

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Guilherme Henrique Silveira Costa

Utilização de um Controlador Lógico Programável em um Sistema de Bombeamento
de Água

Uberlândia

2022

Guilherme Henrique Silveira Costa

Utilização de um Controlador Lógico Programável em um Sistema de Bombeamento
de Água

Trabalho de Conclusão de Curso
apresentado à Faculdade de Engenharia
Elétrica da Universidade Federal de
Uberlândia como parte dos requisitos para a
obtenção do título de bacharel em
Engenharia Elétrica.

Orientador: Prof. Dr. Luciano Coutinho
Gomes

Uberlândia
2022

Utilização de um Controlador Lógico Programável em um Sistema de Bombeamento de Água

Trabalho de Conclusão de Curso aprovado
pela banca examinadora da Faculdade de
Engenharia Elétrica formada por:

Uberlândia, 12 de agosto de 2022.

Prof. Dr. Luciano Coutinho Gomes, UFU/MG

Prof. Dr. Augusto Wohlgemuth Fleury Veloso da Silveira, UFU/MG

Prof. Dr. Carlos Eduardo Tavares, UFU/MG

SUMÁRIO

1 INTRODUÇÃO	12
2 ASPECTOS HISTÓRICOS	14
3 REVISÃO SOBRE ACIONAMENTOS ELÉTRICOS.....	16
3.1 Motores de Indução Trifásicos.....	16
3.2 Dispositivos de Proteção e Comando	20
3.3 Chave de Partida	22
4 FUNDAMENTAÇÃO TEÓRICA SOBRE CLP	26
4.1 Definição, Princípios de Funcionamento e Constituição.....	26
4.2 Formas Básicas	29
4.3 Unidade de Processamento Central	32
4.4 Entradas e Saídas	40
4.5 Dispositivos de Programação	47
4.6 Modo de Operação	49
4.7 Programação	53
4.8 Linguagem Ladder	54
4.9 Sistemas Combinacionais	72
4.10 Noções de Redes de Comunicação	80
5 APLICAÇÃO	86
5.1 Sistema de Bombeamento.....	86
5.2 Programação em Ladder	88
6 CONSIDERAÇÕES FINAIS	92
7 REFERÊNCIAS.....	93
Apêndice A – Reabastecimento do tanque em condições normais.	97
Apêndice B – Alarme de falha no reabastecimento do tanque.	98
Apêndice C – Reset do contador de falhas de reabastecimento do tanque.	99
Apêndice D – Partida do motor em estrela.	100

Apêndice E – Comutação do motor para triângulo.	102
Apêndice F – Impossibilidade de partida devido ao baixo nível do tanque.....	104
Apêndice G – Alarme e proteção contra operação a seco.	105
Apêndice H – <i>Reset</i> da falha decorrente da operação a seco.....	106

LISTA DE ILUSTRAÇÕES

Figura 1 – Evolução dos sistemas de controle.	15
Figura 2 – Motor de indução trifásico.	16
Figura 3 – Rotor gaiola com e sem laminas de ferro (à esq. e à dir.).....	17
Figura 4 – Corrente de partida.	18
Figura 5 – Curva típica de conjugado do MIT.....	18
Figura 6 – Conjugado acelerante.	19
Figura 7 – Ligações delta (à esq.) e estrela (à dir.).	20
Figura 8 – Simbologia dos contadores.	22
Figura 9 – Simbologia de um disjuntor com disparadores térmico e magnético.....	22
Figura 10 – Circuito de potência chave estrela-triângulo.	24
Figura 11 – Curvas de corrente e conjugado em função da velocidade.....	25
Figura 12 – Funcionamento de um CLP.....	27
Figura 13 – Diagrama de blocos de um CLP.....	28
Figura 14 – CLP Compacto Siemens LOGO! 230RC.....	30
Figura 15 – Arquitetura de um CLP modular.....	31
Figura 16 – Siemens SIMATIC S7-300.	31
Figura 17 – Constituição da CPU.....	32
Figura 18 – Bateria de <i>Backup</i> 3.6V 0,85Ah de Lítio para CLP Siemens.....	34
Figura 19 – Memória EPROM.	35
Figura 20 – Unidades de memória.	36
Figura 21 – Partes da memória de um CLP.	38
Figura 22 – Fonte Siemens Simatic PM1207.	39
Figura 23 – Sinais digital e analógico.....	40
Figura 24 – Optoacoplador.....	41
Figura 25 – Interface de entrada de um sinal digital.....	42
Figura 26 – Tipos de saídas digitais a) relé b) transistor e c) TRIAC.	43
Figura 27 – Conversor A/D.....	44
Figura 28 – Sinais digitalizados.....	44
Figura 29 – Multiplexador conectado ao conversor A/D.....	45
Figura 30 – Conversor D/A.....	46
Figura 31 – Demultiplexador conectado ao conversor D/A.	46
Figura 32 – Programador portátil.....	47

Figura 33 – Utilização de um PC como dispositivo de programação.	48
Figura 34 – IHM Schneider HMIST6200 para CLP.	48
Figura 35 – Fluxo de dados durante a varredura.	50
Figura 36 – Fluxograma do modo de execução do CLP.	51
Figura 37 – Indicadores do CLP Rockwell Micro820.....	52
Figura 38 – Estrutura do código <i>ladder</i>	56
Figura 39 – Caminhos possíveis de continuidade.....	56
Figura 40 – Endereçamento de memória no Siemens S7-200.....	57
Figura 41 – Endereçamento de memória na família de CLPs Rockwell SLC.....	58
Figura 42 – Contato e bobina virtuais.....	58
Figura 43 – Etapas de funcionamento com linguagem <i>ladder</i>	59
Figura 44 – Ordem de varredura dos degraus.	60
Figura 45 – Ordem de varredura em cada degrau.	60
Figura 46 – Instante de acionamento das bobinas.....	61
Figura 47 – Correção do fluxo reverso com repetição de contatos.	61
Figura 48 – Diagrama <i>ladder</i> com contato de selo.....	62
Figura 49 – Diagrama elétrico simples e seu equivalente em <i>ladder</i>	62
Figura 50 – Diagrama elétrico com contato em ponte e seu equivalente em <i>ladder</i>	63
Figura 51 – Aplicação das bobinas <i>set</i> e <i>reset</i>	64
Figura 52 – Aplicação do contato e da bobina de detecção positiva.....	65
Figura 53 – Bloco do temporizador.	66
Figura 54 – Diagrama com <i>timer</i> (à esq.) e seu equivalente elétrico (à dir.).	66
Figura 55 – Nível lógico da entrada e da saída do TON.	67
Figura 56 – Nível lógico da entrada e da saída do TOF.....	68
Figura 57 – Contador formatado em bloco (cima) e em bobina (baixo)	68
Figura 58 – Contador crescente.....	69
Figura 59 – Diagrama de eventos do contador crescente.....	70
Figura 60 – Contador decrescente.....	70
Figura 61 – Diagrama de eventos do contador crescente.....	71
Figura 62 – Contador bidirecional.	71
Figura 63 – Diagrama de eventos do contador bidirecional.	72
Figura 64 – Circuito elétrico com duas variáveis de entrada e uma saída.	73
Figura 65 – Portas lógicas básicas.....	74
Figura 66 – Função lógica <i>xor</i>	75

