de máquina antes do mesmo ser executado, enquanto o interpretador traduz o programa instrução por instrução durante a execução do programa. Isto significa que os programas compilados têm um processamento mais rápido. As linguagens de programação de alto nível mais comuns atualmente são o Pascal e a Linguagem C. Ambas são linguagens compiladas. Um exemplo de linguagem interpretada era o Basic mais antigo.

As instruções de uma linguagem de alto nível são semelhantes às funções matemáticas, sendo portanto mais intuitivas e fáceis de usar. Todas as linguagens de alto-nível são altamente padronizadas, onde as partes principais de um programa podem ser escritas independentemente o tipo de computador onde ela será executada. A compatibilização com do tipo de computador é feita pelo programa compilador ou interpretador durante a tradução para o código de máquina. Programas escritos em linguagens de alto nível são chamados de programa fonte, enquanto que o resultado da compilação é chamado de código objeto.

O programador não necessita conhecer detalhes técnicos do tipo de computador ou memória para a programação em uma linguagem de alto nível. Outra vantagem é que o programa gerado pode ser portado para outro tipo de computador, desde que exista o compilador adequado.

As desvantagens dos programas escritos em linguagens de alto nível são o maior consumo de memória e desempenho inferior em relação à linguagem assembler.

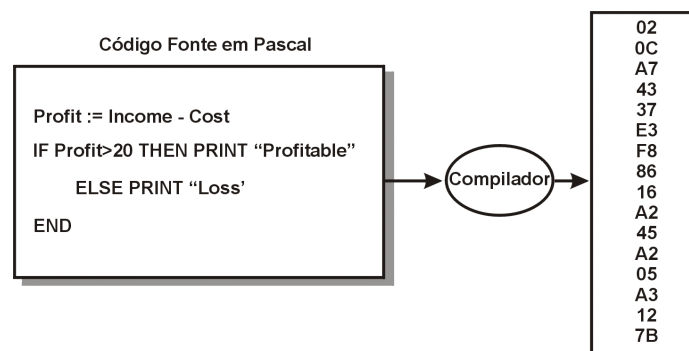


Figura 6 – Programas escritos em linguagens de alto nível são independentes do hardware e são traduzidos para o código de máquina pelo programa compilador.

1.6 Controladores Programáveis

Para atender à demanda da indústria, vários fabricantes desenvolveram no início dos anos 70 o Controlador Lógico Programável (CLP) ou *Programmable Logic Controller* (PLC), do inglês. O CLP é um computador adaptado para o ambiente industrial com uma linguagem de programação simplificada. Os primeiros CLPs foram desenvolvidos inicialmente para a substituição dos sistemas de controle baseados em relés. Desde então, os CLPs desenvolveram e são os controladores mais utilizados em sistemas de controle em todos os tipos plantas industriais, do controle de máquinas até o controle de uma linha completa de manufatura de grandes processos industriais.

Independentemente do fabricante e do tipo de CLP, a maioria dos CLPs possui três partes principais: CPU, memória e a unidade de Entradas e Saídas (E/S), todas comunicando através de um barramento de comunicação. A CPU coordena todas as tarefas do CLP e executa o programa de controle armazenado na memória. Os estados reais do processo são monitorados e amostrados pela unidade de E/S. Além das instruções lógicas, o CLP atual também possui uma grande capacidade aritmética. Portanto, muitos fabricantes estão adotando o termo Controlador Programável (CP) ao invés de CLP. A programação de CLPs é feita através de um computador externo, o qual é chamado de estação de engenharia. O programa compilado é carregado na CPU e depois armazenado na memória utilizando-se uma porta serial ou uma rede local (LAN). A maioria dos CLPs permitem a monitoração dos estados do processo no modo *on-line* utilizando-se a estação de engenharia, enquanto o programa está sendo executado.

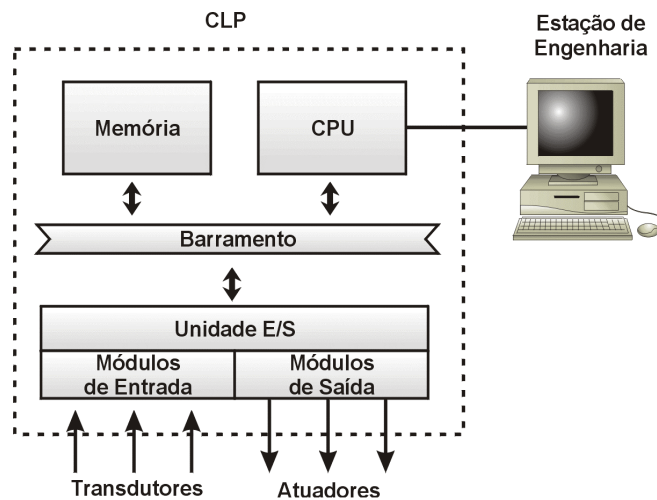


Figura 7 – Componentes de um controlador programável

1.6.1 Unidades de Entradas e Saídas

Uma característica importante do CP é que este é projetado para trabalhar no ambiente industrial. Muitos controladores têm uma unidade de E/S modularizada, para conexão direta com os sinais dos transdutores e atuadores.

