



JEAN LUCAS GIANOTI GABAN

**CONCEITOS FUNDAMENTAIS SOBRE CLPs -
PROPOSTA PARA O SEGUNDO CAPÍTULO DO PROJETO
"INTRODUÇÃO À INFORMÁTICA INDUSTRIAL EM 15
SEMANAS: TEORIA E PRÁTICA"**

LAVRAS – MG

2022

JEAN LUCAS GIANOTI GABAN

**CONCEITOS FUNDAMENTAIS SOBRE CLPs -
PROPOSTA PARA O SEGUNDO CAPÍTULO DO PROJETO "INTRODUÇÃO À
INFORMÁTICA INDUSTRIAL EM 15 SEMANAS: TEORIA E PRÁTICA"**

Trabalho de conclusão de curso apresentado à
Universidade Federal de Lavras, como parte
das exigências do Curso de Engenharia de
Controle e Automação, para obtenção do título
de Bacharel.

Prof. Dr. Dimitri Campos Viana
Orientador

LAVRAS – MG

2022

JEAN LUCAS GIANOTI GABAN

**CONCEITOS FUNDAMENTAIS SOBRE CLPs - PROPOSTA PARA O SEGUNDO
CAPÍTULO DO PROJETO "INTRODUÇÃO À INFORMÁTICA INDUSTRIAL EM 15
SEMANAS: TEORIA E PRÁTICA"**

**FUNDAMENTAL CONCEPTS ABOUT PLCs - PROPOSAL FOR THE SECOND
CHAPTER OF THE PROJECT "INTRODUCTION TO INDUSTRIAL INFORMATICS IN
15 WEEKS: THEORY AND PRACTICE"**

Trabalho de conclusão de curso apresentado à
Universidade Federal de Lavras, como parte
das exigências do Curso de Engenharia de
Controle e Automação, para obtenção do título
de Bacharel.

APROVADO em 02 de Maio de 2022.

Dr. Dimitri Campos Viana (UFLA)

Dr. Danilo Alves de Lima (UFLA)

Bacharel Alysson Alves Fernandes (UFLA)

Prof. Dr. Dimitri Campos Viana
Orientador

**LAVRAS – MG
2022**

RESUMO

Este trabalho consiste na elaboração de um texto dissertativo, abordando os conceitos fundamentais dos Controladores Lógico-Programáveis (CLPs), que são amplamente utilizados na área da engenharia conhecida como "Informática Industrial". A proposta é que este texto faça parte de um projeto maior, concretizando-se como o segundo capítulo do livro "Introdução à Informática Industrial em 15 Semanas: teoria e prática". Este livro, por sua vez, tem o objetivo de atender, como referência bibliográfica, aos diversos cursos de graduação que incorporam essa área do conhecimento em sua grade curricular, suprimindo a necessidade de um material didático que contemple tanto aspectos teóricos quanto práticos. Assim sendo, o capítulo do livro, aqui apresentado, foi estruturado de forma que uma seção contenha um levantamento sobre os principais aspectos relacionados ao hardware destes controladores e outra seção seja dedicada à organização de sua memória e declaração de variáveis. Além disso, foram apresentados exercícios teóricos e um guia para aulas práticas, uma vez que estas são fundamentais, como forma de fixar o conteúdo, permitindo ao estudante aplicar o conhecimento adquirido nas aulas teóricas. Estima-se que o resultado deste trabalho tenha sido a produção de um texto de fácil leitura, transmitindo as informações necessárias de maneira clara, proporcionando ao leitor uma leitura rápida e confortável, cumprindo com seu objetivo de ser utilizado como material de suporte ao ensino.

Palavras-chave: Informática industrial. Controladores Lógico-Programáveis. Hardware. Memória.

ABSTRACT

The paper consists of the elaboration of a dissertation text, addressing the fundamental concepts of Programmable Logic Controllers (PLCs), which are widely used in the engineering area known as "Industrial Informatics". The proposal is that this text will be part of a larger project, as the second chapter of the book "Introduction to Industrial Informatics in 15 Weeks: theory and practice". In fact, this book aims to be used as a bibliographical source for various undergraduate courses that incorporate this area of knowledge in its academic curriculum, satisfying the demand for a didactic material that includes both theoretical and practical aspects. Therefore, the book chapter was structured with one section containing a survey on the main aspects related to the hardware of these controllers, and the other one section dedicated to memory organization and declaration of variables. Besides that, theoretical exercises and a guide for practical classes were presented, since they are very important to consolidate knowledge, allowing students to apply the concepts learned in the classroom. It is estimated that the result of this work has been the production of an easy-to-read text, transmitting the necessary information in a clear way, providing both fast and comfortable reading, fulfilling its mission to serve as a teaching support material.

Keywords: Industrial informatics. Programmable logic controllers. Hardware. Memory.

SUMÁRIO

PRIMEIRA PARTE.	5
1 INTRODUÇÃO	5
2 CONSIDERAÇÕES FINAIS	6
SEGUNDA PARTE	7
CAPÍTULO DE LIVRO - CONCEITOS FUNDAMENTAIS SOBRE CLPs.	8

PRIMEIRA PARTE

1 INTRODUÇÃO

Os Controladores Lógico-Programáveis (CLPs) são equipamentos comumente encontrados no meio industrial, assim faz-se necessário o ensino do uso desses durante a graduação em Engenharia de Controle e Automação. Considerando que, para contemplar toda a ementa proposta para a disciplina de Informática Industrial, da forma como ela é atualmente ministrada para o mencionado curso na Universidade Federal de Lavras (UFLA), é necessária a consulta a várias referências, foi proposto o desenvolvimento de um material didático voltado para essa demanda específica.

Os livros didáticos são fundamentais como material de referência e aprendizado e, além disso, é válido ressaltar a importância das aulas práticas na formação do pensamento crítico do estudante, por ser uma forma de fixar o conteúdo apresentado em sala de aula, permitindo ao estudante aplicar o conhecimento adquirido nas aulas teóricas. Realizando aulas práticas paralelamente às aulas teóricas, é possível perceber as relações entre os aspectos teóricos e os experimentos realizados. Assim sendo, viu-se a necessidade de se incorporar ao material produzido um roteiro para práticas em laboratório que fosse diretamente relacionado ao assunto abordado em sala de aula.

Diante do exposto, o projeto "Introdução à Informática Industrial em 15 Semanas: Teoria e Prática" visa suprir a necessidade dos alunos no que se refere ao material didático voltado às aulas teóricas e práticas desenvolvidas durante a disciplina de Informática Industrial. A proposta é que essa obra, ao final, seja composta por 15 capítulos estruturados de forma padronizada sendo que cada um deles abordará temas que serão objeto de estudo durante uma semana do respectivo período letivo. Cada capítulo será composto por sete seções sendo elas: 1) Introdução; 2) Conteúdo teórico referente a primeira aula teórica; 3) Conteúdo teórico referente a segunda aula teórica; 4) Roteiro para a aula prática; 5) Tarefas para a próxima semana; 6) Exercícios e 7) Referências.

Dessa forma, o texto em anexo é a proposta para o segundo capítulo do mencionado projeto, sendo intitulado "Conceitos Fundamentais sobre CLPs". O texto desenvolvido apresenta aos discentes um panorama sobre o hardware desses controladores, bem como uma descrição da forma com a qual sua memória é organizada, incluindo a declaração de variáveis em um projeto.

Estes aspectos são apresentados como uma forma de preparação para ensino das linguagens de programação dos CLPs, previstas para serem abordadas nos próximos capítulos do mencionado projeto.

2 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo apresentar conceitos fundamentais dos CLPs, abordando os principais aspectos de hardware desses controladores, bem como a forma com a qual a memória dos mesmos é organizada. Considerando a necessidade de se abordar tais temas, antes que as linguagens de programação propriamente ditas sejam apresentadas, indica-se que o conjunto deste trabalho com os demais capítulos em desenvolvimento busque suprir a necessidade dos alunos no que se refere a ter um material completo como referência e alternativa para o estudo.

Um dos desafios encontrados durante a realização deste trabalho foi como confeccionar um material que fosse ao mesmo tempo atrativo para aplicação em sala de aula como forma de suporte ao ensino e, ainda assim, tratasse os assuntos com o aprofundamento necessário, porém sem ser um texto enfadonho. Para isso, foi utilizada uma linguagem objetiva, apresentando os fatos de maneira clara e concisa, buscando captar o interesse dos estudantes, que são o principal público alvo desse livro.

Outro desafio encontrado foi o de produzir um capítulo do livro de maneira independente dos autores dos capítulos seguintes. Tal fato trouxe a dificuldade de fazer um texto que se junte aos outros capítulos de maneira concisa. Para garantir que o material final apresente uma unidade (coesão) entre os capítulos elaborados pelos outros autores, estabeleceu-se um padrão de escrita e imagens a ser seguido.

A utilização de recursos visuais como imagens e gráficos foi uma maneira de tornar o material mais interessante aos olhos dos leitores, procurando-se trazer informações sobre os diversos tópicos abordados em conjunto com um texto fluído, com frases curtas e objetivas. Dessa forma, buscou-se proporcionar ao leitor a possibilidade de uma leitura dinâmica e confortável.

SEGUNDA PARTE

Semana 2

CONCEITOS FUNDAMENTAIS SOBRE CLPs

Jean Lucas Gianoti Gaban e Dimitri Campos Viana

2.1 INTRODUÇÃO

Para se fazer bom uso de um CLP, antes de programá-lo, é necessária uma boa compreensão de suas características, tanto as específicas (quantidade de memória disponível, capacidade de processamento, limites para a expansão de entradas e saídas etc.) como as gerais (modularidade de *hardware*, segmentação da memória, ciclo de funcionamento etc.). Enquanto as primeiras precisam ser verificadas nos manuais de cada equipamento, as últimas devem ser objeto de estudo dos profissionais envolvidos com essa tecnologia e serão abordadas ao longo deste capítulo.

Conforme mencionado no capítulo anterior, se alguém lhe perguntar o que é um CLP, você pode começar a explicação por meio de uma comparação entre esses dispositivos e os PCs convencionais, dizendo que os membros dos dois grupos são microprocessados, possuem memórias retentivas e voláteis e, certamente, capacidade de trocar dados via rede. Em contraponto, as diferenças começam na robustez do *hardware*: os CLPs são fabricados para operar continuamente, durante anos, em ambientes não climatizados e na presença de outras adversidades, como partículas em suspensão, vibrações moderadas e ruídos eletromagnéticos; já os PCs, para que possam ser utilizados em um sistema de automação industrial, precisam ser instalados em ambientes nos quais esses problemas sejam controlados.

Outra diferença fundamental entre os dois tipos de equipamento está relacionada às formas com as quais eles interagem com outros elementos do mundo real: enquanto os PCs possuem periféricos como teclado, *mouse* e monitor, para interface com os seres humanos; os CLPs possuem terminais elétricos de entrada e de saída para, entre outras coisas, interligação de instrumentos de medição e atuadores, sendo que seus respectivos sinais passam a ser monitorados e controlados por meio de valores em formato digital, armazenados em determinadas posições de memória do CLP.

Por exemplo, no sistema de controle *on/off* apresentado no capítulo anterior, o circuito lógico formado por relés eletromecânicos (Figura 2.1A) pode ser substituído por um CLP devidamente programado (Figura 2.1B), sem que haja necessidade de se fazer alterações no circuito de potência (Figura 2.2). Nesse caso, é interessante notar que as chaves de nível e o relé utilizado para acionar este último circuito precisam ser mantidos no sistema, pois eles são os meios pelos quais o CLP recebe informações e interfere no mundo real. No entanto, os dois outros relés usados no primeiro circuito foram suprimidos no sistema com CLP, uma vez que os papéis desempenhados por eles podem ser substituídos por recursos de programação.

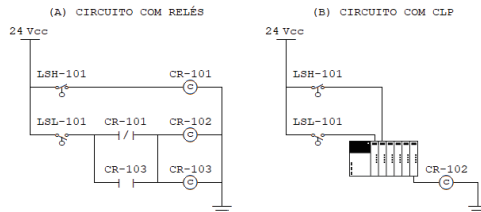


Figura 2.1: Possíveis diagramas elétricos para o sistema de controle *on/off* apresentado como exemplo prático no capítulo anterior: o primeiro, com base em relés eletromecânicos (A); o segundo, com base em um CLP (B).

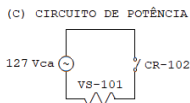


Figura 2.2: Circuito de potência que pode atender a qualquer um dos dois circuitos lógicos representados na figura anterior.

Dessa forma, ao se transpor e extrapolar as vantagens decorrentes do uso do CLP nesse exemplo para os complexos sistemas de automação que controlam processos industriais de grande porte, onde um controlador do tamanho de uma caixa de sapatos pode conter um programa que substitui centenas ou mesmo milhares de relés eletromecânicos e um enorme volume de cabos elétricos, dois benefícios são imediatamente percebidos: **significativa economia de material elétrico** e **redução do espaço físico ocupado pelo sistema de automação**.

Para os sistemas de automação que se baseiam em CLPs, as tarefas de elaborar e manter atualizados os projetos elétricos que especificam, entre outras coisas, as interligações entre elementos periféricos e suas entradas e saídas (como na Figura 2.1B) são uma necessidade. No entanto, toda a parte lógica do sistema é implementada por meio de um ou mais programas, que podem ser consultados ou mesmo impressos, sempre que necessário. Além disso, as diversas instruções que compõem esses programas podem ser incluídas, alteradas e excluídas de forma muito mais simples do que as correspondentes tarefas em um painel composto por elementos elétricos. Inclusive, em um ambiente de programação, os códigos podem ser desenvolvidos e testados antes de serem descarregados no CLP. Tais fatos agregam pelo menos mais duas vantagens a esse tipo de sistema: **documentação simplificada** e **redução do tempo despendido em desenvolvimentos e alterações**.

Existem ainda várias outras vantagens decorrentes da substituição de circuitos lógicos baseados em relés por CLPs, entre elas aquelas proporcionadas por algumas linguagens de programação, que por esse motivo, serão discutidas no próximo capítulo. No entanto, neste momento, ainda há de se destacar uma qualidade muito importante desses dispositivos: **a versatilidade**, na medida em que um mesmo modelo pode ser utilizado em pontos distintos de um processo produtivo ou mesmo em diferentes segmentos industriais, como extrativista, siderúrgico, petroquímico, alimentício, em estações de tratamento de água e esgoto ou na geração, transmissão e distribuição de energia.

2.2 PRINCIPAIS CARACTERÍSTICAS DE HARDWARE

CARACTERÍSTICAS BÁSICAS

Como mencionado no início do capítulo, os CLPs são fabricados para trabalhar majoritariamente em uma planta industrial, que além de ser um ambiente agressivo para componentes eletrônicos, pode sofrer algumas modificações ao longo do tempo. Dessa forma, é necessário que o CLP consiga lidar com essas adversidades facilitando reparos e permitindo que novos dispositivos sejam incorporados. Devido à sua característica de modularidade de *hardware*, isso é possível.