Figura 67 – Funções lógicas em <i>ladder</i>	76
Figura 68 – Tabela-verdade e seu respectivo mapa de Karnaugh.....	77
Figura 69 – Agrupamento de células.....	78
Figura 70 – Tabela-verdade do sistema fictício.....	78
Figura 71 – Mapa de Karnaugh do sistema fictício.	79
Figura 72 – Diagrama <i>ladder</i> para o sistema fictício.	79
Figura 73 – Topologias de rede de comunicação industrial.	81
Figura 74 – Métodos de controle de acesso à rede.	82
Figura 75 – Utilização de <i>gateways</i> para conectar dispositivos.	82
Figura 76 – Sistemas convencional e <i>DeviceNet</i>	83
Figura 77 – Rede Profibus-DP multimestre.....	85
Figura 78 – Sistema SCADA.....	85
Figura 79 – Sistema de suprimento de água.....	86
Figura 80 – Circuito de potência MIT.	88
Figura 81 – Tabelas de entrada e de saída com variáveis associadas.	89
Figura 82 – Lógica em <i>ladder</i>	90

LISTA DE QUADROS

Quadro 1 – Simbologia das chaves.....	21
Quadro 2 – Prefixos para múltiplos binários.....	36
Quadro 3 – Dados da fonte Siemens Simatic PM1207.	39
Quadro 4 – Divisão das linguagens de programação.....	53
Quadro 5 – Simbologia Ladder.....	55
Quadro 6 – Tabela-verdade para o circuito elétrico.	73
Quadro 7 – Leis da álgebra booleana.	75

LISTA DE ABREVIATURAS E SIGLAS

TTL – *Transistor-Transistor Logic*

CMOS – *Complementary Metal-Oxide Semiconductor*

CI – Circuito Integrado

CLP – Controlador Lógico Programável

PLC – *Programmable Logic Controller*

MIT – Motor de Indução Trifásico

CPU – *Central Processing Unit*

IHM – Interface Homem-Máquina

PC – *Personal Computer*

NA – Normalmente Aberto

NF – Normalmente Fechado

E/S – Entrada e Saída

RAM – *Random Access Memory*

ROM – *Read Only Memory*

PROM – *Programmable Read Only Memory*

EPROM – *Erasable Programmable Read Only Memory*

EEPROM – *Electrically Erasable Programmable Read Only Memory*

CC – Corrente Contínua

CA – Corrente Alternada

A/D – Analógico/Digital

D/A – Digital/Analógico

LED – *Light Emissor Diode*

SCADA – *Supervisory Control And Data Acquisition*

CU – *Count Up*

CD – *Count Down*

PV – *Preset Value*

CV – *Count Value*

RESUMO

No mundo moderno os processos industriais necessitam de ferramentas que os tornem cada vez mais eficientes e competitivos. Com a evolução da ciência nas áreas da eletrônica e informática, o uso da lógica cabeada para o controle de sistemas se tornou obsoleto. Os computadores conhecidos como controladores lógicos programáveis (CLPs) se tornaram muito importantes devido às suas diversas vantagens e atualmente marcam forte presença no campo de atuação profissional do engenheiro eletricista. Dessa forma, objetiva-se com este trabalho realizar um estudo bibliográfico acerca desses dispositivos para complementar os conhecimentos obtidos ao longo do curso de graduação. Nesse contexto, almeja-se desenvolver por meio de simulação computacional a programação de um CLP em linguagem *Ladder*, para que este possa ler informações de sensores e acionar atuadores a fim de atender às necessidades de um sistema de bombeamento de água.

Palavras-chave: Controlador Lógico Programável, Linguagem *Ladder*, Controle e Automação, Programação, Eletrônica.

ABSTRACT

In the modern world, industrial processes need tools that make them increasingly efficient and competitive. With the evolution of science in the areas of electronics and informatics, the use of wired logic to control systems has become obsolete. Computers known as programmable logic controllers (PLCs) have become very important due to their various advantages and currently have a strong presence in the electrical engineer's professional field. Thus, the objective of this work is to carry out a bibliographic study about these devices to complement the knowledge obtained during the undergraduate course. In this context, aim to develop, through computer simulation, the programming of a PLC in Ladder language, so that it can read information from sensors and trigger actuators in order to meet the needs of a water pumping system.

Keywords: Programmable Logical Controller, Ladder Language, Control and Automation, Programming, Electronics.

1 INTRODUÇÃO

Os sistemas de controle proporcionam integração e coordenação entre os equipamentos, configurando-se como fundamentais em uma realidade na qual os processos estão cada vez mais otimizados, automatizados e complexos.

Um sistema automatizado é assim definido quando tem capacidade de tomar decisões e executar tarefas em função de sinais provenientes de dispositivos do mesmo sistema, como por exemplo os sensores. Dessa forma, é possível substituir atividades manuais relacionadas a decisão e comando-resposta e, por conseguinte, aumentar a produtividade, a qualidade e os lucros, além de melhorar o ambiente de trabalho.

Os sistemas automáticos normalmente são organizados em duas partes: o circuito de potência e o circuito de comando. O primeiro corresponde à parte responsável pelo movimento, onde se encontram as cargas a serem acionadas. O segundo é a parte inteligente, onde se encontra a lógica do sistema.

Por muitos anos utilizou-se as próprias conexões elétricas entre dispositivos do circuito de comando como a lógica do sistema, fenômeno que recebeu o nome de lógica cabeada. Suas principais limitações são a rigidez na modificação, a complexidade nas ligações e a dificuldade de manutenção ou pesquisa por falhas.

Com a evolução da informática, foi possível desenvolver dispositivos de comando com lógica programável. Trata-se de computadores específicos, conhecidos como controladores lógicos programáveis (CLPs), capazes de armazenar instruções que podem ser alteradas sem remontar todo o circuito elétrico. Essa característica supera as limitações dos modelos anteriores e vai muito além.

O controlador lógico programável é um elaborado sistema eletrônico desenvolvido para operar em ambiente industrial, responsável por realizar funções de controle em diversos níveis de complexidade de maneira flexível, podendo ser facilmente programado.

Com a utilização de um CLP, é possível melhorar praticamente qualquer linha de produção, máquina ou processo. Devido às suas vantagens, estão fortemente presentes em diversos campos de atuação do engenheiro eletricista. Na indústria moderna, dificilmente aplica-se a lógica cabeada para o desenvolvimento de novos sistemas.