O propósito da unidade de E/S é de converter os sinais de processo para os baixos níveis utilizados pelo controlador, além de filtrar os transientes elétricos provenientes dos equipamentos de processo. Anormalmente, isto é feito através do uso de isoladores óticos, os quais utilizam foto-diodo e foto-transistor encapsulados em um acoplador ótico.

Uma vez que existem diferentes níveis de sinais em um processo industrial, muitas unidades de E/S permitem intercambiar diferentes módulos de E/S. Assim, uma unidade de E/S poder ser ajustada às características dos níveis de sinais específicos da planta industrial. Os módulos de E/S mais utilizados são os de entradas e saídas digitais, com níveis de sinais que vão desde 24 V e 48V em corrente contínua, até 110 V e 220 V em correntes alternadas.

Cada vez mais, os CPs possuem a funcionalidade de processamento aritmético. Tais sistemas permitem o uso de módulos de E/S analógicos. Muitos transdutores representam a grandeza física através de um sinal de saída de 4 a 20 mA, sendo utilizado 4 mA como valor mínimo para permitir a detecção de cabo partido.

Atualmente a maioria dos CPs dispõem de módulos especiais para funcionalidades específicas. Dentre os módulos disponíveis podemos citar os módulos de contagem rápida, controladores de movimento, interfaces de comunicação, co-processadores, etc.

1.6.2 Métodos de Programação

Os primeiros CPs utilizavam uma linguagem de programação baseada no Diagrama Ladder, para a lógica de relés. O programa era editado utilizando-se um terminal de programação, o qual tinha um teclado que teclas mostravam símbolos de contato (normalmente aberto / normalmente fechado), bobinas e ramificações, com os quais o electricista de manutenção era familiar.

O terminal de programação compilava o Diagrama Ladder para o código de máquina, o qual era carregado no CP para execução. Durante a execução do programa, o terminal de programação permitia a visualização do programa de controle, animando os contatos e bobinas energizados para facilitar a análise do programa e depuração de erros.

A programação utilizando o Diagrama Ladder é muito intuitiva, especialmente para os profissionais com conhecimento prévio de sistemas de controle baseados em relés. Portanto, este método é o preferido pelos fabricantes de CPs norte-americanos.

Em plantas muito grandes ou quando os profissionais sem um conhecimento prévio de lógica a relés têm que desenvolver o programa de controle, a Lista de Instruções é mais utilizada, a qual faz uso de lógica booleana. Muitos fabricantes de CPs europeus adotaram a Lista de Instruções como linguagem padrão para os seus produtos.

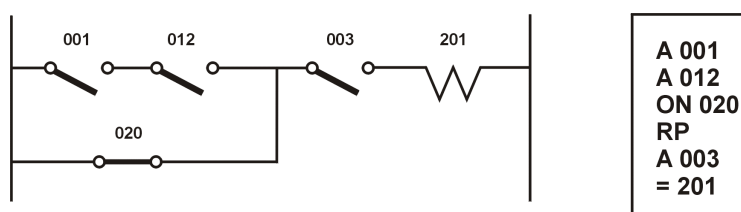


Figura 8 – Exemplo de programa utilizando Diagrama Ladder e Lista de Instruções.

1.6.3 Ferramentas de Programação baseadas em Computadores

Os primeiros CPs eram programados através de terminais dedicados para este propósito e para os sistemas de um fabricante específico. Atualmente todos os CPs são programados através de computadores pessoais de mercado (PCs), que executam um *software* ou ferramenta de programação. Um sistema completo com o PC e o *software* de programação é chamado de estação de engenharia.

Muitos softwares de programação de CPs contêm diversas aplicações integradas, as quais simplificam o desenvolvimento de programas para o sistema de controle.

O editor é utilizado para definição das variáveis e para escrita das instruções do programa de controle. Muitos editores permitem a verificação de sintaxe e ajudam o programador a evitar erros. A edição de programas é feita normalmente no modo *off-line*, o que significa que a estação de engenharia está fora de linha, ou seja, sem comunicação com o controlador.