A modularidade de *hardware* é uma característica importante dos CLPs, isso significa que os seus principais componentes podem ser disponibilizados em forma de módulos independentes. Entre esses, podemos citar:

Rack principal: trata-se de um trilho que serve para a sustentação mecânica e agrupamento dos módulos, permitindo que eles se conectem eletricamente.

Fonte: módulo responsável por proteger e fornecer a corrente elétrica em níveis de tensão adequados para a alimentação dos demais módulos.

CPU (do termo em inglês *Central Processing Unit*): é o componente responsável pelo gerenciamento do funcionamento do CLP. Esse componente é formado pelo processador e pela memória que interagem entre si continuamente para armazenar e executar o programa do usuário.

Cartões de entrada e saída: também conhecidos como módulos de entrada e saída, ou módulos *Input/Output* ou simplesmente como módulos de I/O, são responsáveis por fazer a interface da CPU com o mundo exterior. Esses cartões possuem terminais elétricos que permitem a conexão de dispositivos de entrada e saída, que são responsáveis por interagir com o ambiente. Exemplos desse tipo de dispositivo são os sensores, atuadores e elementos de interface com os operadores da planta, que podem ser tanto digitais quanto analógicos, como botões, sirenes, sensores de temperatura, válvulas proporcionais e outros.

Processadores de comunicação: são responsáveis pela comunicação do CLP com a plataforma de desenvolvimento, sistemas supervisórios, outras CPUs, sensores e atuadores inteligentes e etc. Esses módulos são usados para aumentar a capacidade de comunicação e permitir, dessa forma, que a CPU se comunique usando diferentes protocolos.

Todos esses tipos de módulos são essenciais e largamente utilizados e, por isso, serão abordados com maior nível de detalhes ao longo do capítulo. No entanto, é possível encontrar outros módulos disponíveis como: módulos de contagem rápida; módulo para controle de posicionamento de eixos; módulo para comunicação que usam protocolos específicos como SMS e GPRS; módulos para monitoramento de vibrações em componentes mecânicos, entre outros. Esses, que são menos comuns, não serão discutidos nesta obra.

A característica de modularidade concede aos CLPs uma grande flexibilidade. Essa vantagem pode ser observada quando uma expansão da capacidade do CLP é requerida, seja

pelo aumento do número de cartões de entrada e saída ou pela inclusão de outros módulos, como o de contagem rápida, por exemplo. Além disso, caso qualquer um dos módulos apresente algum tipo de problema, basta substituí-lo pontualmente. Portanto, devido às características mencionadas, os CLPs modulares (Figura 2.3) são os mais utilizados.

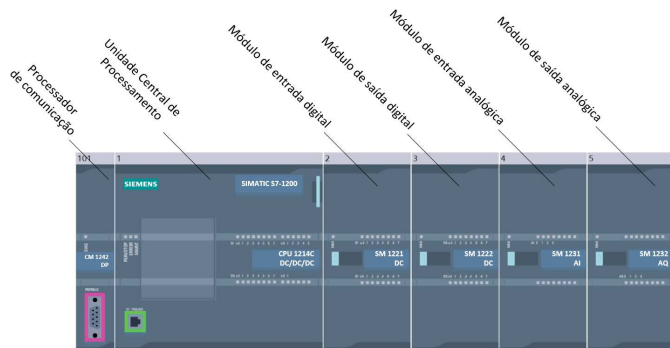


Figura 2.3: Representação de um CLP modular. Cada módulo é encaixado em uma região específica, chamada slot. Apesar da fonte não aparecer na imagem, geralmente ela fica no slot mais à esquerda.

Existem ainda CLPs cuja CPU é disponibilizada com recursos incorporados, como processadores de comunicação e pontos de conexão para entradas e saídas. Alguns CLPs desse tipo permitem expansão e são chamados de compactos, outros não permitem expansão e são chamados de micro CLPs (que, além dessa desvantagem, geralmente possuem uma menor capacidade de processamento).

Devido a característica de modularidade de hardware, existem diversas maneiras de se agrupar os módulos em um CLP. Para que a CPU reconheça esses módulos, é necessário que o usuário declare a sua utilização. Essa declaração geralmente ocorre por meio de uma ferramenta computacional, que pode ser instalada em um computador pessoal comum, chamada de plataforma de desenvolvimento.

Além da configuração de *hardware*, a plataforma de desenvolvimento também é utilizada para fazer a implementação do programa do usuário e, posteriormente, sua transferência para a memória do CLP, por meio de um procedimento chamado de *download*. Portanto, verifica-se que essa ferramenta é um componente essencial para quem trabalha com CLPs, incorporando também outras funções importantes. Dessa forma, pode-se resumir a importância de uma plataforma de desenvolvimento por meio dos seguintes tópicos:

Configuração de *hardware*: onde é feita a declaração do *hardware* que vai ser utilizado, como o modelo da CPU, o processador de comunicação e os cartões de entradas e saídas.

Desenvolvimento de programas: por meio da plataforma de desenvolvimento, o usuário consegue implementar o programa que vai ser executado. Geralmente, essa implementação pode ser feita tanto em linguagens gráficas, como Diagrama Ladder

(LD), Diagrama de Blocos Função (FBD) e Diagrama de Fluxo Sequencial (SFC), quanto em linguagens textuais, como Lista de Instruções (IL) e Texto Estruturado (ST), dependendo do modelo do CLP.

Procedimentos de *download* e *upload*: esses procedimentos consistem na transferência do programa do usuário entre a plataforma de desenvolvimento e a memória do CLP e vice-versa, respectivamente, como na Figura 2.4.



Figura 2.4: CLP conectado a um computador com a plataforma de desenvolvimento instalada. É possível realizar os procedimentos de *download* e *upload*, nesse caso, por meio da conexão feita com um cabo.

Diagnóstico de *hardware*: consiste na verificação das condições do hardware em tempo real. Isso é possível porque pode-se estabelecer uma comunicação constante (*online*) entre a plataforma de desenvolvimento e o CLP, permitindo identificar falhas, como: falha na fonte de alimentação que, porventura, não esteja fornecendo os níveis de tensão adequados; falhas nos cartões de I/O, ocasionadas por uma conexão feita de forma equivocada. Caso seja identificada uma falha desse tipo, o CLP não entra no modo de execução, ou a interrompe imediatamente.

Monitoramento de posições de memória: permite monitorar em tempo real os valores armazenados nas posições de memória. Esse é um recurso muito útil para quando se deseja verificar valores de variáveis associadas às entradas e saídas, e outros valores da memória interna, por exemplo: no caso de uma esteira transportadora, na qual cada peça é identificada por um sensor de presença e que, quando isso acontece, incrementa-se um contador armazenado em uma posição de memória, pode-se facilmente verificar quantos itens foram produzidos até determinado momento. O mesmo valeria se desejássemos verificar a velocidade do motor que movimenta a esteira.

Monitoramento de programas: função que permite, entre outras coisas, encontrar erros na implementação do programa do usuário por meio de monitoramento *online*. Isso possibilita verificar, em tempo real, a execução das lógicas do programa e também o estado dos dispositivos de entrada e saída, auxiliando muito na identificação da causa de algum problema de campo, por exemplo: caso o monitoramento de determinado trecho do programa acuse a energização de uma lâmpada e esta não esteja acesa, significa que ela está queimada ou existe algum rompimento de cabo.

Cada fabricante costumeiramente fornece a sua própria plataforma de desenvolvimento, sendo que sua disponibilidade pode ser gratuita ou não. Alguns exemplos dessas ferramentas são: SIMATIC Step7 (Siemens); RSLogix (Rockwell); MELSOFT (Mitsubishi). A maioria desses pacotes de software são exclusivos, ou seja, não é permitido que se desenvolva programas na plataforma de um fabricante para posteriormente descarregá-los em CLPs de outro.

Alguns CLPs possuem dois modos de funcionamento: modo de programação (PROG) e modo de execução (RUN). Quando o CLP está em modo de programação, isso significa que não há um programa sendo executado. Nesse momento, o CLP está apto a receber um novo programa desenvolvido pelo usuário por meio do procedimento de *download*. Uma das facilidades permitidas pelo uso da plataforma, é que o usuário pode desenvolver o programa em outro ambiente do qual se localiza o CLP, pois a conexão entre o PC e o CLP somente é necessária para descarregar e monitorar o programa e não para desenvolvê-lo. No modo de programação, também é permitido que se faça o procedimento de *upload*, isso é, a transferência de um programa que está na memória do CLP para o computador.

Quando o CLP está executando um programa, ele está em modo de execução. Nesse modo de operação é possível usar a plataforma de desenvolvimento para monitorar o andamento do programa, fazer algumas modificações se necessário, verificar as lógicas do processo e os valores das variáveis internas em tempo real (*online*). Existem modelos que permitem até mesmo fazer alterações na configuração de *hardware*.

Outra característica vantajosa do uso dos CLPs é que a partir do momento que a execução do programa estiver acontecendo conforme o esperado, o usuário pode desconectar o seu computador. Isso quer dizer que, assim que esteja configurado e programado, o CLP não tem necessidade de estar conectado ao computador para executar o programa do usuário. Dessa forma, é possível fechar a plataforma de desenvolvimento e cessar a comunicação entre eles, sem que o funcionamento do CLP seja interrompido.

FONTE DE ALIMENTAÇÃO

A função da fonte de alimentação é fornecer a corrente elétrica em níveis de tensão adequados para a CPU e para os demais módulos associados a ela. Ao desempenhar essa função, a fonte também serve como um meio de proteção para esses componentes, já que fornece certo isolamento em relação a possíveis problemas da rede elétrica, como sobretensão. Quando um problema desse tipo acontece, o fornecimento de energia é interrompido, protegendo o dispositivo como um todo.

A alimentação das fontes pode se dar com tensão alternada ou com tensão contínua e os valores variam de acordo com o modelo e o fabricante. Para tensões alternadas, são comumente utilizadas as faixas entre 85-132VAC, 176-264VAC e 85-264VAC. No caso das fontes alimentadas por tensão contínua, é possível encontrar valores em patamares de 12VDC, 24VDC, 10-28VDC, 125VDC (GEORGINI, 2018).

Com relação à tensão de saída, o valor de tensão mais comum é 24VDC. “Historicamente, a tensão de 24VDC foi bastante aplicada em relés na indústria, e isso acabou por manter-se: quando os CLPs e o controle por computadores começaram a ser uma realidade na indústria, eles simplesmente seguiram o padrão existente” (SILVEIRA; SANTOS, 1998). Esse valor padrão é o mesmo valor da tensão de alimentação da maioria das CPUs, dos módulos de entradas e saídas etc.

Para saber se a fonte é capaz de alimentar um determinado agrupamento de módulos, o valor de sua potência deve ser levado em consideração durante a etapa de dimensionamento. A potência da fonte deve ser maior que a soma das potências dos módulos que ela alimenta, somada a uma folga de segurança e a uma margem para possíveis expansões. Existem, inclusive, casos nos quais é necessário se utilizar mais de uma fonte.

Em alguns CLPs mais antigos, mais especificamente no caso dos fabricados antes da popularização dos cartões de memória flash, muitas informações (por exemplo, o programa do usuário) eram armazenadas em memórias não retentivas. Por isso, é possível encontrar fontes de alimentação (igualmente antigas) que possuem baterias internas, com a função de manter um fornecimento mínimo em sua saída em caso de interrupção da alimentação em sua entrada, com o objetivo de evitar a perda de dados. Mas com a popularização dos cartões de memória *flash*, esse recurso deixou de ser incorporado nas fontes de alimentação.

Vale ressaltar que a fonte não necessariamente deve ser do mesmo fabricante dos demais módulos, desde que possua as mesmas características elétricas e potência adequada.

ENTRADAS DIGITAIS

O termo “entradas digitais” é utilizado para designar os pontos de conexão que recebem sinais discretos, provenientes de dispositivos como: chaves de nível, termostatos, pressostatos, fluxostatos, botoeiras, chaves seletoras, chaves de fim de curso e outros. Esses dispositivos têm em comum a característica de operarem predominantemente em dois estados, como: energizado (ligado) ou desenergizado (desligado). Exemplos desses dispositivos conectados a um CLP podem ser observados na Figura 2.5.

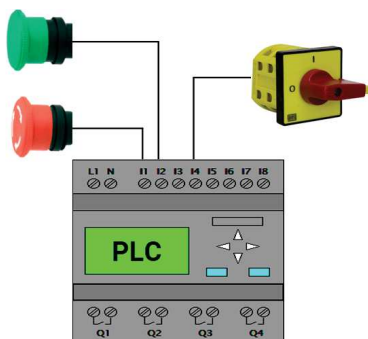


Figura 2.5: Diagrama esquemático que mostra dispositivos de entrada (botoeiras e chave seletora) conectados aos terminais de entrada digital de um CLP.

Alguns módulos de entrada digital permitem conexão com dispositivos que trabalham com tensão contínua, outros com tensão alternada. Para dispositivos que usam tensão contínua, é comum encontrar valores de entrada de 12VDC, 24VDC ou até 125VDC. Enquanto que para os dispositivos que usam tensão alternada, os valores mais comuns são 110VAC e 220VAC (GEORGINI, 2018).

Por isso, a eletrônica interna desses módulos é encontrada com algumas variações que se adequam ao tipo de sinal utilizado. Apesar dessas variações, os circuitos têm o comum objetivo de fornecer um isolamento para a CPU, a fim de protegê-la dos sinais elétricos fora do padrão.

Uma representação simplificada do circuito interno de um módulo de entrada que usa corrente contínua pode ser observada na Figura 2.6.

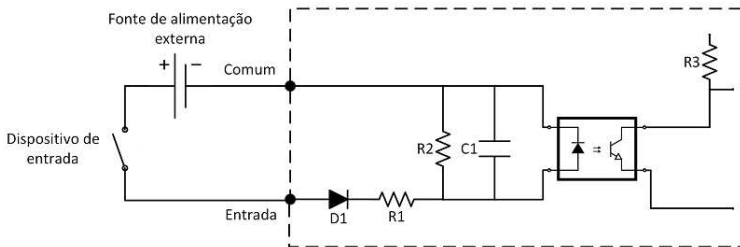


Figura 2.6: Representação simplificada do circuito interno de um módulo de entrada digital que é usado por dispositivos que usam tensão contínua.

Analisando a Figura 2.6, é possível observar que a partir do momento em que a entrada é acionada, a corrente percorre o caminho do pólo positivo da fonte de alimentação externa até o LED do optoacoplador que irá emitir uma luz que será detectada pelo fototransistor. É dessa forma que o circuito consegue detectar que a entrada está energizada, garantindo o isolamento elétrico entre o sinal de entrada e a CPU.