Por serem tão importantes, objetiva-se com este trabalho complementar os conhecimentos acerca de acionamentos elétricos por meio do aprendizado sobre o funcionamento dos controladores lógicos programáveis para automatização de sistemas.

Nesse contexto, almeja-se realizar simulação via *software* de um sistema de bombeamento de água controlado por um CLP que recebe dados de sensores, processa as informações em conformidade com um programa desenvolvido na linguagem *Ladder* e aciona atuadores.

2 ASPECTOS HISTÓRICOS

A sociedade busca avanços no campo da automação desde a segunda metade do século XVIII. Nessa época, o modelo de produção artesanal começou a modificar-se para o industrial com a introdução de máquinas adaptadas que substituíam algumas atividades manuais, aumentando a produtividade.

A Primeira Revolução Industrial, no século XIX, foi marcada pelo desenvolvimento dos primeiros sistemas de controle industrial. Tratavam-se de funções de controle implementadas por dispositivos de natureza mecânica que visavam automatizar ações repetitivas nas linhas de produção. Esses dispositivos eram desenvolvidos com finalidade específica e restrita, além de possuir pouco tempo de vida útil.

Como exemplo pode-se citar as válvulas utilizadas para o controle de velocidade da máquina a vapor, que tinham sua abertura ou fechamento controlado pela força centrífuga exercida em duas esferas. Quanto maior a velocidade da máquina, maior a força centrífuga e menor a abertura das válvulas.

Em meados de 1920 foi desenvolvida a lógica a relés, que tomou o lugar dos dispositivos mecânicos e possibilitou a criação de funções de controle mais complexas. Com a utilização de dispositivos de controle eletromecânicos interligados eletricamente, a intervenção humana era necessária apenas em alguns pontos decisórios. Por meio de botões, temporizadores, relés, contadores, sensores, entre outros, era possível criar sequências de movimento lógico após acionar ou desligar motores e atuadores.

Nos anos seguintes, o desenvolvimento no campo da eletrônica possibilitou que os sistemas de controle entrassem em uma nova geração. Com o advento dos circuitos integrados (CIs) baseados nas tecnologias *Transistor-Transistor Logic* (TTL) ou *Complementary Metal-Oxide-Semiconductor* (CMOS), foi alcançada maior durabilidade, agilidade na execução de tarefas e redução de tamanho quando comparados aos dispositivos eletromecânicos.

Entretanto alguns inconvenientes perpetuaram. Com a lógica ainda definida pela interligação elétrica de componentes, a reconfiguração do sistema era difícil e demorada.

Nesse contexto os computadores passaram a ser utilizados para automação, pois suas lógicas são programáveis. Tal fenômeno facilitou modificações na operação

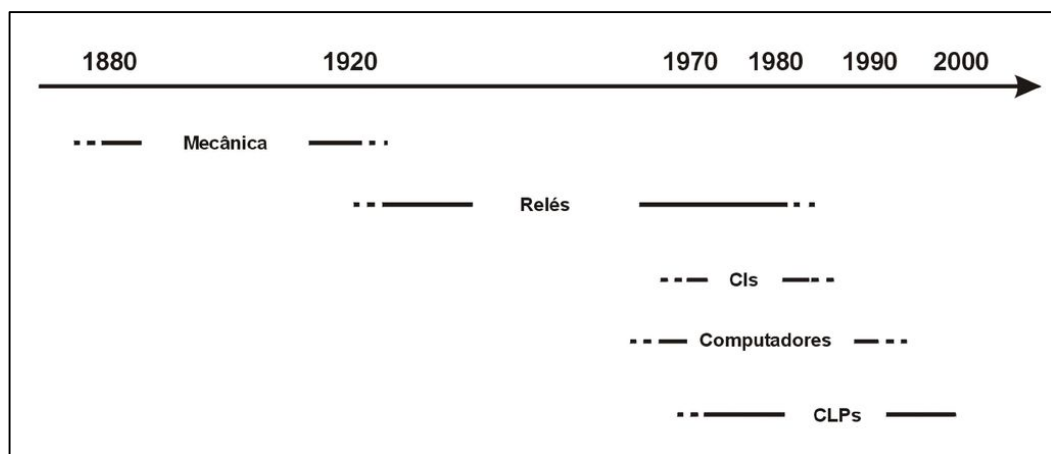
e tornou a automação mais flexível. Ademais, devido à presença de um hardware avançado era possível trabalhar com sinais analógicos, armazenar dados e realizar operações matemáticas.

Os primeiros computadores utilizados para automação, em meados de 1970, eram de finalidade comercial. Eram sensíveis a ambientes adversos, caros e difíceis de programar. Nesse contexto, para atender melhor às necessidades da indústria, foram desenvolvidos computadores mais robustos: os Controladores Lógicos Programáveis.

A Modicon foi pioneira na fabricação de CLPs entre 1970 e 1980, mas com o passar dos anos muitas empresas de automação ingressaram no mercado. Devido a isso, na atualidade encontra-se uma grande variedade de CLPs, com custos mais acessíveis.

Observa-se na figura 1 uma representação da evolução dos sistemas de controle.

Figura 1 – Evolução dos sistemas de controle.



Fonte: FRANCHI; Claiton Moro, 2008 (adaptado).

3 REVISÃO SOBRE ACIONAMENTOS ELÉTRICOS

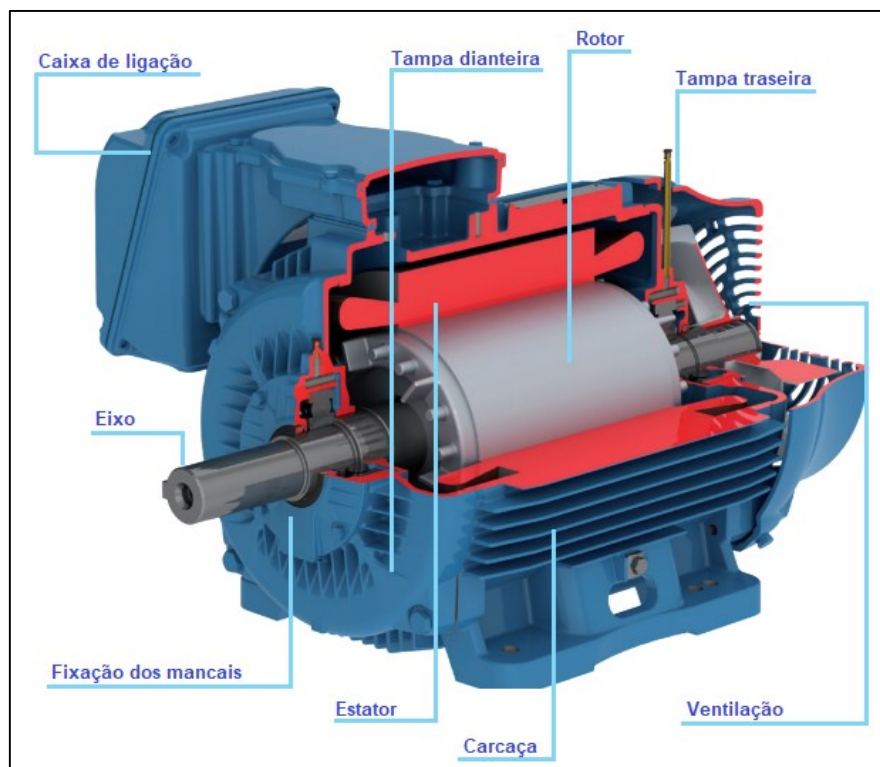
Neste capítulo serão apresentadas breves revisões teóricas sobre alguns dispositivos eletromecânicos que serão utilizados no sistema de suprimento de água, juntamente com as estratégias de acionamento elétrico.