O compilador traduz o programa de controle para o código de máquina e faz o carregamento deste código para execução no CP.

Muitos *softwares* de programação possuem uma funcionalidade muito útil, a qual compila e simula a execução do programa de controle sem ser necessário o carregamento deste no controlador. Os estados simulados das entradas e saídas são visualizados no próprio *software* de programação. A simulação possibilita o teste do programa de controle através da alteração dos sinais de entrada e visualização do processamento da lógica e atuação das saídas do programa.

Alguns *softwares* de programação podem ser utilizados no modo on-line, para monitoração dos estados do processo na tela do computador, enquanto o programa de controle é executado pelo CP.

Com o crescente aumento do desempenho das estações de engenharia baseadas em PCs e dos próprios controladores, muitos fabricantes oferecem *softwares* de programação onde, além do Diagrama Ladder e da Lista de Instruções, é também possível a programação utilizando-se os métodos de Texto Estruturado, Gráfico Sequencial de Funções e Blocos Funcionais.

1.6.4 Execução Cíclica

Sistemas de controle industrial são sistemas de tempo real, o que significa que alterações nos sinais de entradas exigem uma imediata ação no sinal de saída correspondente. Ou seja, o tempo de reação do sistema de controle deve ser condizente com as necessidades do processo controlado. Um exemplo pode ser uma máquina onde algum movimento deve ser interrompido quando um limite de posição é excedido. Se o controlador não reagir no tempo necessário, o resultado poderá ser a quebra da máquina ou o ferimento do operador. As conseqüências de uma reação atrasada são inaceitáveis.

Para garantir o atendimento às exigências de um sistema de tempo real, o programa de controle deve monitorar constantemente os sinais de entrada provenientes do processo. Para possibilitar isso, o programa compilado é executado ciclicamente numa frequência específica, ou varredura (*Scan*). Alterações nos sinais de entrada somente afetarão os sinais de saída no fim de cada ciclo completo de programa. O tempo de ciclo necessário para o programa é determinado pelo atraso máximo permitido para o processo controlado.

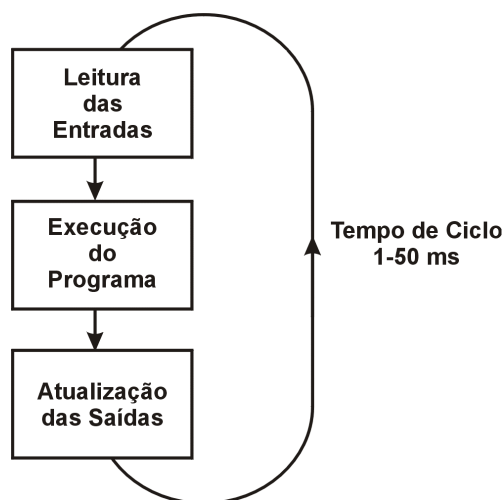


Figura 9 – Varredura típica de um Controlador Programável

Como diferentes sub-processos podem exigir diferentes tempos de reação, alguns CPs permitem que o programa de controle seja dividido em diferentes tarefas, cada qual com seu próprio tempo de ciclo. Estes CPs são chamados de multi-tarefas.

1.6.5 Sistemas Distribuídos

Em grandes plantas industriais é necessária a distribuição da função de controle para vários CPs e computadores de processo. Esta estratégia melhora o desempenho e reduz o risco para processos de manufatura.

A interligação entre transdutores, atuadores e CPs representa o maior custo para um sistema de controle. Se a planta é distribuída em uma área muito grande, é possível reduzir os custos através da utilização de sub-sistemas de E/S remotas, localizadas próximas aos sub-processos.

Sistemas de controle distribuídos utilizam um protocolo padronizado para comunicação de dados. Muitos fabricantes desenvolveram seus protocolos proprietários durante os anos 90, dentre os quais podemos citar o COMLI, Modbus, *Data Highway Plus* e 3964R. Atualmente, existem protocolos padrões suportados pela maioria dos fabricantes de CPs, onde podemos citar o Profibus, ControlNet, Foundation Fieldbus, WorldFip, etc.

1.6.6 Soft PLC

Um problema relacionado aos CPs, é que os fabricantes utilizam *hardware* proprietário para o controlador, com uma linguagem de programação também proprietária. Apesar das funções básicas serem praticamente idênticas, as instruções possuem nomes diferentes e existem regras diferentes para a sintaxe dos programas. Isto torna impossível a comunicação e intercâmbio de programas entre sistemas de fabricantes diferentes.

Vários fornecedores de *software* apresentaram um novo tipo de controlador chamado *Soft PLC*. O *Soft PLC* é um *software* de tempo real que executa o programa de controle em um PC padrão, comunicando com o processo através de uma unidade de E/S modular.