Explorando ainda a Figura 2.6, percebe-se que é a polaridade negativa da fonte externa que está conectada ao terminal comum do módulo. Se invertêssemos a polaridade da fonte externa e posteriormente o dispositivo de entrada fosse acionado, o sentido da corrente seria do terminal comum até o terminal de entrada. Mas como o LED do optoacoplador estaria polarizado inversamente, a corrente não iria percorrer todo o circuito e, conseqüentemente, o LED não emitiria luz para o fototransistor. A conseqüência disso é que a CPU não conseguiria detectar o acionamento da entrada.

A possível não detecção do acionamento da entrada por causa da inversão da polaridade da fonte no circuito acontece porque quando se usa tensão contínua para um módulo de entrada digital, existem duas variações de configuração: configuração *current-sinking* (consumidora de corrente) e configuração *current-sourcing* (fornecedora de corrente). Essas duas configurações definem qual a polaridade da fonte que deve estar conectada ao terminal comum do módulo e se a conexão for feita de forma invertida, não haverá danos, mas o módulo não conseguirá detectar o acionamento da entrada de forma correta.

A configuração *current-sinking* implica que o negativo da fonte externa deve estar conectado ao comum do módulo de entrada, como na Figura 2.6. Isso faz com que quando o dispositivo de entrada for acionado, de acordo com o sentido convencional da corrente elétrica, a corrente irá “entrar” através do terminal de entrada e “sairá” pelo terminal comum. Nesse caso, é comum usar a terminologia de que o ponto de entrada é “consumidor de corrente” ou que as entradas são do tipo *sinking*.

Para a configuração *current-sourcing*, é o positivo da fonte que deve estar conectado ao comum do módulo. Sendo assim, para que a CPU consiga identificar que a entrada foi acionada, dentre outras mudanças na eletrônica, o LED do optoacoplador deve estar disposto no sentido contrário ao da configuração *current-sinking*. Nesse caso, quando a entrada for acionada, a corrente “entrará” pelo terminal comum e “sairá” através do terminal de entrada, como se esse terminal fosse um “fornecedor de corrente”. Usando a terminologia, diz-se que as entradas são do tipo *sourcing*. Na Figura 2.7, é possível perceber essa diferença com maior clareza.

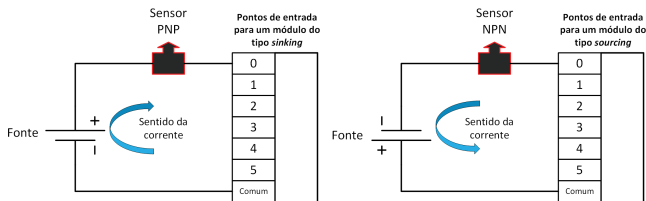


Figura 2.7: Diferença entre as entradas do tipo *sinking* e *sourcing* de acordo com a polaridade da fonte e o sentido da corrente.

A designação das entradas em *sinking* e *sourcing* pode ser encontrada nos manuais desses módulos e faz parte de suas especificações técnicas. Essa informação é importante inclusive para determinar quais sensores podem ser usados de acordo com o tipo da entrada. Entradas do tipo *sinking* somente permitem o uso de sensores do tipo PNP e entradas do tipo *sourcing* permitem apenas o uso de sensores do tipo NPN (Figura 2.8). Sensores PNP e NPN fazem parte de um grupo de sensores que utilizam transistores para fazer o chaveamento do sinal elétrico, mas se o sensor a ser utilizado for composto por apenas um contato mecânico (como botoeiras, por exemplo), não há distinção de entrada (*sinking* ou *sourcing*) a qual ele possa se conectar.

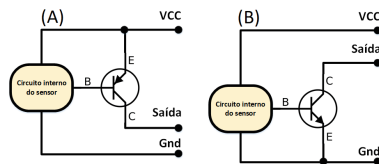


Figura 2.8: Comparativo entre dois sensores de proximidade: (A) PNP, indicado para módulos com entradas do tipo *sinking*; e (B) NPN, indicado para módulos com entradas do tipo *sourcing*.

Apesar dos módulos terem essas duas configurações (*current-sinking* e *current-sourcing*) para as entradas digitais que usam corrente contínua, existem casos em que é possível conectar tanto sensores PNP quanto NPN aos terminais de entrada. Isso acontece porque, dentre outras características, são usados dois LEDs dispostos em antiparalelo no optoacoplador. Essa alteração na eletrônica tira a obrigatoriedade de o terminal comum estar conectado exclusivamente com o polo positivo ou com o polo negativo da fonte externa.

Quando o módulo de entrada digital é disponibilizado para conexão com dispositivos que são alimentados com tensão alternada, as terminologias de entrada do tipo *sinking* e *sourcing* não são utilizadas devido ao fato de que esse sinal não tem polaridade. Para esse circuito (Figura 2.9), nota-se a existência de uma ponte retificadora, além da permanência do uso do optoacoplador, que ainda é usado com a finalidade de isolar o processador dos sinais elétricos.

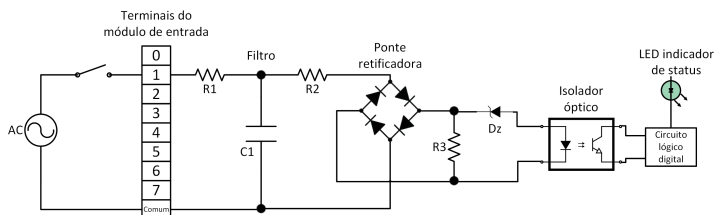


Figura 2.9: Circuito eletrônico do módulo de entrada digital para conexão de dispositivos que usam tensão alternada.

Apesar dos módulos de entrada digital geralmente serem usados para detectar dois estados (ligado e desligado), em alguns casos, fabricantes disponibilizam algumas de suas entradas para funcionarem como contadores rápidos ou RSC (do termo em inglês *Rapid Speed Counter*). Isso permite realizar outros tipos de medição como vazão, posição e velocidade de motores, que apresentam mais do que dois estados. Para realizar essas medições, é contabilizado a quantidade de pulsos elétricos que esses sensores enviam em um intervalo definido de tempo.

Por exemplo, na Figura 2.10 é representado um medidor de vazão que envia sinais pulsados para uma entrada de contagem rápida do CLP. O usuário pode desenvolver um programa que consulte, em intervalos fixos de tempo, o número de pulsos que serão armazenados em uma posição de memória. Dessa forma, basta subtrair a quantidade de pulsos atual da quantidade anterior, obtendo assim quantos pulsos ocorreram no intervalo. De posse desse resultado, sabendo-se que cada pulso representa um determinado volume de líquido que passou pelo medidor, bastam simples manipulações algébricas para se encontrar o volume apurado no intervalo e, posteriormente, expressar a vazão na unidade em que se deseja.

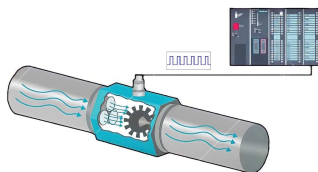


Figura 2.10: Medidor de vazão com sinais pulsados de saída sendo usado por uma entrada de contagem rápida de um CLP.

ENTRADAS ANALÓGICAS

No caso em que o CLP recebe sinais elétricos que variam dentro de um intervalo preestabelecido, sendo convertidos em valores decimais que posteriormente são armazenados em sua memória, temos as popularmente conhecidas “entradas analógicas” que são, na verdade, conversores do tipo ADC (do termo em inglês, *Analog to Digital Converter*). A essas entradas, geralmente, são conectados medidores de grandezas como nível, pressão, vazão, podendo-se também conectar potenciômetros para servirem como interface de operação (apesar de que essa seja uma prática já em desuso).

Os fabricantes disponibilizam os módulos de entrada analógica de modo que eles possam detectar valores em um intervalo específico de tensão ou corrente. Os padrões mais utilizados para tensão são: 0V a 5V, 0V a 10V, $\pm 10V$, $\pm 5V$, $\pm 2.5V$, $\pm 1.25V$. Para corrente, geralmente, esses valores se encontram nos intervalos de 4 a 20 mA ou 0 a 20 mA (AG, 2021).

A seguir, pode-se observar como é o circuito para quando é utilizada uma entrada por tensão (Figura 2.11) e como é o circuito para quando se usa a entrada por corrente (Figura 2.12). Nota-se que ambos circuitos utilizam o conversor ADC para que o sinal elétrico possa ser convertido em valores digitais que a CPU irá armazenar em uma posição de memória. É interessante perceber que apesar de permitir entrada em corrente, para que o conversor funcione, primeiramente essa corrente é convertida em um valor proporcional de tensão.

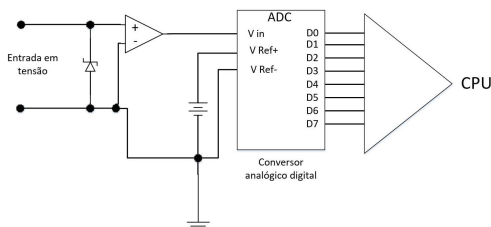


Figura 2.11: Diagrama elétrico de uma entrada analógica que usa valores de tensão como entrada para expressar grandezas físicas.

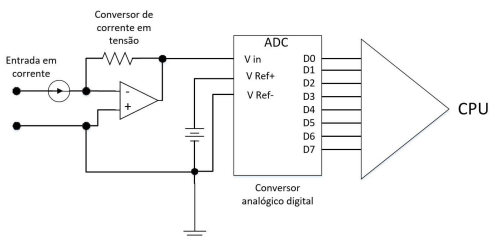


Figura 2.12: Diagrama elétrico de uma entrada analógica que usa valores de corrente como entrada para expressar grandezas físicas.

Por se tratar de circuitos diferentes, as especificações dos módulos informam qual o padrão (tensão ou corrente) é utilizado. No entanto, existem fabricantes que possibilitam o uso de ambos no mesmo terminal de entrada do módulo, como o modelo SM 1231 da SIEMENS, que permite entradas de tensão nos intervalos de $\pm 10V$, $\pm 5V$, $\pm 2,5V$ ou intervalos de corrente de 0 a 20mA e 4 a 20mA (AG, 2021). Nesse caso, o padrão a ser usado, assim como a faixa de valores, devem ser declarados na plataforma de desenvolvimento na etapa da configuração de hardware. Em CLPs antigos, a escolha entre o uso do padrão por tensão ou corrente era feito por meio de *dip switches*, como o modelo 1762-IF2OF2, da Rockwell.

Assim como as entradas que trabalham recebendo sinais de corrente e transformam em tensão antes de serem codificadas, existem módulos que tratam os sinais para fazer a medição de temperatura, incorporando circuitos preliminares como a ponte de Wheatstone, que é utilizada por sensores para fazer o condicionamento eletrônico de sinais fornecidos por medidores de temperatura. Esses módulos permitem que se faça ligações a 2, 3 e até 4 fios.

Na Figura 2.13 é possível observar como fica a ligação de uma termoresistência do tipo PT-100 em um módulo desse tipo usando a configuração a 3 fios, e qual o circuito que está incorporado no módulo (AG, 2016).

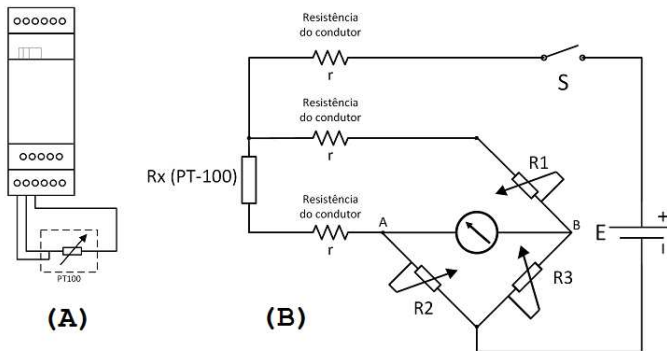


Figura 2.13: Representação da conexão do sensor de temperatura PT100 no módulo AM2 RTD da SIEMENS (A) e configuração a 3 fios usando ponte de Wheatstone (B).

Apesar desses medidores conseguirem expor diferentes valores dentro de um intervalo definido, é importante saber que existem limitações que são representadas por uma característica do ADC chamada “resolução”.

A resolução indica qual a menor variação da grandeza que o módulo de entrada analógica consegue detectar. Quanto melhor a resolução, maior a capacidade de percepção do dispositivo e, com isso, se consegue valores mais precisos. Para se determinar a resolução é usado o número de *bits* disponível nos catálogos, sendo que, quanto maior o número de *bits* (N), melhor a resolução, como demonstrado na Equação 2.1.

$$\text{Resolução} = \frac{\text{Intervalo}}{2^N} \quad (2.1)$$

Para entender melhor como a resolução interfere na representação de um sinal, pode-se imaginar uma situação na qual um módulo de entrada analógica trabalha com variações de tensão em um intervalo de 0V a 10V e que, de forma fictícia possua apenas 2 *bits* de resolução (os conversores práticos possuem uma quantidade de *bits* significativamente maior). A conversão desse sinal pode começar a ser entendida aplicando-se a Equação 2.1 nessa situação, de forma que:

$$\text{Resolução} = \frac{\text{Intervalo}}{2^N} = \frac{10V - 0V}{2^2} = 2,5V$$

Dessa forma, é possível concluir que a menor variação de tensão possível de ser detectada pelo uso de um ADC fictício de apenas 2 *bits* no módulo de entrada analógico seria de 2,5V.

Isso significa que o intervalo é dividido em segmentos de 2,5V, ou seja, quando os valores de 0V até 2,5V estiverem sendo aplicados na entrada, o ADC enviará o mesmo sinal digital (00) para a CPU. Na Tabela 2.1, é mostrado a correspondência entre os valores de tensão da entrada e os valores binários do ADC.

Tabela 2.1: Correspondência entre os valores de tensão na entrada de um ADC e o valor binário da saída.

Valor de tensão na entrada [V]	Valor binário do ADC
7,5 à 10	11
5 à 7,5	10
2,5 à 5	01
0 à 2,5	00

Para esse exemplo fictício, foi obtida uma resolução de 2,5V que é um valor enorme, se comparado ao intervalo total de 10V. No entanto, se for aumentado o número de bits em 1 unidade (3 bits), a resolução cai pela metade (1,25 V), o que torna a representação do sinal mais precisa. Módulos de entrada analógica são disponibilizados geralmente com uma resolução de 10 a 15 bits.

Independentemente da resolução dos conversores, o valor gerado por eles geralmente é armazenado em uma posição de memória de 16 bits (mesmo havendo aparente desperdício). Essa informação é relevante para se entender como as entradas analógicas são endereçadas na memória do CLP, tópico este que será discutido na Seção 2.3.