3.1 Motores de Indução Trifásicos

Os motores elétricos são dispositivos capazes de transformar energia elétrica em mecânica. Possuem grande importância econômica e são muito utilizados para acionar equipamentos nas indústrias. Segundo Franchi (2008), cerca de 80% do consumo elétrico do setor industrial é em decorrência do uso de motores.

Dentre os diversos tipos existentes, o motor de indução trifásico (MIT) é o mais aplicado devido à sua robustez, facilidade de partida, construção econômica e eficiência energética. Também está presente em muitas aplicações domésticas e de pequeno porte. A figura 2 ilustra um MIT seccionado, onde é possível visualizar algumas de suas partes constituintes.

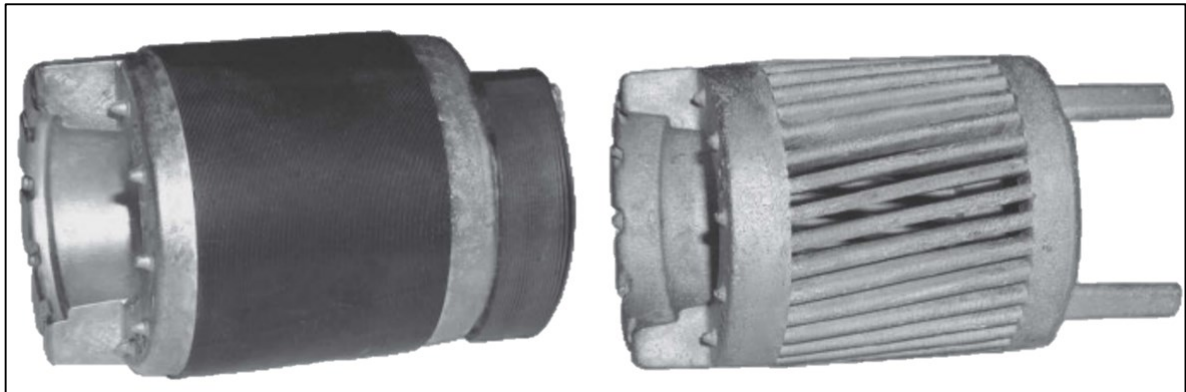
Figura 2 – Motor de indução trifásico.



Fonte: Catálogo Comercial Weg W22 (adaptado).

O MIT é projetado para que seu estator seja alimentado diretamente em corrente alternada, enquanto o rotor recebe corrente por indução. O tipo de rotor mais comum é denominado “gaiola de esquilo”, o qual é construído a partir de barras metálicas inseridas em lâminas de ferro e curto-circuitadas por anéis condutores em ambos os lados. Vide figura 3.

Figura 3 – Rotor gaiola com e sem laminas de ferro (à esq. e à dir.)



Fonte: FITZGERALD; A. E., 2014 (adaptado).

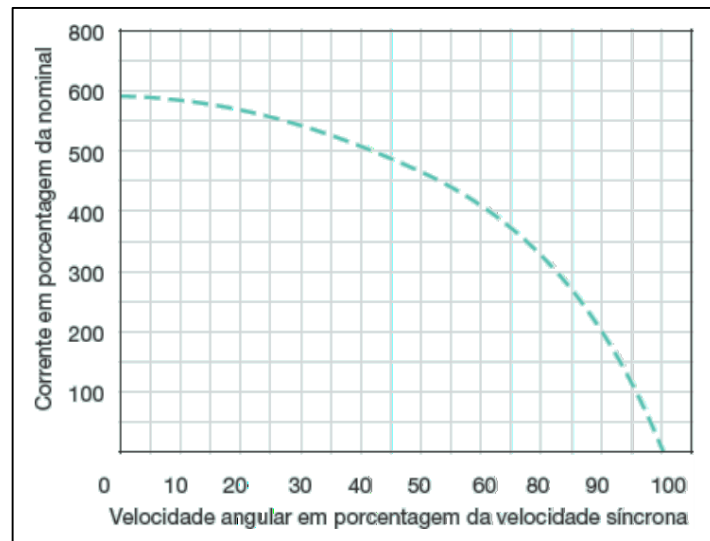
As correntes que circulam nas bobinas do estator produzem um campo magnético girante, cuja rotação é determinada pelo número de polos e pela frequência da rede elétrica. O movimento relativo entre o campo girante e o rotor é denominado escorregamento, e faz com que apareçam tensões induzidas nas barras condutoras.

Durante a partida, quando o rotor está parado, a tensão induzida é máxima e tem a mesma frequência da rede. Como consequência, uma corrente de alto valor circula nas barras e produz outro campo magnético. Este gira em sincronia com campo do estator e faz surgir um conjugado que dá início ao movimento do eixo.

Em condições normais de operação o rotor atinge entre 90% e 98% da velocidade do campo girante e, à medida que o rotor acelera, a tensão e a corrente induzidas diminuem em amplitude e frequência. Como as grandezas elétricas rotóricas e estatóricas estão relacionadas, a corrente solicitada da instalação também sofre redução, permanecendo em um valor aproximadamente seis vezes menor do que na partida.

Observa-se na figura 4 um gráfico que representa esse comportamento, onde o eixo vertical e horizontal representam, respectivamente, a corrente de entrada no motor e a velocidade angular do eixo.