A maior vantagem do *Soft PLC* é que todo o *hardware* necessário é independente do fabricante. Infelizmente, os fornecedores ainda não estabeleceram o *Soft PLC* como um padrão industrial. Isto significa que os programas de controle desenvolvidos com um *Soft PLC* não podem ser transferidos para o *Soft PLC* de outro fornecedor.

2. MOTIVAÇÃO PARA OS SISTEMAS ABERTOS

2.1 Diferentes Dialeto de Programação

O CP é um dos componentes mais críticos da indústria atual. Com a utilização dos sistemas de controle na maioria das indústrias, incluindo aplicações que exigem segurança, é muito importante que os programas possam ser facilmente entendidos por uma grande parte dos profissionais do chão de fábrica. Além do programador, o programa de controle deve ser fácil para todos os técnicos, gerentes de fábrica e engenheiros de processo.

Por quase duas décadas o mercado tem sido dominado por uma meia dúzia de fabricantes que oferecem soluções muito parecidas, porém com particularidades nos dialetos de programação. Muitos usuários de CPs têm decidido eleger no mínimo dois fornecedores, com o objetivo principal de minimizar o risco. Nas aplicações reais, isto implica em um maior custo devido ao retrabalho e problemas de comunicação entre produtos de diferentes fabricantes.

2.2 Qualidade de *Software*

Quanto mais tarefas da indústria de manufatura e de processos são automatizadas, os programas se tornam cada vez maiores e mais complexos, dificultando o gerenciamento dos mesmos. Em muitos casos, são necessários mais de um programador para o desenvolvimento do programa para automação industrial. A experiência mostra que o risco devido a erros de programação cresce exponencialmente com o número de programadores envolvidos e, conseqüentemente, com o tamanho do programa.

A experiência também mostra que as novas plantas industriais apresentam problemas por um longo período após a instalação. Algumas falhas podem interromper a produção e, no pior caso, também podem causar danos aos equipamentos e produtos.

É bem sabido que *software* de boa qualidade representa um custo maior. Normalmente, o *software* de controle é desenvolvido pela própria indústria ou por empresas integradoras de sistemas dedicadas ao ambiente industrial. Em ambos casos, a produção de *software* e o seu custo não competem no mercado aberto. Conseqüentemente, os fornecedores de *software* não são motivados a buscar ferramentas e métodos de desenvolvimento mais eficientes.

A grande maioria dos programas de controle é escrita utilizando pacotes de *software* proprietários dos fabricantes de produtos de controle. Muitos destes pacotes não dispõem de recursos adequados para trabalhar com módulos, para reutilização de código e para documentação. Portanto, a qualidade do *software* é dependente da capacidade intelectual dos programador.

Antes da norma IEC 61131-3, uma boa engenharia de *software* era a meta principal das aplicações de controle.

2.3 Custo de Software

Na última década, a padronização dos pacotes de *software* para os computadores pessoais, tais como processadores de texto e planilhas eletrônicas, tornou-se muito popular, vendendo milhões de cópias. Isto proporcionou aos fornecedores a redução drástica dos preços de produtos.

A distribuição pela Internet ampliou ainda estes limites e hoje são muitos as aplicações padrões disponíveis como licenças *shareware*, quase sem custo.

De forma contrária, os *softwares* para aplicações de controle são adaptadas para características específicas de uma única planta industrial. Isto significa que o custo total do desenvolvimento recorre sobre um único usuário.

Muitos usuários consideram difícil absorver o custo total do desenvolvimento de *software*. Um usuário sem experiência no desenvolvimento de *software* consegue apresentar apenas uma pobre descrição funcional para o desenvolvedor. Em muitos casos, isto leva ao desenvolvimento de um *software* que atende parcialmente aos requisitos do usuário. Mesmo pequenas modificações e melhorias implicam em altos custos para implementação, especialmente na fase

final do desenvolvimento.

O crescente aumento da velocidade de desenvolvimento do *hardware* têm levado a uma constante queda dos preços. Atualmente, os computadores pessoais têm desempenho comparável aos antigos *mainframes*. Com a crescente melhoria da relação de custo-desempenho do *hardware*, o custo total da instalação é mais e mais determinado pelo tempo para desenvolvimento do *software* de controle.

Em muitos projetos, o maior peso está na padronização e reutilização dos programas do que na busca de *hardware* ótimo.