SAÍDAS DIGITAIS

O termo “Saídas digitais” é usado para designar os terminais elétricos de um módulo no qual podem ser conectados dispositivos como: indicadores luminosos, relés, sirenes, contadores, válvulas, motores, cilindros pneumáticos e outros dispositivos que trabalham em dois estados (energizado/desenergizado).

Assim como os módulos de entradas digitais, os módulos de saídas digitais também permitem a conexão de dispositivos que trabalham com tensão contínua ou tensão alternada. Por essa característica, existem variações na eletrônica interna dos cartões. Basicamente, existem três tipos de saídas digitais, sendo elas:

Transistor: módulos desse tipo são utilizados para quando se deseja conectar dispositivos que são alimentados com tensão contínua. Os valores mais utilizados são de 5VDC, 12VDC, 24VDC e 125VDC (GEORGINI, 2018). As principais características de uma saída que usa transistor são a rapidez e a durabilidade, visto que esse componente eletrônico pode suportar alto número de comutações em um curto intervalo de tempo, sem ter seu funcionamento comprometido, além de ter um tempo de resposta, na ordem de nanosegundos. Os módulos de saída digitais com acionamento por transistor também podem ter configuração do tipo *current-sinking* e *current-sourcing*. A configuração *current-sinking* é caracterizada pela conexão do polo negativo da fonte externa com o terminal comum do módulo. Dessa forma, quando o dispositivo de saída estiver energizado, o terminal no qual ele está conectado funciona como “consumidor de corrente”. Por outro lado, quando é o polo positivo da fonte que está conectado ao terminal comum do módulo, assim que o dispositivo de saída estiver energizado, o terminal de saída funciona como um “fornecedor de corrente”, ou seja, as saídas são do tipo *sourcing*. Na Figura 2.14 é apresentado de forma esquemática como são essas configurações.

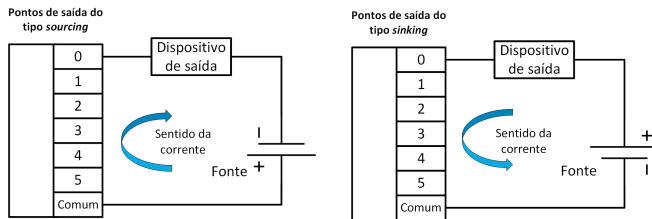


Figura 2.14: Diferença entre as saídas do tipo sinking e sourcing de acordo com a polaridade da fonte externa e o sentido da corrente.

Triac: esse tipo de saída é usado para acionar dispositivos que usam tensão alternada em sua alimentação. Entre os valores mais comuns estão 24VAC, 110VAC e 220VAC (GEORGINI, 2018). O triac é um componente que apresenta características semelhantes ao transistor com relação à rapidez e durabilidade. Na Figura 2.15 é possível observar a eletrônica interna de uma saída que usa esse componente.

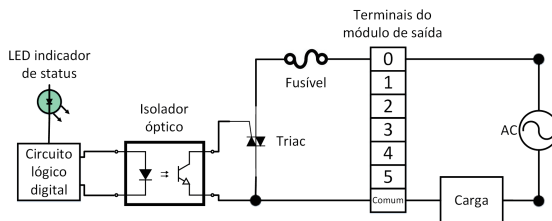


Figura 2.15: Eletrônica interna de um módulo que usa triac para fazer o acionamento das saídas.

Micro relés: o uso de micro relés para acionar os dispositivos de saída acontece por meio da energização de sua bobina, independentemente do sinal de tensão que o dispositivo de saída use em sua alimentação, como demonstrado na Figura 2.16. Por isso, micro relés podem ser usados tanto em tensão contínua quanto em tensão alternada. Apesar dessa versatilidade, as saídas que usam esse tipo de componente têm a característica de serem mais lentas quando comparadas aos outros dois tipos (transistor e triac), com um tempo de acionamento na ordem de milissegundos. Além disso, por se tratar de um dispositivo que é composto por partes mecânicas, é suscetível a maior desgaste resultando em uma menor durabilidade.

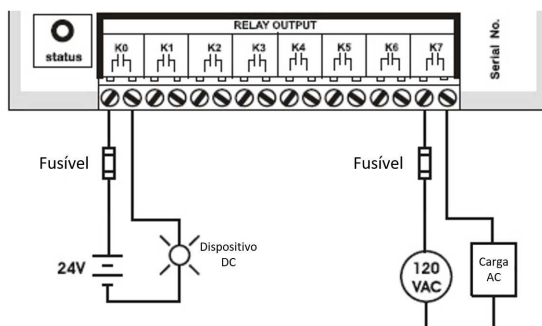


Figura 2.16: Ligação de dispositivos que usam tensão contínua e alternada em um módulo de saída digital cujo acionamento é feito por micro relés.

Além do tipo de componente (transistor, triac e micro relés), outra especificação técnica que deve ser considerada para a escolha de um módulo de saída digital é a seu limite de corrente. Geralmente, essa informação vem exposta de duas formas: máxima corrente permitida por ponto de saída, ou corrente máxima por terminal comum.

Módulos com acionamento feito por transistor e triac, normalmente, são disponibilizados com um ponto de conexão comum que é usado por todos os outros pontos de saída para fechar o circuito, como é possível observar na Figura 2.14. Dessa forma, usualmente, a informação sobre o limite de corrente é fornecida considerando a corrente máxima que pode circular pelo terminal comum.

Para o caso de saídas que usam micro relés, normalmente, existem dois terminais para cada saída, como demonstrado na Figura 2.16. Por isso, é conveniente que a informação de corrente máxima seja dada individualmente por saída.

Apesar das limitações quanto ao valor máximo de corrente, o CLP pode ser usado quando há a necessidade de acionar cargas que demandam correntes superiores (ou para acionar cargas trifásicas). Para isso, geralmente se usa contadores cujo princípio de funcionamento é idêntico ao dos relés (a abertura ou fechamento dos contatos são conduzidos pela energização de sua bobina), mas são usados para acionar cargas de alta potência, como mostrado na Figura 2.17.

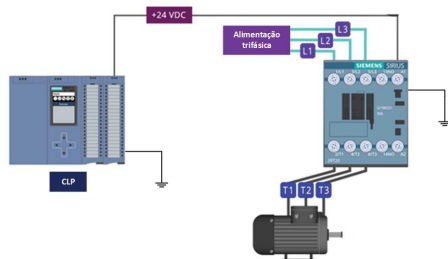


Figura 2.17: CLP acionando um motor trifásico de alta potência com o uso de um contatores.

Outra consideração interessante diz respeito ao fato de que, apesar dos módulos de saída digital serem usados predominantemente para acionar dispositivos que possuem dois estados, em alguns casos, também é possível usá-las para controlar dispositivos de maneira contínua (não discreta), por meio da técnica conhecida como Modulação por Largura de Pulso, PWM (do termo em inglês, *Pulse Width Modulation*). Alguns fabricantes disponibilizam terminais de seus módulos de saída para fornecer sinais PWM e assim permitir que haja o controle de dispositivos como válvulas proporcionais, bombas dosadoras, sistemas de aquecimento e outros.

Uma válvula proporcional (Figura 2.18) pode receber como referência de abertura, um sinal de tensão gerado por um circuito PWM que controla seu valor médio, pois, esse sinal consiste em uma série de pulsos “on” e “off” que variam o *duty cycle* (tempo ligado) (GEMS SENSORS AND CONTROLS, 2021). O *duty cycle* é uma medida em porcentagem da quantidade de tempo em que a tensão de alimentação está “ligada” (em comparação com a quantidade de tempo em que está “desligada”). Um *duty cycle* de 100% significa que a tensão média estaria em nível máximo, enquanto um *duty cycle* de 0% significa que a tensão seria nula, como mostrado na Equação 2.2.

$$\text{Tensão média [V]} = \text{Tensão de Alimentação [V]} \times \text{Duty Cycle [\%]} \quad (2.2)$$

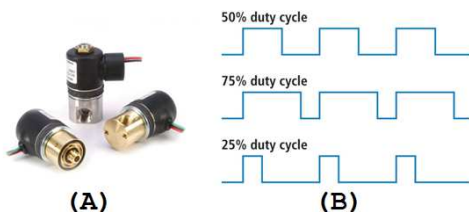


Figura 2.18: Válvula proporcional (A) cuja abertura é controlada por um valor modulado de tensão (B).

O *hardware* do CLP é o responsável por gerar esse sinal pulsado com o *duty cycle* adequado. Isso é feito, por exemplo, com o armazenamento de um valor entre 0 e 32767, (dependendo do modelo e fabricante) em uma posição de memória específica, que corresponde respectivamente a 0% e 100%. Dessa forma, modulando a duração dos pulsos de tensão, o percentual de abertura da válvula é controlado, e conseqüentemente a quantidade de fluxo através dela.

Além de poder controlar o *duty cycle*, há casos em que características secundárias como a amplitude e a frequência dos pulsos podem ser escolhidas por meio da configuração de *hardware*, já em outros, estas são fixas e devem ser consideradas quando se adquire o módulo de saídas.

SAÍDAS ANALÓGICAS

Valores decimais armazenados na memória do CLP podem ser convertidos em sinais elétricos por meio das saídas analógicas, historicamente utilizadas para transmitir referências elétricas a elementos como inversores de frequência e válvulas proporcionais, controlando-se assim grandezas como velocidade e posição. Esses conversores são do tipo DAC (do termo em inglês, *Digital to Analog Converter*) e, assim como as entradas analógicas, são oferecidos pelos fabricantes em módulos com diferentes características, nos quais pode-se encontrar conversores que trabalham com sinais nos intervalos padrões de 0 a 10V, $\pm 10V$, $\pm 5V$, $\pm 2.5V$, 4 a 20mA e 0 a 20mA (AG, 2021).

Na Figura 2.19 é apresentado de forma simplificada como é o circuito eletrônico interno de uma saída analógica. É possível notar que a CPU envia um sinal codificado em bits para o DAC que o converte no sinal elétrico de tensão proporcional.

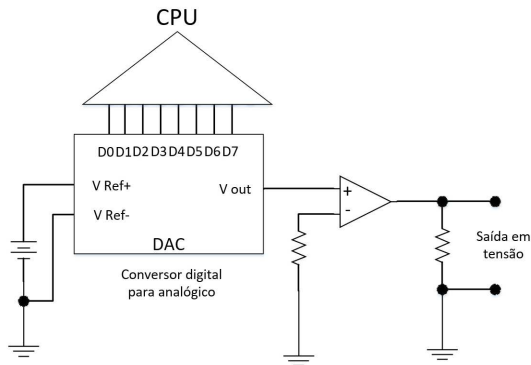


Figura 2.19: Eletrônica interna de uma saída analógica que fornece um sinal de tensão.

Assim como os módulos de entrada analógica, um módulo de saída analógica também pode operar com sinais de corrente. Isso acontece com a adição de um circuito que converte o sinal de tensão em um sinal proporcional de corrente, como mostrado na Figura 2.20.

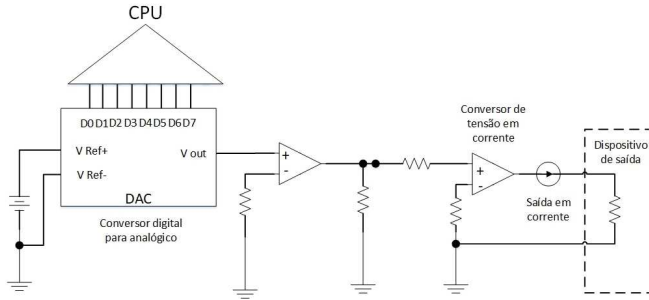


Figura 2.20: Eletrônica interna de uma saída analógica que fornece um sinal de corrente para o dispositivo de saída.

Como mencionado na abordagem sobre os módulos de entrada analógica, alguns módulos de saída também permitem que se conecte dispositivos que funcionam com tensão ou com corrente no mesmo terminal. Para isso, basta selecionar qual o padrão a ser usado e a faixa de valores no momento de configurar o hardware na plataforma de desenvolvimento. Alguns modelos antigos permitem essa escolha alterando a posição de algumas chaves (*dip switch*) fisicamente no módulo.

A resolução e a faixa de trabalho também são características importantes para os módulos de saída analógica. A resolução, expressa em número de *bits*, nesse caso é utilizada para corresponder o intervalo de valores de uma determinada variável a um nível tensão (ou corrente) que o módulo deverá fornecer.

Por exemplo, supondo que um módulo fictício possuísse resolução de 2 *bits* e que sua saída fosse no intervalo de 0V à 10V, significaria que as variações de tensão fornecidas seriam de 3,33V em 3,33V, já que com 2 bits é possível obter apenas quatro estágios, com dois valores entre 0V e 10V, como mostrado na Figura 2.21.

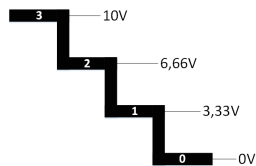


Figura 2.21: Representação dos valores de tensão que um módulo de saída analógico fictício de 2 bits no padrão 0V à 10V pode fornecer.

Normalmente, o conversor recebe um valor de uma variável do tipo “inteiro” da CPU, que é armazenado em um espaço de 16 bits de memória, podendo expressar valores que vão de 0 a 65535. Continuando com o exemplo do conversor fictício de 2 bits, pode-se usar a Equação 2.3 para determinar os intervalos que irão corresponder aos valores de tensão mostrados na Tabela 2.2.

$$\text{Resolução} = \frac{\text{Intervalo}}{2^N} = \frac{65536}{2^2} = 16384 \quad (2.3)$$

Tabela 2.2: Correspondência entre os valores decimais na entrada do DAC e o valor de tensão da saída.

Valor decimal do DAC	Valor de tensão na saída [V]
49152 à 65535	10
32768 à 49151	6,66
16384 à 32767	3,33
0 à 16383	0

De acordo com o valor obtido no cálculo da resolução, obtém-se a Tabela 2.2 que faz a correspondência entre os valores decimais da entrada do DAC e sua saída em tensão.

De acordo com a Figura 2.21, fazendo uma analogia com uma escada, 2 bits correspondem a 4 degraus e aumentar o número de bits significa que a escada terá maior número de degraus e cada degrau equivalerá a um incremento menor no valor de tensão, pois, os valores de 0V e 10V são mantidos. Isso resulta em uma representação mais precisa do sinal que se deseja enviar para o dispositivo de saída.

Na realidade, esse tipo de módulo é fornecido com um número maior de bits, podendo chegar até 15 bits que resultaria em degraus com variação de apenas 0,31mV.

Assim como para as entradas analógicas, independentemente do número de bits da resolução dos conversores, o valor em que eles se baseiam é geralmente armazenado em uma posição de memória de 16 bits. Nesse momento pode parecer que existe um desperdício de memória, mas na Seção 2.3 será explicado porque isso acontece.