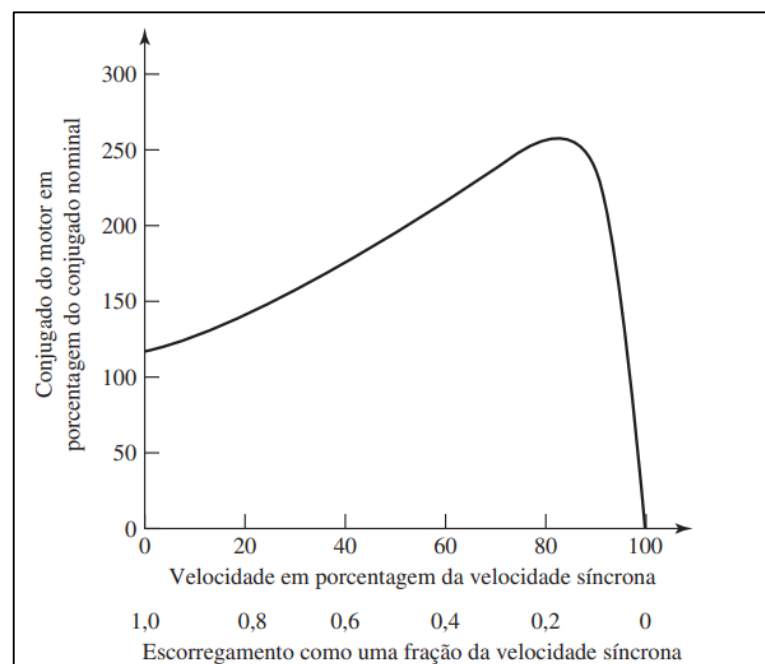
Figura 4 – Corrente de partida.



Fonte: MAMEDE FILHO; João, 2017 (adaptado).

O conjugado produzido pelo MIT também está relacionado, entre outros fatores, ao escorregamento. De acordo com Fitzgerald (2014), ao incrementar o escorregamento o torque aumenta até um ponto de valor máximo e, a partir daí, passa a diminuir. Uma das explicações para esse comportamento é a impedância variável que a gaiola apresenta diante das diferentes frequências de tensão induzida. Observa-se na figura 5 a curva típica do conjugado em função velocidade do rotor.

Figura 5 – Curva típica de conjugado do MIT.



Fonte: FITZGERALD; A. E., 2014 (adaptado).

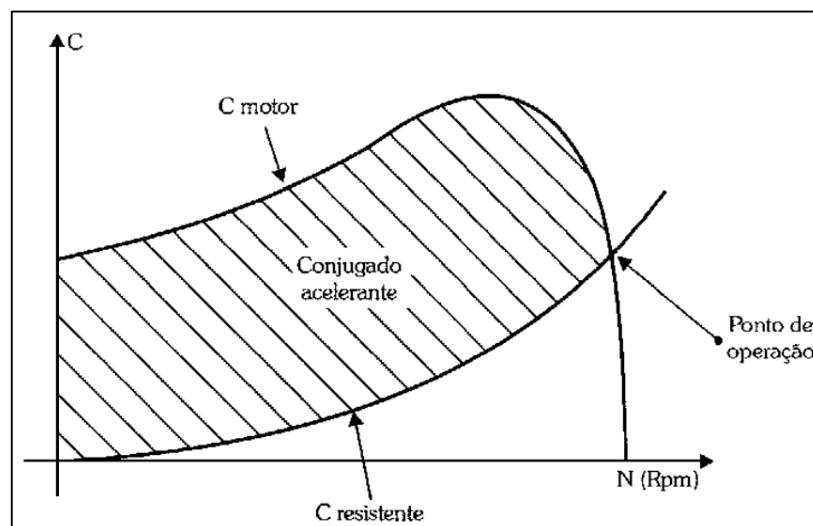
As cargas mecânicas apresentam curvas de conjugado resistente que contrariam a aceleração do motor. Segundo Mamede Filho (2017), essas curvas podem ter comportamento constante, linear, parabólico ou hiperbólico. Nesse contexto, o valor de torque do motor durante a partida pode ser problemático.

As bombas utilizadas para movimentar líquidos são cargas que normalmente apresentam pouca resistência no início do movimento, a qual cresce em função do quadrado da rotação do motor. Isso as caracterizam como cargas parabólicas, categoria sensível a torques de partida elevados.

A figura 6 ilustra as curvas de conjugado do MIT e de uma carga parabólica. Observa-se que em baixas rotações a diferença entre ambas é muito grande (área hachurada), o que resulta em uma brusca aceleração da carga. Caso se trate de um sistema de bombeamento de água, a tubulação pode ser danificada ou até mesmo rompida por esse motivo.

Nessas situações é conveniente aplicar métodos que reduzam os valores de conjugado e suavizem a partida, como por exemplo a chave estrela-triângulo que será abordada posteriormente.

Figura 6 – Conjugado acelerante.



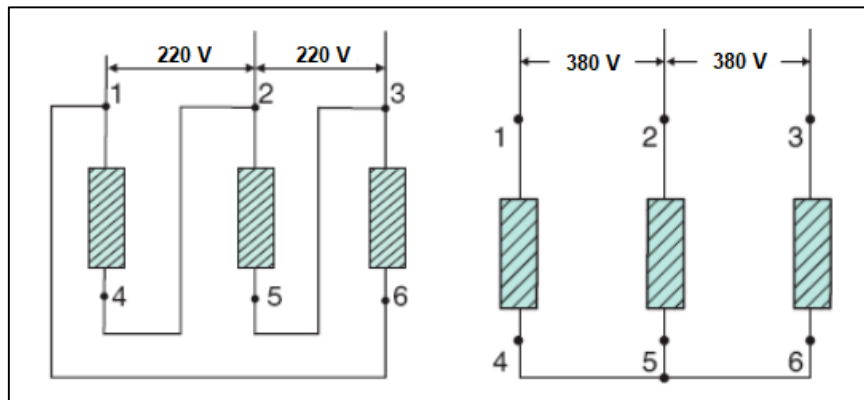
Fonte: FRANCHI; Claiton Moro, 2008.

Cada bobina presente no estator dos motores de indução trifásicos normalmente tem seus terminais disponíveis na caixa de ligação. Sendo assim, é possível realizar diferentes tipos de conexão entre os enrolamentos e energizar o motor em redes elétricas com variados valores de tensão.

As tensões elétricas suportadas e o esquema de ligação são disponibilizados, entre outros dados, na placa de identificação do motor. O menor valor entre as múltiplas tensões caracteriza a tensão nominal da fase, a qual não pode ser ultrapassada. Caso isso aconteça, o motor é avariado.

Os motores que possuem os seis terminais das bobinas disponíveis podem ser ligados em triângulo ou estrela. A primeira ligação é para energizar o motor em redes elétricas com nível de tensão igual ao de suas fases (geralmente 220V). Já a segunda é feita para adequar o motor a redes com tensão $\sqrt{3}$ vezes maior que a de sua fase (380V). Vide figura 7.

Figura 7 – Ligações delta (à esq.) e estrela (à dir.).



Fonte: MAMEDE FILHO; João, 2017 (adaptado).

Existem motores com apenas três terminais disponíveis, fato que impossibilita a mudança da conexão entre as bobinas. Há também motores cujas bobinas são divididas em duas partes, o que resulta em nove ou doze terminais disponíveis para realizar outros tipos de ligação. Entretanto, esses modelos não vêm ao caso da aplicação deste trabalho.