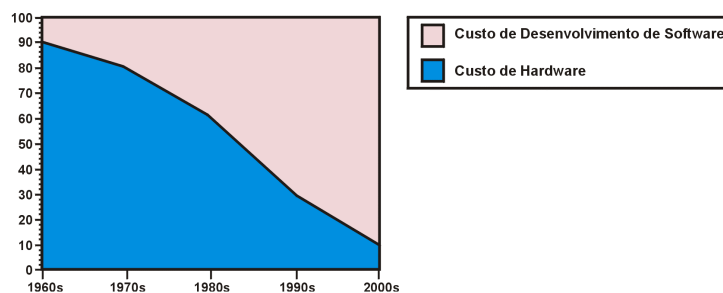


Figura 10 – Custo do Hardware versus Software

A automação de plantas e máquinas pode se tornar perigosa para o operador ou para os produtos se o *software* de controle apresenta erros críticos. Portanto, o *software* depende de um procedimento rigoroso para teste e validação. Nas aplicações reais, o teste pode ser bastante demorado, especialmente se o trabalho tem que ser feito com o processo em funcionamento. Se o *software* de controle é desenvolvido por programadores menos experientes, o custo do teste pode ser maior que o custo da programação.

2.4 Portabilidade de Aplicações

O computador pessoal juntamente com o sistema operacional *Windows* é um padrão de fato para as aplicações de escritório em todo o mundo. A principal razão para esta grande difusão do uso dos PCs é a compatibilidade de *software*. Os programas aplicativos desenvolvidos para o *Windows* podem ser usados em quase todos os PCs espalhados pelo mundo.

Mais de 25 anos após a introdução dos controladores programáveis, este mercado ainda carece de uma padronização internacional similar ao que ocorreu com os PCs. Muitos fornecedores usam um dialeto de programação proprietário que funciona apenas com o seu *hardware* proprietário.

Do lado oposto, a quase totalidade das indústrias usuárias de controladores programáveis têm uma alta demanda pela portabilidade do *software* do sistema de controle. Desde que o custo de desenvolvimento de um *software* bem testado é muito maior que o custo do *hardware*, é cada vez maior a necessidade de se portar as aplicações existentes de uma plataforma de *hardware* antiga para novos sistemas.

Muitos observadores consideram um mistério o fato de serem necessários mais de 25 anos para o mercado de controladores programáveis promover o estabelecimento de uma padronização da programação de controladores proposta pela norma IEC 61131-3.

2.5 Reutilização de Software

Até pouco tempo, os programadores mediam a sua eficiência através do número de linhas de código geradas por dia. Normalmente, os programadores não gostam de “gastar” o seu tempo na estruturação e detalhamento da especificação do *software*. Eles preferem partir diretamente para a geração de linhas de código a partir de uma especificação pobre, normalmente feita pelo usuário final, utilizando a sua linguagem de programação favorita, normalmente o Diagrama Ladder ou a Lista de Instruções.

Atualmente, mesmo os programadores reconhecem que o ponto de partida de qualquer projeto quando uma determinada função é analisada, a estruturação e o planejamento são as chaves para o desenvolvimento de um programa eficiente e de baixo custo.

O método tradicional para redução de custos de *software* é a reutilização de partes comuns do código do programa em várias aplicações similares. Infelizmente, isto é uma tarefa difícil na automação industrial devido ao fato da maioria dos processos terem comportamentos diferentes.

Outro obstáculo para a reutilização de *software* é que o código gerado é fortemente afetado pelo estilo ou filosofia utilizada por cada programador. Quando a aplicação final é o resultado de um trabalho em equipe, são perceptíveis as diferenças e perceptível as diferenças dos códigos gerados por diferentes programadores. A única forma para reduzir este problema é encorajar (entenda obrigar) todos os membros da equipe a seguir regras e formalismos para a produção de código.

2.6 **Comunicação com outros Sistemas**

O primeiro controlador programável apresentado nos anos 70 não possuía meios para interagir com o operador ou para comunicação com outros controladores.

Nas plantas industriais atuais, é de grande importância que o operador tenha interação com o sistema. Os grandes centros de controle estão sendo substituídos pelos sistemas SCADA e SDCD (**S**istema **D**igital de **C**ontrol e **D**istribuído) baseados em PCs, utilizando grandes telas coloridas para visualização do processo e do seu estado real.

Em grandes plantas industriais, as funções de controle são distribuídas para diferentes CPs comunicando rede com cada outro, através de um protocolo padronizado.

Todas as plantas industriais possuem um Sistema de Gerenciamento de Informações (SGI) baseado em PCs para tratamento de informações de caráter estatístico e econômico. É cada vez mais importante a integração dos sistemas SGI com os sistemas SCADA ou SDCD, resultando em um sistema de controle e gerenciamento total.