CONSIDERAÇÕES FINAIS: Os fabricantes disponibilizam os módulos de entrada e saída com quantidades diferentes de pontos de conexão. Alguns são exclusivos para entradas, outros são exclusivos para saídas, existem ainda módulos mistos que possuem pontos para entradas e saídas. Abaixo tem-se alguns modelos da família S7-1200 que exibe essas variações (AG, 2021).

Modelo	Pontos de entrada	Pontos de saída
SM 1221 (digital)	8 ou 16 (24VDC)	-
SM 1222 (digital)	-	8 ou 16 (transistor ou micro relé)
SM 1223 (digital)	8 ou 16 (24VDC)	8 ou 16 (transistor ou micro relé)
SM 1231 (analógico)	4 ($\pm 10\text{VDC}$, $\pm 5\text{VDC}$, $\pm 2.5\text{VDC}$, 4..20 mA ou 0..20 mA)	-
SM 1232 (analógico)	-	4 ($\pm 10\text{VDC}$, 4..20 mA ou 0..20 mA)
SM 1234 (analógico)	4 ($\pm 10\text{VDC}$, $\pm 5\text{VDC}$, $\pm 2.5\text{VDC}$, 4..20 mA ou 0..20 mA)	2 ($\pm 10\text{VDC}$, 4..20 mA ou 0..20 mA)

Todos esses modelos são utilizados por dispositivos que operam por meio do uso dos padrões elétricos (tensão e corrente), mas existe uma tendência de migração desses dispositivos por sensores e atuadores inteligentes que tem a capacidade de se comunicar via rede de dados. Portanto, esses módulos ainda muito usados na indústria, devem ser substituídos no futuro.

PROCESSADORES DE COMUNICAÇÃO

Esse tipo de módulo tem o objetivo de agregar capacidade de comunicação aos CLPs. Como visto, é comum que o módulo da CPU inclua uma ou duas portas de comunicação, algumas vezes com protocolos distintos. No entanto, como a diversidade de protocolos para redes industriais é muito grande, muitas vezes faz-se necessário incluir novos módulos de comunicação em um projeto, permitindo que um CLP passe a se comunicar em outros “idiomas”.

Para um melhor entendimento dos Processadores de Comunicação (CPs, do termo em inglês *Communications Processors*) é interessante observar em que ponto os CLPs estão localizados na famosa “Pirâmide de automação” (Figura 2.22).

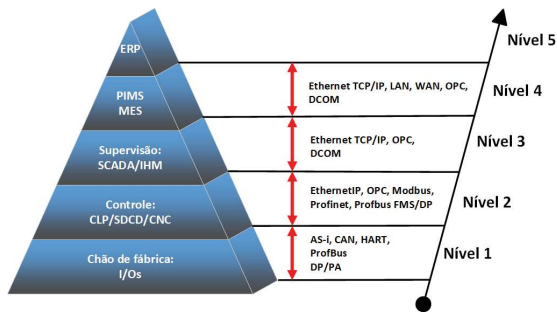


Figura 2.22: Pirâmide da automação disposta em 5 níveis.

Os cinco níveis da pirâmide podem ser resumidos da seguinte forma: 1) Chão de fábrica, composto por sensores e atuadores que, inclusive, podem ter a capacidade de comunicação em rede de dados; 2) Controle, composto por dispositivos e sistemas como CNCs (Comando Numérico Computadorizado), SDCDs (Sistemas Digitais de Controle Distribuído) e CLPs, que podem usar as redes de dados para se comunicar com dispositivos do mesmo ou de qualquer outro nível da pirâmide; 3) Supervisão, camada em que se faz a Interface Homem-Máquina, por meio de sistemas SCADA, Painéis Eletrônicos de Operação (PEOs) e similares; 4) Gerenciamento da produção, que ocorre principalmente por meio de elementos como MES (do termo em inglês *Manufacturing Execution System*) e PIMS (do termo em inglês *Plant Information Management System*); 5) Gerenciamento corporativo, geralmente composto por pacotes de software responsáveis pela administração dos recursos de uma ou mais unidades da empresa, normalmente conhecidos como ERP (do termo em inglês *Enterprise Resource Planning*), que acompanham a produção em conjunto com dados administrativos e financeiros.

Conforme mencionado, os CLPs podem se comunicar com dispositivos do mesmo nível da pirâmide (outros CLPs, por exemplo), com elementos da camada inferior (sensores e atuadores inteligentes) ou das camadas superiores (sistemas SCADA e PIMS são os exemplos bastante comuns). Além disso, ainda observando-se a Figura 2.22, pode-se ver uma amostra da grande variedade de protocolos de rede que podem ser utilizados em cada um dos mencionados tipos de comunicação.

Sabe-se que esses protocolos apresentam características distintas e, em muitos casos, requerem portas de comunicação diferentes. Então, os processadores de comunicação são usados de modo a viabilizar a comunicação com esses diferentes protocolos e disponibilizar essas diferentes portas.

A comunicação com os dispositivos do Nível 3 acontece com os sistemas supervisórios e de interface, que podem usar, por exemplo, o protocolo MODBUS por meio de porta serial (RS232/RS485) ou Ethernet. Um exemplo de dispositivo de Nível 2 com o qual um CLP pode se comunicar é outro CLP ou, mais especificamente, a família S7-1200 CPU 1214C, da SIEMENS, permite comunicação com o protocolo Profinet. Para esse modelo conseguir se comunicar com o modelo S7-300, que por padrão se comunica usando o protocolo Profibus, é necessário acrescentar o módulo de comunicação CM 1242-5 (Figura 2.23). Somente dessa forma, o S7-1200 tem a capacidade de se comunicar com o protocolo Profibus, o que torna a comunicação entre o S7-1200 e o S7-300 possível.



Figura 2.23: Módulo de comunicação CM 1242-5 SIEMENS, que permite com que a família S7-1200 se comunique com o protocolo Profibus.

No Nível 1, a comunicação acontece, por exemplo, com sensores e atuadores inteligentes presentes no chão de fábrica que podem usar protocolos como CAN, Profibus, AS-i e outros. Outra forma de comunicação nesse nível é usando as Unidades Remotas.

Unidades Remotas podem formar *racks* secundários com a conexão de módulos de entradas e saídas. Elas também possuem um processador de comunicação e sua função é facilitar a comunicação do CLP com sensores e atuadores que estejam localizados a grandes distâncias, permitindo que a CPU do *rack* principal se comunique com os dispositivos do *rack* secundário por meio de um único cabo.

Imaginando que um CLP seja o responsável por fazer o controle e o monitoramento de uma planta industrial distribuída em uma área da qual a distância entre o CLP e os sensores/atuadores seja elevada, por exemplo 200 metros. Fazer a conexão dos sensores/atuadores usando um fio por dispositivo, como na Figura 2.24, teria grandes desvantagens, dentre as quais pode-se citar:

- Desorganização e dificuldade para reparos e expansão, organizar altas quantidades de fios e localizar um ponto de possível rompimento seria muito difícil, além de que, acrescentar uma nova entrada ou saída no processo exigiria muito esforço;
- Custo elevado, pois, seria exigido alta quantidade de cabos com grandes comprimentos;
- Perdas de sinal, para sensores e atuadores que funcionam usando o padrão de tensão de 0 à 10V, por exemplo, poderiam ocorrer quedas de tensão devido a resistência dos condutores.

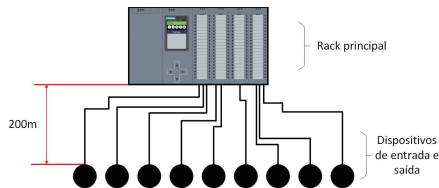


Figura 2.24: Representação de um CLP trocando informação com dispositivos de campo no qual é usado um cabo por dispositivo.

Uma forma de solucionar esses problemas é usar uma Unidade Remota, como a ET200SP da SIEMENS, que pode ser disposta em um rack secundário com a conexão de módulos de entrada e saída próximos aos sensores e atuadores, o que exige menor comprimento de condutores para a conexão com esses dispositivos de chão de fábrica. Dessa forma, para que a CPU do rack principal consiga se comunicar com todos os dispositivos que estão a 200m de distância, basta usar um único cabo Profinet conectado com a ET200SP, como na Figura 2.25.

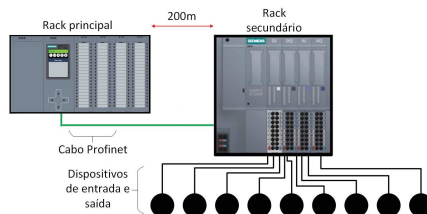


Figura 2.25: Representação de um CLP trocando informação com dispositivos de campo por meio de uma Unidade Remota usando um cabo Profinet.

As unidades remotas tem grande importância para que a comunicação com dispositivos de chão de fábrica que usam padrões de sinais de tensão e corrente seja facilitada. No entanto, com a elevação do uso de tecnologias inteligentes que caracterizam a indústria 4.0, existe uma tendência de que esses dispositivos que transmitem informação por meio de sinais elétricos sejam substituídos por dispositivos inteligentes. Um dispositivo desse tipo pode fazer mais do que detectar propriedades físicas básicas, eles também podem realizar a conversão de dados, processamento digital, e podem se comunicar com dispositivos externos por meio de uma plataforma de serviço em nuvem.

Um exemplo desse tipo de dispositivo são os sensores inteligentes, que além de funcionarem como os sensores convencionais, são equipados com um microprocessador e são capazes de se comunicar via rede de dados. Isso permite que algumas funções sejam incorporadas, como realizar diagnósticos e auto calibrações, que dentre outras coisas, permitem detectar falhas, como problemas de contato e até mesmo problemas ocasionados por sujeiras. Alguns sensores inteligentes são capazes de aferir mais do que uma grandeza física como pressão, temperatura, umidade, fluxo de gás em um único dispositivo.

Como esses sensores são equipados com microprocessador e podem se comunicar, um CLP não tem mais a necessidade de processar os sinais elétricos como os adotados pelos padrões 0 a 10v e 4 a 20mA, a informação pode ser recebida já com o valor na unidade desejada diretamente da nuvem, diminuindo inclusive a quantidade de cabos para transmitir os dados. Apesar dessa vantagem, sensores que usam os padrões elétricos ainda são muito utilizados por serem duráveis e de menor custo.

Dispensar o uso de cabos para se comunicar, não é uma tendência exclusiva dos dispositivos de chão de fábrica que vêm sendo substituídos pelos sensores inteligentes, pois existem também Processadores de Comunicação, como o modelo CP 1242-7 da SIEMENS, que permitem realizar comunicação sem fio entre duas CPUs usando a tecnologia GSM/GPRS, além de possibilitar o envio de mensagens SMS para telefones celulares, o que pode vir a ser útil para enviar alertas de um evento importante e alarmes.

CPU

Geralmente, posicionado fisicamente ao lado da fonte, a CPU pode ser considerada o principal módulo do dispositivo. Tradicionalmente, abriga o processador principal do CLP, as memórias volátil e não-volátil e pelo menos uma porta de comunicação, que permite entre outras coisas, que o sistema receba de um PC convencional as configurações de *hardware* e o *software* aplicativo a ser processado. Além disso, é nesse módulo que se encontra o *firmware* do CLP, responsável por gerenciar todo o seu funcionamento, incluindo o ciclo de leitura das entradas, processamento do programa e atualização das saídas. Algumas vezes, pode-se encontrar mais de um tipo de porta de comunicação e também algumas entradas e saídas elétricas embutidas no módulo da CPU

O processador, um dos principais componentes da CPU, interage continuamente com o sistema de memória e é o responsável por controlar o barramento de endereços e a comunicação, entre suas principais funções está a interpretação e execução do programa do usuário, além do gerenciamento de todo o sistema por meio de diagnósticos.

Para análise do desempenho de um processador, é considerado sua habilidade em manipular dados, conhecida como poder de processamento. O poder de processamento varia de acordo com a arquitetura usada e, nos catálogos, um parâmetro adotado para mensurar

esse quesito são os tempos de processamento para manipular dados binários e decimais, sendo que esses dados podem estar relacionados às variáveis criadas pelo próprio usuário e também às variáveis atreladas aos cartões de entradas e saídas.

Outro componente principal da CPU são as memórias, que são responsáveis por armazenar os dados e também o firmware do CLP, que basicamente dita como será seu funcionamento, desde a configuração do hardware, até a atualização das saídas e comunicação com outros dispositivos. Existem dois tipos de memória disponíveis em uma CPU: volátil e não volátil.

A memória volátil, também chamada de memória RAM, tem como característica ser rápida para escrita e leitura de dados e, por isso, é usada para armazenar variáveis de entrada e saída, registradores de contagem e temporização, que podem sofrer alteração em todos os ciclos de varredura, essa memória necessita de alimentação elétrica para que seus dados não sejam perdidos.

A memória não volátil não é tão veloz quanto a memória RAM, mas tem a capacidade armazenar os dados mesmo com a ausência de energia elétrica e, por essa característica, é usada para armazenar o firmware e também o programa do usuário. Dessa forma, se houver uma queda de energia, a CPU permanece configurada e programada.

A quantidade de memória disponível também é um parâmetro que deve ser considerado para a escolha de uma CPU, pois, há casos em que a CPU é disponibilizada com uma quantidade fixa de memória e não possibilita o aumento de sua capacidade. Em outros casos, é possível expandi-la com o uso de cartões de memória FLASH com capacidade de 4MB, 12MB, 256MB, 2GB até 32GB. Portanto, por ser o principal componente do CLP, a escolha do modelo de CPU para um projeto deve ser feita com muito cuidado, pois, para o seu dimensionamento são atrelados fatores essenciais, dos quais pode-se listar:

- Tempo de processamento: é um valor que vem disponível nos catálogos. No caso da SIEMENS, são informados os tempos necessários para o processamento de uma operação binária, operação aritmética e operações que usam variáveis que ocupam valores específicos de memória, como uma variável do tipo Real. Por exemplo, os dados do tempo de processamento para a CPU 1211C da família S7 1200 são (AG, 2021):
 - Operações envolvendo uma variável binária – 0,08 μ s por instrução;
 - Para operações que envolvam uma variável de 16 bits (Word) – 1,7 μ s por instrução;
 - Para operações com uma variável de 32 bits (em formato Real) – 2,3 μ s por instrução.
- Quantidade de memória e tipo de tecnologia: esse valor também pode ser encontrado em catálogos e geralmente está relacionado à quantidade de memória de programa (memória não volátil) e memória de dados (memória volátil). Ao longo do capítulo será abordado de forma mais detalhada esses tipos de memória.
- Quantidade de entradas e saídas: as CPUs podem já estar disponíveis com uma quantidade de entradas e saídas, alguns modelos permitem o aumento desse valor com a adição de módulos. No entanto, existe um número máximo de entradas e saídas que a CPU consegue processar, caso a necessidade de um determinado projeto ultrapassar esse número, faz-se necessário optar por outro modelo, talvez até mesmo a troca por outra família de CLPs dependendo do fabricante.