3.2 Dispositivos de Proteção e Comando

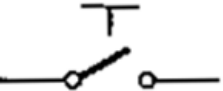
Dentre a grande variedade de dispositivos de proteção e comando existentes, para o propósito deste trabalho é pertinente apresentar apenas três: chaves, contadores e disjuntores.

As chaves são dispositivos comandados manualmente, e têm como finalidade estabelecer ou interromper a passagem de corrente em um circuito. São classificadas como retentivas ou não-retentivas.

Os modelos retentivos são aqueles que, após acionados, só retornam à sua posição anterior através de um novo acionamento. Já os não-retentivos retornam ao seu estado anterior assim que a força do acionamento é cessada, característica que os tornam conhecidos como chaves de impulso.

Ademais, uma chave pode possuir seu contato normalmente aberto (NA) ou normalmente fechado (NF). O primeiro é aquele que originalmente não permite a passagem de corrente e, ao ser acionado, passa a conduzi-la. Já o segundo conduz em sua posição original, e interrompe a condução quando acionado. Observa-se ilustrado no quadro 1 a simbologia das chaves.

Quadro 1 – Simbologia das chaves.

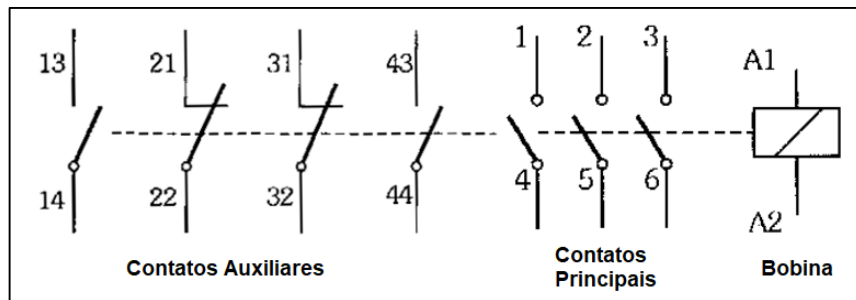
	Normalmente Aberto	Normalmente Fechado
Chave Retentiva		
Chave Não Retentiva		

Fonte: Elaborado pelo autor com base em FRANCHI; Claiton Moro, 2008.

De forma similar às chaves, os contatores são dispositivos capazes de estabelecer ou interromper a passagem de corrente elétrica. Entretanto, são operados por meio de uma bobina que muda a posição de seus contatos quando energizada. As bobinas podem ser especificadas para alimentação em corrente alternada ou contínua com uma ampla faixa de valores, e seus terminais são identificados por A1 e A2.

De acordo com Franchi, cada contator pode apresentar contatos principais para manobras no circuito de potência e contatos auxiliares para utilização no circuito de comando. Geralmente há três contatos principais, e seus terminais recebem numeração de 1 a 6. Já os contatos auxiliares se apresentam em quantidades variadas e são classificados em NA ou NF, com terminais identificados por 3 e 4 ou 1 e 2 no final de sua numeração. Vide figura 8.

Figura 8 – Simbologia dos contatores.

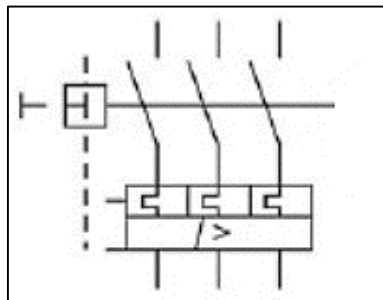


Fonte: FRANCHI; Claiton Moro, 2008 (adaptado).

Os disjuntores são dispositivos destinados à proteção dos circuitos. Seus contatos são fechados manualmente por meio uma alavanca e abertos por disparadores. Estes podem ser térmicos ou magnéticos, para proteção contra sobrecorrente ou curto-circuito, respectivamente.

Existem nas formas monopolar, bipolar, tripolar ou tetrapolar e são dimensionados em função da tensão de operação, corrente nominal, tempo de atuação, capacidade de interrupção, entre outros fatores. Sua simbologia está ilustrada na figura 9.

Figura 9 – Simbologia de um disjuntor com disparadores térmico e magnético.



Fonte: PRUDENTE; Francesco, 2011 (adaptado).

3.3 Chave de Partida

A partida dos motores de indução trifásicos é um momento que requer atenção tanto do ponto de vista elétrico quanto mecânico. Nesse contexto, a chave de partida deve ser selecionada de forma a atender às necessidades da instalação elétrica e da carga acionada.

Conforme apresentado anteriormente, a corrente de partida solicitada pelo MIT é muitas vezes maior do que a nominal, o que pode causar subtensão na rede e até mesmo fazer com que dispositivos de proteção atuem. O alto torque de partida, por sua vez, pode acelerar bruscamente cargas que possuam pouco conjugado resistente e danificar o sistema.

Considerando um sistema de bombeamento de água, a chave estrela-triângulo é uma boa escolha por amenizar os impactos negativos causados pela corrente e pelo torque. O método consiste em energizar o motor com suas bobinas conectadas em estrela e, após um determinado tempo, comutar para triângulo.

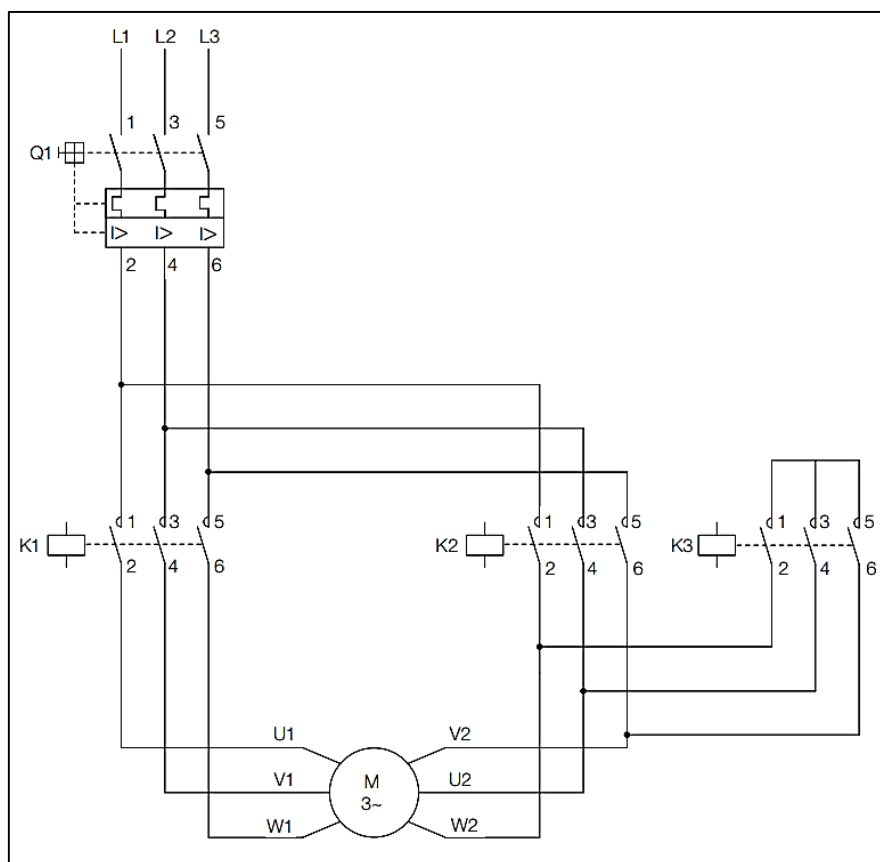
No momento em que o motor partir com a utilização da chave, a tensão aplicada em seus terminais não será alterada. Entretanto, devido à característica da conexão das bobinas, cada uma de suas fases receberá um valor de tensão igual ao nominal dividido por $\sqrt{3}$ (aproximadamente 58%). Por esse motivo, tanto a corrente quanto o conjugado de partida ficam reduzidos a $1/3$ de seus valores originais.

Trata-se de um método de partida econômico do ponto de vista financeiro, haja vista que requer apenas três contatores e um temporizador para que seja realizada a mudança na ligação das bobinas.

Observa-se na figura 10 o diagrama do circuito de potência para a utilização da chave. O circuito de comando não será exibido pois a lógica de acionamento sequencial a relé será substituída por um controlador lógico programável (CLP).

A primeira etapa de operação consiste em fechar o disjuntor Q1 e, em seguida, os contatores K3 e K1. Nesse momento, o motor é energizado com as bobinas ligadas em estrela e um temporizador é acionado. Passado o tempo configurado, K3 é aberto e K2 fechado, o que implica na troca da ligação de estrela para triângulo.

Figura 10 – Circuito de potência chave estrela-triângulo.



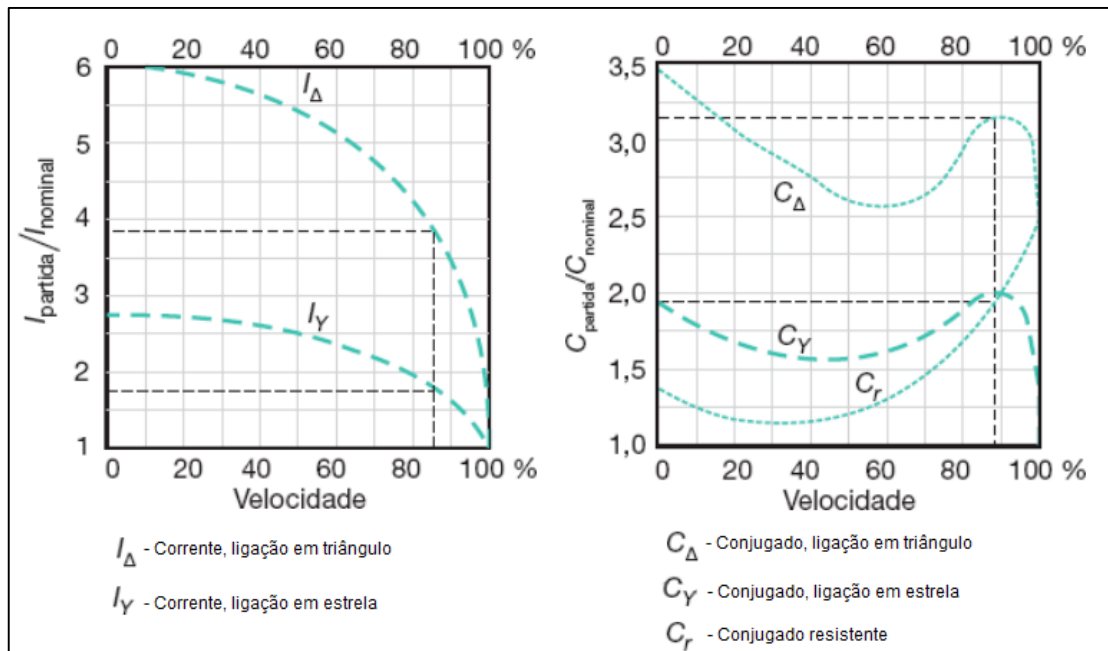
Fonte: Guia de seleção de partidas Weg (adaptado).

Apesar da suavização na corrente de partida com a utilização do método, há um transitório inconveniente na comutação entre as ligações estrela e triângulo. No momento da troca, o MIT é brevemente desligado e o campo magnético girante é perdido. Ao ser energizado com a nova conexão, ocorre um pico de corrente com valores próximos à da corrente de partida original, porém com duração muito menor.

Devido à grande redução na curva de conjugado produzido durante a partida, o MIT pode enfrentar dificuldades para acelerar diversos tipos de carga. Entretanto, quando se trata de cargas com baixo valor de conjugado resistente, o eixo do motor irá acelerar sem dificuldades e de forma menos abrupta.

O gráfico à esquerda na figura 11 ilustra as curvas de corrente de partida em relação à nominal quando o motor está conectado em triângulo e em estrela. Já o gráfico à direita ilustra as curvas de conjugado em relação ao nominal para as mesmas situações. Verifica-se que na figura está indicado uma velocidade entre 80% e 90%, que é o momento adequado para realizar a comutação estrela-triângulo.

Figura 11 – Curvas de corrente e conjugado em função da velocidade



Fonte: MAMEDE FILHO; João, 2017 (adaptado).

4 FUNDAMENTAÇÃO TEÓRICA SOBRE CLP

Neste capítulo serão apresentados os conceitos acerca da composição, operação e programação dos controladores lógicos programáveis. Trata-se da parte teórica necessária para a compreensão e utilização dos mesmos.

4.1 Definição, Princípios de Funcionamento e Constituição

Segundo Franchi (2008), os controladores lógicos programáveis podem ser definidos como equipamentos de estado sólido capazes de executar instruções para controlar dispositivos, máquinas e processos por meio da implementação de funções, como lógica de controle, sequenciamento, controle de tempo e operações matemáticas.