- Linguagem e recursos de programação disponíveis: designa quais linguagens a CPU suporta, tais como, ladder, diagrama de blocos funcionais (FCB) e texto estruturado.
- Portas de comunicação: alguns modelos de CPU já vêm com alguma porta para comunicação, como a porta RS-232 ou RS-422, mas caso exista a necessidade de comunicar usando uma rede que usa uma porta diferente, é necessário consultar o catálogo para verificar se existem módulos que permitem essa comunicação e se a CPU suporta a comunicação usando os protocolos atrelados a essas entradas.

PRINCÍPIO DE FUNCIONAMENTO

A função de um CLP é interagir com o ambiente no qual ele está inserido, coletando informações por meio dos dispositivos de entrada e/ou interferindo no ambiente gerando sinais elétricos que serão usados pelos dispositivos de saída. Essa interação com o mundo real acontece devido a uma sequência de comandos que são repetidos continuamente quando o controlador estiver em modo de execução.

Quando o CLP está no modo de execução, são iniciados alguns procedimentos que incluem testes de memória, verificação se o hardware declarado no ambiente de programação realmente condiz com o equipamento físico instalado, também são feitos testes que detectam erros no programa do usuário antes dele entrar em execução. Caso algum desses testes apontem algum tipo de inconsistência, por segurança, o CLP não entra em funcionamento, mas se não houver nenhum tipo de problema, então inicia-se um ciclo composto por três etapas que se repetem continuamente. Vale ressaltar que são permitidas alterações no programa enquanto o ciclo está em andamento. Essas etapas, mostradas na Figura 2.26, compreendem:

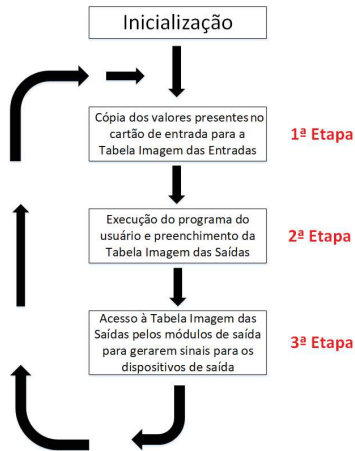


Figura 2.26: Representação esquemática de um ciclo completo de funcionamento.

Primeira etapa: o sistema operacional do CLP consulta a eletrônica dos cartões de entrada e mapeia essas entradas em posições de memória sem a necessidade de o operador interagir nesse processo. Nessa etapa, os estados dos dispositivos de entrada são copiados para a memória RAM em uma região chamada de TIE (Tabela Imagem das Entradas);

Segunda etapa: é executada as diversas instruções do programa do usuário. Algumas dessas instruções usam os valores das entradas, já armazenadas na memória na etapa anterior, outras instruções produzem valores que serão armazenados em outra região da memória chamada de TIS (Tabela Imagem das Saídas);

Terceira etapa: sem que haja a necessidade de qualquer intervenção do usuário, os valores armazenados na Tabela Imagem das Saídas são utilizados pelos cartões de saída para gerar sinais elétricos para os dispositivos de saída. Esses sinais serão responsáveis por acionar um motor, controlar a abertura de uma válvula, acionar uma sirene, avançar ou recuar um cilindro pneumático e etc.

Um ciclo completo (contendo as três etapas), em geral, tem duração entre 5ms e 50ms, sendo que esse tempo é chamado de “*Scan Time*”. Essa variação existe devido aos diversos fatores que influenciam no processamento, como:

- Quantidade do número de entradas e saídas que estão sendo usadas;
- Tamanho do programa do usuário e tipo das instruções que estão sendo utilizadas;
- Modelo da CPU (existem casos que é necessária a troca de família de um CLP devido a limitação do poder de processamento da CPU);

Para que os conceitos relacionados ao ciclo de funcionamento fiquem mais claros, pode-se imaginar a situação em que um usuário colocou um osciloscópio no terminal da saída digital atrelada à variável OUT1 de um CLP, transferindo também um programa desenvolvido de acordo com o pseudocódigo a seguir. Dessa forma, quando as variáveis de entrada *INP1* e *INP2* são levadas a nível lógico alto durante vários ciclos seguidos, o sinal mostrado pelo osciloscópio será igual ao que pode ser observado na Figura 2.27.

```
Trecho 1 (duração de 8ms)
```

```
Se (INP1=1) -> SET(OUT1)
```

```
Trecho 2 (duração de 4ms)
```

```
Se (INP2=1) -> RESET(OUT1)
```

```
Trecho 3 (duração 8ms)
```

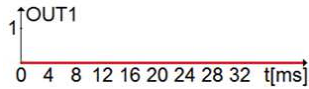


Figura 2.27: Sinal da saída $OUT1$, mostrado pelo osciloscópio.

Para entender a razão que leva a esse resultado, vamos analisar o que acontece em cada etapa do ciclo:

- Na primeira etapa, acontece apenas a cópia dos valores de entrada para a TIE (teríamos então que $INP1 = INP2 = 1$);
- Na segunda etapa, é executado todo o programa do usuário, usando os valores que estão na TIE. No primeiro momento, o valor de $OUT1$ a ser armazenado na TIS é igual a 1, mas logo após o Trecho 2, o valor é alterado para 0. Dessa forma, o valor de $OUT1$ enviado para a TIS no final dessa etapa é 0;
- Na terceira etapa, os valores armazenados na TIS são apenas usados para gerar o sinal elétrico que será aplicado aos dispositivos de saída. Assim, como o valor de $OUT1$ presente na TIS é 0, a saída terá o mesmo valor lógico.

2.3 DECLARAÇÃO DE VARIÁVEIS

TIPOS BÁSICOS DE MEMÓRIA

Assim como nos PCs convencionais, os CLPs fazem uso de dois tipos básicos de memória: não-volátil e volátil. A memória não volátil tem como característica a retentividade dos dados armazenados, assegurando a integridade dos mesmos durante a ausência de alimentação elétrica. Devido a essa característica, esse tipo de memória é usada, entre outras coisas, para o armazenamento do Sistema Operacional (OS, do termo em inglês, *Operational System*) do CLP (também conhecido com *firmware*) e do programa do usuário.

Por sua vez, a memória volátil, também conhecida como Memória de Acesso Aleatório (RAM, do termo em inglês *Random Access Memory*) não retém o conteúdo nela armazenado durante a ausência de alimentação elétrica (nos casos em que isso é desejado, o *hardware* deve incluir uma bateria de *backup*). No entanto, uma vantagem desse tipo de memória é a facilidade de gravação e de leitura dos dados, que acontece por meio de operações muito rápidas (atualmente, na ordem de nanosegundos). Devido a essa última característica, esse tipo de memória é usado para armazenar, por exemplo, valores relacionados às entradas e saídas do CLP, valores de contagem e temporização e muitos outros, que necessitam ser gravados e consultados muitas vezes ao longo da execução do programa do usuário.

Curiosidades sobre a memória não volátil

Desde que os primeiros CLPs foram criados, as memórias não voláteis passaram por um processo de evolução das tecnologias envolvidas, com os objetivos de facilitar os processos de programação e reprogramação, aumentar a velocidade de acesso, a capacidade de armazenamento e a durabilidade. A seguir, as principais tecnologias relacionadas a esse tipo de memória são descritas sucintamente, na ordem cronológica em que surgiram:

ROM (do termo em inglês, *Read Only Memory*): como o próprio nome já diz, é uma memória que, uma vez gravada, não permite alterações em seu conteúdo (seu uso passa a ser exclusivo para leitura). Dessa forma, os dados precisavam ser gravados pelo fabricante do *chip*, que precisavam ser especificamente encomendados para cada aplicação.

PROM (do termo em inglês, *Programmable Read Only Memory*): os *chips* que utilizam essa tecnologia também podem ser gravados apenas uma vez, porém os fabricantes podiam comercializá-los “em branco”, juntamente com um aparelho de gravação específico. Assim, seus clientes podiam evoluir o conteúdo a ser gravado de maneira mais flexível.

EPROM (do termo em inglês, *Erasable Programmable Read Only Memory*): utilizada primeiramente na década de 1970, trata-se de uma memória que pode ser apagada e reescrita várias vezes. No entanto, para a limpeza dos dados, é necessário expor sua escotilha (Figura 2.28) a uma fonte de luz ultravioleta por alguns minutos. Por se tratar de um processo demorado, no qual é necessário levar o *chip* para uma câmara de radiação ultravioleta, o uso dessa memória era incômodo. Além disso, para o processo de gravação, também é necessário retirar o *chip* da placa a qual se destina e inseri-lo em um gravador eletrônico.



Figura 2.28: Pastilha de memória EPROM, com sua escotilha ao centro, pela qual é possível expor seus componentes à luz ultravioleta e conseqüentemente apagar os dados armazenados.

EEPROM (do termo em inglês, *Electrically Erasable Programmable Read Only Memory*): esse tipo de memória solucionou o principal problema da EPROM, permitindo que os dados fossem apagados e reescritos de forma mais rápida, eletricamente. Assim, em relação à tecnologia anterior, a troca de conteúdo do *chip* passou a ser mais prática, dispensando-se a exposição à luz ultravioleta. No entanto, ainda era necessário retirar o *chip* do dispositivo a que se destina e levá-lo a um gravador externo.

FLASH EEPROM: trata-se da tecnologia mais utilizada na atualidade, inclusive nas famosas “*pen drives*” e telefones celulares. Apesar de ser não-volátil, apresenta quase as mesmas características de velocidade da memória RAM. Além disso, permite que as operações de escrita e reescrita sejam feitas por meio do mesmo dispositivo que faz uso de suas informações. Inicialmente, na informática industrial, seu uso era restrito aos CLPs de grande porte mas, com o passar do tempo, esta tecnologia passou a ser amplamente adotada por todos fabricantes para os mais diversos tipos de dispositivo.

Dessa forma, com a tecnologia atual, é muito comum que as ferramentas de configuração e programação de CLPs, além das funcionalidades discutidas anteriormente, permitam que o usuário atualize o OS de um determinado CLP sem se levantar da cadeira. Para isso, basicamente, basta encontrar uma nova versão de *firmware* no site do fabricante e, após verificar sua adequação, descarregá-lo na memória não volátil do controlador, similantemente ao que se faz no processo de *download* do *software* aplicativo.

Memória RAM

Por permitir operações de leitura e escrita na ordem de nanossegundos, a memória RAM é usada pelos OSs dos CLPs para armazenar dados que são constantemente consultados e alterados. Além disso, as variáveis criadas durante a elaboração do programa do usuário também fazem uso dessa memória.

Dessa forma, pode-se dizer que a memória RAM é segmentada em áreas, sendo que há um trecho de uso exclusivo do OS, um trecho exclusivo para o programa do usuário e trechos em que ambos interagem, ou seja, o OS disponibiliza dados para o programa do usuário e vice-versa. De forma mais específica, na área associada às entradas elétricas do CLP, o OS

coleta os dados dos respectivos módulos e os disponibiliza para que programa do usuário faça consultas; ao passo que, ao ser processado, esse último vai armazenando os resultados que produz na área de memória associada às saídas elétricas do CLP e, após o fim desta etapa, o OS coleta esses dados e atualiza os módulos de saída.

Devido a essas segmentações, geralmente cada trecho físico da memória RAM é tratado de forma distinta, recebendo uma identificação própria e com uma indicação de acesso que começa na posição zero. Tal distinção fica clara, principalmente, no momento de se fazer a declaração de variáveis do projeto (conforme mostrado nos próximos tópicos deste capítulo). No caso de vários modelos de CLP do fabricante Siemens, a área de uso exclusivo do OS não aparece para o programador, sendo que algumas das demais possuem as características mostradas na lista a seguir.

Área M (em referência a um espaço de armazenamento genérico, da palavra em inglês *Memory*): nessa região são armazenados os valores de uso exclusivo do programa do usuário, como os resultados e valores intermediários de manipulações algébricas e booleanas, sinais a serem recebidos e disponibilizados para o Sistema Supervisório e valores coletados nas saídas das funções de contagem e temporização;

Área I (em referência às entradas elétricas do CLP, da palavra em inglês *Input*): nessa região da memória o OS escreve os valores provenientes dos dispositivos de entrada, como botoeiras, chaves fim-de-curso, contatos auxiliares de disjuntores e contatores, potenciômetros e medidores de grandezas físicas;

Área Q (em referência às saídas elétricas do CLP, da palavra em inglês *Output*, sendo a letra *Q* escolhida no lugar da letra *O* para facilitar a distinção em relação ao algarismo 0): nessa região de memória o programa do usuário escreve os valores destinados aos dispositivos de saída, como lâmpadas, sirenes, bobinas de contatores e referências para controle de grandezas como velocidade e posição.

Neste ponto, é importante esclarecer que o tamanho dessas áreas varia de acordo como o modelo de cada CPU. Em algumas delas, mais elaboradas, é possível reduzir uma determinada área para se ganhar espaço em outra (isso é feito com a plataforma de programação). Além disso, deve-se destacar que outras áreas também existem. Para se ter um bom exemplo, é preciso notar que nas áreas *M*, *I* e *Q* são declaradas as **variáveis globais** de um projeto (portanto, acessíveis a partir de qualquer bloco de programação). No entanto, existem ainda as **variáveis locais** das funções destinadas a encapsulamentos de código, cujos valores também são armazenados em um trecho específico da memória RAM, omitido da lista anterior. Por fim, ressalta-se que outros fabricantes adotam nomenclaturas diferentes para as áreas estudadas, como *X* no lugar de *I* e *Y* em equivalência à *Q*.

DECLARAÇÃO DE VARIÁVEIS (CONCEITO GERAL)

Quando se faz um programa em um PC convencional, usando, por exemplo, a linguagem C, a declaração de uma variável é feita, basicamente, escolhendo-se o nome, definindo-se o tipo de dado para essa variável e, de acordo com as boas práticas de programação, registrando-se um comentário pertinente para ela. Certamente todas as variáveis ocuparão espaços na memória RAM, mas a alocação de cada uma é feita automaticamente pelo sistema.

No caso dos CLPs, a declaração de variáveis se baseia nos mesmos princípios, ou seja, deve-se escolher o nome, definir o tipo de dado para cada uma delas e registrar respectivos os comentários, sendo que alguns exemplos podem ser observados na Tabela 2.3. No entanto, neste momento, cabe uma pergunta fundamental: em qual área da memória RAM cada variável será alocada (M , I , Q ou outra)? Isso precisará ser definido pelo desenvolvedor do projeto, conforme mostrado a seguir.

Tabela 2.3: Exemplo de declaração de variáveis utilizando a nomenclatura prevista em alguns CLPs Siemens para seus principais tipos de dado, bem como comentários figurativos sobre o intervalo de valores que cada um pode armazenar.

Símbolo	Tipo de dado	Comentário
Ligado	Bool	binário (0 ou 1)
Position	SInt	8 bits c/ sinal (-128 a 127)
Secs	USInt	8 bits s/ sinal (0 a 255)
Saldo	Int	16 bits c/ sinal (-32.768 a 32.767)
Horas	UInt	16 bits s/ sinal (0 a 65.535)
RawVal	DInt	32 bits c/ sinal (-2.147.483.648 a 2.147.483.647)
Pulsos	UDInt	32 bits s/ sinal (0 a 4.294.967.295)
Pressure	Real	32 bits (-3,402823 · 10 ³⁸ a -1,175495 · 10 ⁻³⁸ e também +1,175495 · 10 ⁻³⁸ a +3,402823 · 10 ³⁸)

Como visto, quando se trata de CLPs (de qualquer modelo), ao se declarar uma variável, espera-se que, no mínimo, quatro definições sejam feitas (algumas delas exigidas pelo sistema e outras indicadas pelas boas práticas de programação): 1) nome; 2) tipo de dado; 3) comentário; 4) área da memória. As três primeiras coincidem com as premissas da informática convencional e a última é pertinente apenas aos CLPs pois, de alguma forma, o programador deve definir se a variável está associada às suas entradas e saídas elétricas ou não, sendo que a forma como essa última definição é feita varia de fabricante para fabricante e também entre as famílias de CPUs que eles disponibilizam no mercado.

Em alguns casos, a forma de se associar uma variável a uma das entradas ou saídas do CLP (ou não) é feita por meio de diferentes tabelas de declaração dentro da ferramenta de programação. Isto é, nesses casos, há no mínimo um local para se declarar as variáveis de entrada, um local para declaração das variáveis de saída e outro para as que não serão associadas aos terminais elétricos do dispositivo.

Um exemplo pode ser observado na Figura 2.29, no qual a programação de um CLP da família CompactLogix (Rockwell Automation) ocorre por meio da plataforma RSLogix 5000. Entre os módulos que compõem o CLP em questão, o que ocupa a segunda posição do *rack* é um módulo misto, com seis entradas e quatro saídas digitais. Na tabela que se vê no lado esquerdo da figura, as variáveis pertinentes a esse módulo podem ser declaradas, sendo que as primeiras linhas são reservadas para as entradas e as últimas para as saídas (escolhendo-se a linha da declaração, escolhe-se a área de memória da variável). Além disso, o conteúdo das células na cor cinza não pode ser alterado, indicando que o sistema define automaticamente o nome e tipo de dado para essas variáveis (cabendo ao programador apenas fazer os comentários). Por sua vez, na tabela que se vê no lado direito da figura, faz-se a declaração das variáveis que serão alocadas no segmento da memória RAM que não possui relação com os pontos de entrada e saída do CLP, sendo que para cada uma, o programador deve definir o nome, tipo de dado e comentário.

Name	Data Type	Description	Name	Data Type	Description
+ Local:2:C	AB:1769_IQ6*DIW4:C:0		Aku_SimTmp_TpAqRs	REAL	Tempo de aquecimento
- Local:2:I	AB:1769_IQ6*DIW4:I:0		Aku_SimTmp_ValCel	REAL	Valor em graus Celsius
+ Local:2:I:Fault	DINT		Ana_MIT01_CmRaw	INT	Corrente bruta
- Local:2:I:Data	SINT		Alr_MIT01_CmExd	BOOL	Limite de corrente excedido
- Local:2:I:Data.0	BOOL	DIn_Reserva017	Alr_MIT01_DrvFal	BOOL	Falha no acionamento
- Local:2:I:Data.1	BOOL	DIn_Reserva018	Alr_MIT01_TmpExd	BOOL	Temperatura excedida
- Local:2:I:Data.2	BOOL	DIn_Reserva019	Alr_MIT01_TppExd	BOOL	Tempo de partida excedido
- Local:2:I:Data.3	BOOL	DIn_Reserva020	+ Ana_GERAL_AlvCPU	INT	Sinalização de atividade da CPU
- Local:2:I:Data.4	BOOL	DIn_Reserva021	+ Ana_MIT01_VeEst	INT	Velocidade estimada
- Local:2:I:Data.5	BOOL	DIn_Reserva022			
- Local:2:I:Data.6	BOOL				
- Local:2:I:Data.7	BOOL				
+ Local:2:I:ReadBack	SINT				
- Local:2:O	AB:1769_IQ6*DIW4:O:0				
- Local:2:O:Data	SINT				
- Local:2:O:Data.0	BOOL	DOu_LIIMUX_M*Q:TNR			
- Local:2:O:Data.1	BOOL	DOu_LIIMUX_M*Q:TRV			
- Local:2:O:Data.2	BOOL	DOu_LIIMUX_AQK:TLG			
- Local:2:O:Data.3	BOOL	DOu_LIIMUX_VLKTAB			
- Local:2:O:Data.4	BOOL				
- Local:2:O:Data.5	BOOL				
- Local:2:O:Data.6	BOOL				
- Local:2:O:Data.7	BOOL				
+ Local:3:C	AB:1769_DO16:C:0				

Figura 2.29: Duas tabelas para configuração de um CLP da família CompactLogix por meio da ferramenta RSLogix 5000: na esquerda, encontram-se os locais de declaração das variáveis referentes às seis entradas e quatro saídas digitais presentes em um módulo misto (o sistema reserva um *byte* para cada um dos locais de declaração, por isso há dois espaços vazios no primeiro e quatro espaços vazios no segundo); na direita, algumas declarações no local previsto para as variáveis que não possuem relação com as entradas e saída elétricas.

Outros fabricantes, no entanto, adotam outra filosofia, permitindo que, caso desejado, todas as variáveis do projeto sejam declaradas em uma mesma tabela, na qual as informações básicas para cada variável são: nome, tipo de dado, **endereço** e comentário. De forma que o **endereço** da variável contém, entre outras, a informação que define a qual área de memória do CLP ela pertence. Um dos fabricantes que adota esta filosofia de declaração é a Siemens, sendo que os detalhes pertinentes ao sistema de endereçamento utilizado em seus CLPs serão discutidos no tópico a seguir.

DECLARAÇÃO DE VARIÁVEIS (ENDEREÇAMENTO SIEMENS)

Conforme mencionado anteriormente, a Siemens adota um sistema de endereçamento de variáveis para seus CLPs que, entre outras coisas, permite definir a qual área da memória RAM (M , I , Q ou outra) cada uma pertence. Conforme apresentado nos tópicos a seguir, há uma distinção entre a forma de endereçamento das variáveis binárias e das variáveis que armazenam valores decimais.

Endereçamento de variáveis binárias

No caso das variáveis binárias, os endereços devem definir apenas a área de memória e a **posição** em que uma determinada informação será alocada dentro dessa área. Para se definir essa **posição**, a Siemens adota um sistema de numeração de *bytes* e *bits*, sendo que ambos começam em zero. Dessa forma, qualquer variável binária pode ser endereçada conforme a regra ilustrada na Figura 2.30, em que o primeiro caractere é a letra que define a área da memória RAM, o segundo é número do *byte* escolhido, o terceiro é apenas um separador e o último é o número do *bit* de interesse. Em complemento, alguns exemplos de declarações de variáveis binárias podem ser observados na Tabela 2.4.

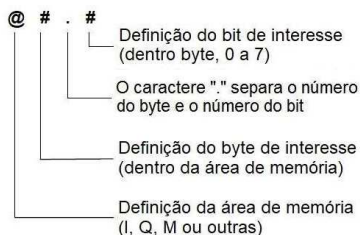


Figura 2.30: Resumo da regra para formação de endereços de variáveis binárias nos CLPs fabricados pela Siemens.

Tabela 2.4: Exemplos de declarações de variáveis binárias: uma na área de memória do usuário, uma na área relacionada às entradas e a última na área relacionada às saídas.

Símbolo	Tipo de dado	Endereço	Comentário
EQP088_AuxLoc	Bool	M102.5	Auxiliar modo local
EQP004_BotLig	Bool	I12.3	Botão liga
EQP088_Abre	Bool	Q27.1	Abre (energia)

No caso das variáveis relacionadas às entradas e saídas do CLP, uma questão ainda deve ser esclarecida: como se estabelece a relação entre um determinado terminal elétrico e seu respectivo endereço? Isso é feito durante a configuração de *hardware* do dispositivo, sendo que um endereço inicial é atribuído a cada módulo de entradas e saídas.

Assim, considerando as declarações que aparecem na tabela anterior, se um módulo de 16 entradas digitais receber como endereço inicial o *byte* 12 da área *I*, o endereço *I12.3* estará associado ao quarto terminal elétrico desse módulo e, após a declaração de variáveis, também poderá ser acessado por meio do nome "EQP004_BotLig". Similarmente, se um módulo de saídas digitais for alocado a partir do *byte* 26 da área *Q*, o endereço *Q27.1* estará associado ao décimo terminal desse módulo e também poderá ser acessado pelo nome "EQP088_Abre", assim que a declaração de variáveis for realizada.

Endereçamento de variáveis decimais

Por sua vez, os endereços de variáveis que armazenam números decimais devem definir a área de memória, a posição em que um determinado valor numérico será alocado dentro desta área e a quantidade de *bytes* necessários para armazená-lo. Portanto, as variáveis decimais podem ser endereçadas conforme mostrado na Figura 2.31, sendo que o primeiro caracter é a letra que define a área da memória RAM, o segundo caracter é a letra que define a quantidade de *bytes* abrangida pelo endereço e o último é o número do *byte* inicial. Alguns exemplos de declarações podem ser observados na Tabela 2.5.

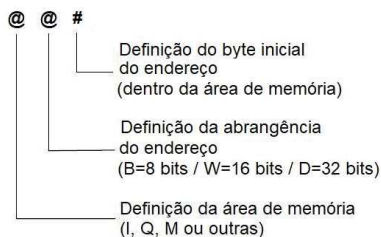


Figura 2.31: Resumo da regra para formação de endereços de variáveis que armazenam números decimais nos CLPs fabricados pela Siemens.

Tabela 2.5: Exemplos de declarações de variáveis decimais: duas na área de memória do usuário, duas na área relacionada às entradas e duas na área relacionada às saídas.

Símbolo	Tipo de dado	Endereço	Comentário
EQP087_VelEng	Real	MD72	Referência velocidade (%)
AUX005_ItmCnt	USInt	MB76	Itens contabilizados
MED022_NivRaw	UInt	IW64	Nível de água (bruto)
MED023_PrsRaw	UInt	IW66	Pressão do óleo (bruto)
EQP087_VelRaw	UInt	QW130	Referência velocidade (bruto)
EQP088_PosRaw	UInt	QW132	Referência posição (bruto)

Quando se deseja alocar variáveis em sequência (sem deixar espaços vazios em uma determinada área de memória), ao se fazer uma declaração, deve-se observar qual é o *byte* inicial da variável anterior e também sua abrangência, ou seja, quantos *bytes* ela consome. Considerando as declarações que aparecem na tabela anterior, pode-se observar que a variável “EQP087_VelEng” possui uma abrangência de quatro *bytes* (definida pela letra “D” em seu endereço), sendo que o primeiro ocupa a posição 72 da área de memória do usuário. Dessa forma, o próximo *byte* disponível desta área está na posição 76, justamente onde foi alocada a variável “AUX005_ItmCnt”, que possui uma abrangência unitária (definida pela letra “B” em seu endereço).

Similarmente ao que foi discutido para as entradas e saídas digitais, a associação entre os terminais elétricos das entradas e saídas analógicas do CLP e seus respectivos endereços também acontece durante a configuração de *hardware* do dispositivo, sendo que um endereço inicial é atribuído a cada módulo.

Ainda considerando os exemplos de declaração da tabela anterior, se um módulo de cinco entradas analógicas receber como endereço inicial o *byte* 64 da área *I*, o endereço *IW*64 estará associado à primeira entrada física desse módulo e, após a declaração de variáveis, também poderá ser acessada por meio do nome “MED022_NivRaw”. Isso acontece porque, independente da quantidade de *bits* utilizados pelos conversores ADC de cada módulo (em geral, a resolução está entre 10 e 16 *bits*), a Siemens reserva 16 *bits* para cada um, sendo que a letra “W” sempre estará presente em seus endereços. Dessa forma, como os *bytes* 64 e 65 foram abrangidos pela primeira entrada física do módulo, a segunda recebe o endereço *IW*66 e também poderá ser acessada pelo nome “MED023_PrsRaw”, assim que a declaração de variáveis for realizada.

Expandindo o raciocínio, conclui-se que a última das cinco entradas do suposto módulo terá o endereço *IW*72 e que a próxima posição de memória disponível na área *I* é o *byte* 74, que eventualmente poderá ser o endereço inicial de outro módulo. Analogamente, o mesmo acontece para as saídas analógicas.

Sobreposição de endereços

Como mencionado nas seções anteriores, é possível trabalhar com variáveis que ocupam diferentes espaços de memória. Como o tamanho desses espaços são considerados no momento de fazer o endereçamento, é importante averiguar se variáveis diferentes não ocupam um espaço em comum na memória RAM, evitando assim o problema de sobreposição de endereços.

A Figura 2.32 traz uma análise bit a bit do endereçamento de três entradas analógicas que supostamente foram alocadas nas primeiras posições da área *I*.

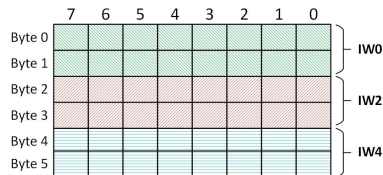


Figura 2.32: Endereçamento de variáveis (análise bit a bit).

Como cada endereço abrange dois *bytes*, se uma variável for declarada no endereço *IW*0 e a outra no endereço *IW*1, haverá conflito de dados, pois elas passam a compartilhar o *byte* que está na posição 1 da mesma área de memória. Isso implica que a mudança de valor em uma delas irá interferir na outra, o que pode ocasionar efeitos catastróficos no programa e no funcionamento do CLP como um todo.

Como o endereçamento de uma variável considera diferentes áreas da memória RAM, sendo que cada uma tem seu próprio mapeamento, endereços definidos como *IW*0, *QW*0 e *MW*0 não são conflitantes entre si, pois se encontram em regiões diferentes.

Algumas plataformas impedem a sobreposição de endereços durante a compilação do projeto. Outras permitem que isso aconteça, gerando uma mensagem de aviso para o usuário e permitindo que, nos raros casos em que a sobreposição é intencional, o procedimento de *download* ocorra assim mesmo.

Nas Figuras 2.33 e 2.34 pode-se observar outras situações em que o problema de sobreposição acontece: no primeiro caso, há sobreposição de duas variáveis decimais que supostamente foram declaradas com os endereços *MD1* e *MW4*; no segundo, mostra-se a sobreposição no caso em que uma variável binária e outra decimal usam, respectivamente, os endereços *Q3.4* e *QB3*;

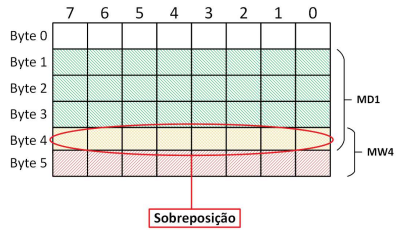


Figura 2.33: Exemplo de sobreposição de endereços de duas variáveis decimais.

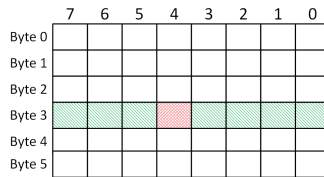


Figura 2.34: Exemplo de sobreposição de uma variável binária e uma variável decimal.

2.4 ROTEIRO PARA AULA PRÁTICA

OBJETIVO

Praticar tarefas fundamentais disponíveis em uma plataforma de programação de CLPs, como: criação de projeto, configuração de *hardware*, alteração do estado da CPU, realização de *download*, declaração de variáveis, operações de leitura e escrita na memória, verificação do tempo de ciclo e *backup* de projeto.

MATERIAL NECESSÁRIO

- Bancada de trabalho com CLP e PC (com plataforma de programação instalada);
- Módulo com elementos elétricos de interface (projeto disponível no Apêndice B.3);
- Módulo para simulação de processo produtivo (projeto disponível no Apêndice B.4).

EXPERIMENTO 1

Como participante deste curso, este será seu primeiro contato com o CLP a ser utilizado ao longo das demais semanas. As tarefas solicitadas a seguir são fundamentais e serão necessárias em diversas outras oportunidades. Por isso, realize-as com bastante atenção e lembre-se de retornar a este ponto em caso de dúvidas.

TAREFA 1 – Abra a plataforma de programação de CLPs adotada neste curso.

Como?



TAREFA 2 – Crie um projeto e salve-o na pasta que você usará ao longo do semestre.

Como?



TAREFA 3 – Adicione o CLP que você usará ao longo do semestre ao projeto que você acabou de criar.

Como?



TAREFA 4 – Configure o *hardware* do CLP que você acabou de adicionar.

Como?



TAREFA 5 – Faça o *download* das configurações de *hardware* realizadas na tarefa anterior.

Como?



TAREFA 6 – Faça alterações no estado da CPU (transições $STOP \rightleftharpoons RUN$), observe as respectivas sinalizações (troca de cor de um dos *LEDs* da CPU) e, ao final, deixe o ciclo da CPU ativado (modo *RUN*). Além disso, verifique o tempo de ciclo da CPU (mínimo, máximo e último valor).



EXPERIMENTO 2

Outra tarefa fundamental relacionada à programação de CLPs é a declaração de variáveis (ou a criação de *tags*, se preferir). Em seu programa, você até pode usar endereços de memória de forma absoluta (por exemplo: *I0.0*, *IW128*, *Q5.4*, *MD500*), mas isso não é uma boa prática pois, a medida em que seu projeto vai se tornando mais elaborado, a falta de nomes adequados e comentários vai lhe deixando mais perdido. Dessa forma, antes de usar qualquer endereço de memória, o ideal é fazer o cadastro de uma variável, atribuindo a este endereço, pelo menos: um nome único, um comentário e a definição do tipo de dado que ele armazenará. As tarefas a seguir tratam desse assunto, ou seja, mostram como se faz o cadastro de variáveis utilizando a plataforma de programação adotada neste curso.

TAREFA 1 – Encontre, dentro da sua plataforma de programação, o módulo com o qual se faz o cadastro de variáveis. Neste primeiro momento, com o objetivo de organizar seu projeto, crie todos os *grupos de variáveis* indicados na Tabela 2.6.



Tabela 2.6: Relação de todos os *grupos de variáveis* utilizados no projeto a ser desenvolvido ao longo deste curso.

Nome do Grupo de variáveis
01 – Entradas Digitais
02 – Saídas Digitais
03 – Entradas Analógicas
04 – Saídas Analógicas
05 – Comandos
06 – Estados
07 – Alarmes
08 – Auxiliares Digitais
09 – Auxiliares Analógicos
10 – Grandezas Analógicas
11 – Parâmetros

TAREFA 2 – Faça o cadastro de todos os *tags* relacionados às conexões elétricas do CLP, ou seja, preencha os quatro primeiros *grupos de variáveis* (Entradas Digitais, Saídas Digitais, Entradas Analógicas e Saídas Analógicas) de acordo com as informações contidas no Projeto Elétrico do sistema (o referido documento encontra-se em um apêndice do presente material, na Seção A.2).



TAREFA 3 – Faça um novo *download* do projeto, que agora contém as variáveis cadastradas na tarefa anterior.



EXPERIMENTO 3

Por fim, mas não menos importante do que os primeiros experimentos, vamos explorar a funcionalidade que nos permite monitorar valores na memória do CLP (por meio de operações de leitura) e também alterá-los (por meio de operações de escrita). Para tal, proceda conforme solicitado na lista de tarefas a seguir.

TAREFA 1 – Encontre, dentro da sua plataforma de programação, o módulo que torna possível a **monitoração e a alteração de valores na memória do CLP**. Em seguida, crie uma tabela para acessar todos os *tags* relacionados às conexões elétricas do CLP, identificando-a com o nome “**Teste de I/O**”.

Como?



TAREFA 2 – Preencha a tabela criada na última tarefa com todos os *tags* relacionados às conexões elétricas do CLP. **Dica:** você pode voltar aos *grupos de variáveis* construídos no experimento anterior, copiar as colunas que contêm os nomes dos *tags* e colar tais informações na tabela em questão.

Como?



TAREFA 3 – Cumpra os procedimentos indicados a lista a seguir, realizando as operações leitura e/ou escrita de valores na memória do CLP solicitadas em cada um deles.

Como?



- Ative o recurso de monitoração cíclica das posições de memória cadastradas na tabela e, por meio das botoeiras presentes no módulo de interfaces elétricas conectado ao seu CLP, altere o nível lógico das 8 primeiras entradas digitais;
- Altere (mais de uma vez, se quiser) o nível lógico das saídas digitais associadas ao **Mixer** e observe o efeito desta ação em sua planta didática (termine este procedimento deixando-o desligado);
- Ainda com o recurso de monitoração cíclica das posições de memória ativado, escreva o valor 15000 na posição de memória associada à referência de velocidade da **Bomba de água** e observe o efeito nas posições de memória relacionadas à medição do nível de água no tanque:
 - A medida que a indicação do nível de água fornecida pelo medidor LT101 aumenta, as chaves de detecção binária desta mesma grandeza sofrem alteração de nível lógico: LSL101 quando o valor está próximo a 11000 e LSH101 quando está por volta de 16500.
 - Quando o valor fornecido pelo medidor LT101 estiver próximo a 18000, desligue a bomba escrevendo 0 em sua referência de velocidade.
- Ligue e desligue o **Aquecedor** alterando o nível lógico da respectiva saída digital e observe: a entrada digital associada ao retroaviso que indica se ele está ligado ou não; a entrada analógica, que indica a temperatura da água do tanque (termine este procedimento deixando o atuador desligado);
- Ative a saída digital apropriada para abrir a **Válvula de dreno direto** (VD101), esgotando a água do tanque de sua planta didática (ao final, feche a válvula).

FINALIZANDO

TAREFA 1 – Se sua plataforma de desenvolvimento estiver *online* com seu CLP, faça a desconexão. Em seguida, salve e faça *backup* do seu projeto.

Como?



TAREFA 2 – Se estiver tudo *ok* e não houver dúvidas em relação ao que foi desenvolvido, feche todas as ferramentas que estiverem abertas em seu PC e desligue-o. Desligue também todos os módulos didáticos da sua bancada de trabalho e, ao sair, deixe o local limpo e organizado.

2.5 TAREFAS PARA A PRÓXIMA SEMANA

OBJETIVO

Cadastrar as variáveis pertencentes aos grupos de sinais utilizados para que o CLP possa, entre outras coisas, receber e fornecer dados ao SSC.

MATERIAL NECESSÁRIO

- Bancada de trabalho com CLP e PC (com plataforma de programação instalada).

RETOMANDO

TAREFA 1 – Abra a plataforma de programação de CLPs adotada neste curso e, em seguida, localize e abra seu projeto (espera-se que nele estejam incorporadas todas as configurações realizadas na última aula prática).

Como?



TAREFA 2 – De acordo com as informações contidas no **Mapa de Memória do CLP** (veja a Seção A.3, no apêndice do presente material), faça o cadastro de todos os *tags* pertencentes aos cinco *grupos de variáveis* projetados para que, entre outras coisas, o CLP possa receber e fornecer dados ao SSC, sendo eles: Comandos, Estados, Alarmes, Grandezas analógicas e Parâmetros.

Como?



FINALIZANDO

TAREFA 1 – Se sua plataforma de desenvolvimento estiver *online* com seu CLP, faça a desconexão. Em seguida, salve e faça *backup* do seu projeto.

Como?



TAREFA 2 – Quando encerrar suas atividades, feche todas as ferramentas que estiverem abertas em seu PC e desligue-o. Certifique-se de que todos os módulos didáticos da sua bancada de trabalho também estejam desligados e, ao sair, deixe o local limpo e organizado.

2.6 EXERCÍCIOS

QUESTÃO 1 – Sabendo que a grande maioria dos CLPs se baseia no conceito de *hardware* modular, cite e explique a finalidade de cinco tipos de módulo.

QUESTÃO 2 – Explique sucintamente cada etapa do ciclo de funcionamento de um CLP.

QUESTÃO 3 – As características de *hardware* dos CLPs estão bastante relacionadas aos seus fabricantes. Ou seja, cada fabricante fornece um ou mais modelos de CLPs com características de *hardware* particulares. Sendo assim, é normal que para cada “família” de CLPs exista uma ferramenta de programação. Porém, existem pontos em comum. Por exemplo, o sistema operacional Windows é utilizado como plataforma para a maioria dessas ferramentas de programação. Dessa forma, o desenvolvimento de programas para CLPs pode ser realizado por meio dos recursos intuitivos desse sistema operacional. Além de facilitar essa tarefa, as diversas ferramentas de programação disponíveis no mercado possuem outras *funcionalidades* em comum, cite 3 delas.

QUESTÃO 4 – Na maioria dos casos, os componentes de *hardware* de um CLP precisam ser declarados de forma apropriada para que a CPU e o programador possam fazer uso dos mesmos. Por exemplo, os cartões de entradas digitais precisam ser declarados e associados a posições específicas da memória dos CLPs, estabelecendo relações do tipo: a primeira entrada do cartão 1 corresponde a posição de memória I0.0, a oitava entrada do cartão 5 corresponde a posição de memória I4.7 e assim por diante. Sendo assim, para alguns modelos de CLP, a inclusão de novos componentes de *hardware* pode causar alguns transtornos caso o CLP esteja em operação (controlando um processo). Explique por que isso pode ocorrer.

QUESTÃO 5 – Com relação ao CLP S7-1200, complete a Tabela de endereçamento 2.7.

Tabela 2.7: Exemplo de quadro para preenchimento dos endereços solicitados no problema em questão.

Endereço	Comentário
	Décima quarta entrada digital (supondo que a primeira é I0.0)
	Vigésima nona saída digital (supondo que a primeira é Q0.0)
	Qualquer entrada analógica
	Próxima entrada analógica após a do item anterior
	Qualquer variável do tipo real na área de memória do usuário
	Próxima variável do tipo real após a do item anterior

QUESTÃO 6 – Considere um processo de produção automatizado com dois Controladores Lógico-Programáveis. O CLP01 controla os equipamentos EQP01 à EQP04 e o CLP02 controla os equipamentos EQP05 à EQP08. Suponha que, inicialmente, estes CLPs eram independentes um do outro, pois todos os equipamentos do processo funcionavam separadamente. Posteriormente, descobriu-se que o processo teria um rendimento maior caso o EQP04 funcionasse apenas quando o EQP05 estivesse ligado. Ou seja, descobriu-se que o sinal “EQP05 ligado” deveria ser enviado do CLP02 para o CLP01. Descreva em um parágrafo uma possível solução para tal situação, considerando as seguintes restrições:

-
- Os CLPs são de fabricantes diferentes, assim não podem se comunicar por meio de rede. Além disso, o envio deste único sinal *não* justifica financeiramente a instalação de módulos de comunicação ou *gateways* que viabilizem a troca de dados via rede.
 - O processo em questão *não* dispõe de um Sistema de Supervisão e Controle, apenas pequenos painéis de operação, independentes para cada CLP;
 - *Não* é viável redistribuir os equipamentos entre os CLPs. Ou seja, o CLP01 deve continuar controlando os equipamentos EQP01 à EQP04 e o CLP02 deve continuar controlando os equipamentos EQP05 à EQP08;
 - Existem entradas e saídas digitais disponíveis em ambos os CLPs. No entanto, a distância entre o CLP01 e o EQP05 torna a ligação elétrica entre os mesmos inviável.

2.7 REFERÊNCIAS

- AG, Siemens (Ed.). **How do you connect a PT100 sensor to the LOGO! analog expansion module AM2 PT100 or AM2 RTD?** 2016. <https://support.industry.siemens.com/cs/document/23671451/how-do-you-connect-a-pt100-sensor-to-the-logo!-analog-expansion-module-am2-pt100-or-am2-rtd?dti=0&d1=en&lc=pt-BR>. [Online; accessed 07-fevereiro-2022].
- AG, Siemens (Ed.). **Products for Totally Integrated Automation**. Catalog ST 70. Germany, Nürnberg, 2021. Disponível em: <https://cache.industry.siemens.com/d1/files/167/109744167/att_1076941/v1/simatic-st70-complete-english-2021.pdf>.
- GEMS SENSORS AND CONTROLS (Ed.). **PULSE WIDTH MODULATION (PWM)**. 2021. [https://www.gemssensors.com/blog/blog-details/pulse-width-modulation-\(pwm\)](https://www.gemssensors.com/blog/blog-details/pulse-width-modulation-(pwm)). [Online; accessed 08-fevereiro-2022].
- GEORGINI, J.M. **Automação Aplicada: Descrição e implementação de Sistemas Sequenciais com PLCs**. 9. ed.: Saraiva Educação S.A., 2018. 166 p. ISBN 9780135048818.
- SILVEIRA, P.R. da; SANTOS, W.E. dos. **Automação e Controle Distreto**. Érica, 1998. 229 p. ISBN 9788571945913.