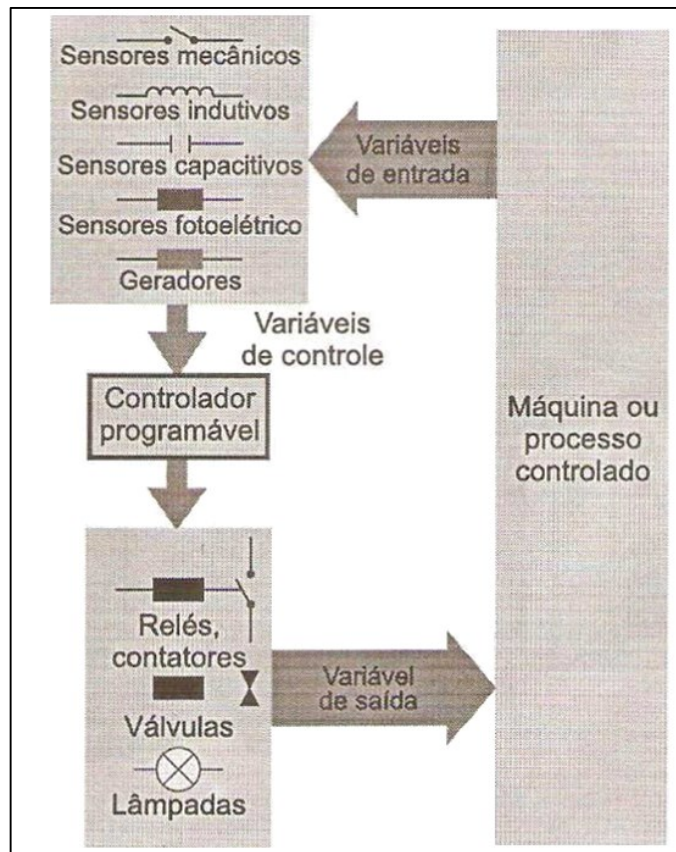
Trata-se de computadores projetados para trabalhar em ambientes severos, com variações de temperatura, umidade, vibrações, distúrbios elétricos e poluição atmosférica.

Os CLPs operam por meio da leitura de variáveis em suas entradas, as quais são processadas em conformidade com uma programação estabelecida pelo operador e, em seguida, as variáveis de saída são atualizadas.

As variáveis de entrada na maioria das vezes são sensores que informam ao controlador, de forma digital ou analógica, as condições em que se encontra o sistema controlado. A leitura dessas informações é feita a cada instante e, em função do programa instalado, o CLP toma decisões e atua no sistema por meio de relés, contadores, válvulas, entre outros.

Observa-se na figura 12 uma representação do funcionamento de um CLP.

Figura 12 – Funcionamento de um CLP.



Fonte: NATALE; Ferdinando, 2008.

Entende-se como *hardware* de um CLP todos os componentes que podem ser fisicamente tocados. De acordo com Franchi (2008), o *hardware* é constituído de cinco componentes fundamentais:

- Unidade Central de Processamento, conhecida pela sigla inglesa CPU (*Central Processing Unit*);
- Fonte de alimentação;
- Módulos de entrada, digitais ou analógicos;
- Módulos de saída, digitais ou analógicos e
- Unidade de comunicação.

Em muitos modelos de controladores, a fonte de alimentação fica inclusa no mesmo bloco da CPU. Observa-se na figura 13 uma ilustração das cinco partes em forma de diagrama de blocos.

Com a presença de componentes de estado sólido, o controlador lógico programável apresenta elevado grau de confiabilidade se comparado aos sistemas com lógica eletromecânica. Ademais, com a evolução da eletrônica de semicondutores, o microprocessador proporciona alta performance na execução de tarefas de automação industrial.

Devido à sua capacidade de expansão, os CLPs têm um número de contatos teoricamente ilimitado. Dessa forma, são capazes de controlar uma grande quantidade de dispositivos em suas entradas e saídas.

Por meio da linguagem de programação é possível eliminar dispositivos do circuito de comando, como contadores e temporizadores, o que possibilita a redução de custos e de espaço ocupado.

Aliado à isso, os CLPs apresentam dimensões reduzidas e têm seu preço cada vez mais acessível, o que o torna uma opção ideal em diversas aplicações de automação industrial.

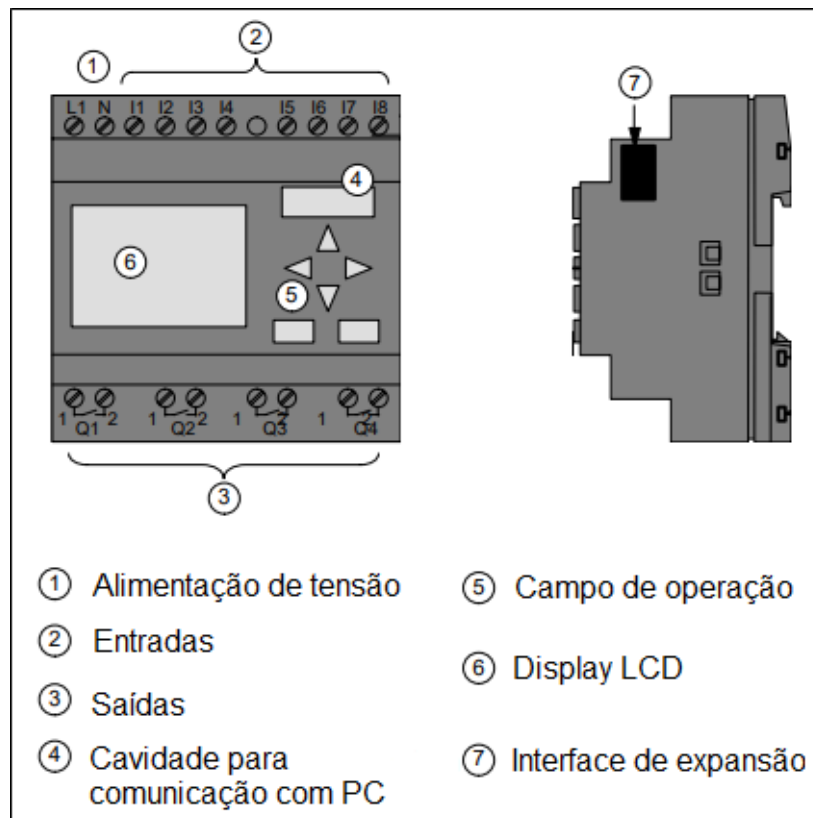
4.2 Formas Básicas

As especificações de um controlador lógico programável quanto ao poder de processamento, conectividade, quantidade de entradas e saídas e capacidade de memória varia conforme o modelo e o fabricante. Os diversos modelos podem ser classificados em duas formas básicas: compactos e modulares.

Os CLPs compactos possuem o processador, memória, fonte de alimentação e conectores do sistema de entradas e saídas (E/S) incorporados em uma única unidade, sendo que apenas os conectores ficam expostos. Geralmente a estrutura compacta é empregada em controladores de pequeno porte, os quais apresentam baixo custo e são utilizados em sistemas com poucos sensores e atuadores.

A maioria desses modelos conta com a possibilidade de adicionar módulos especiais, tais interfaces homem-máquina (IHM) para programação e/ou visualização dos parâmetros, módulos de comunicação e até mesmo pequenas expansões de E/S. A figura 14 ilustra um CLP compacto da Siemens.

Figura 14 – CLP Compacto Siemens LOGO! 230RC.

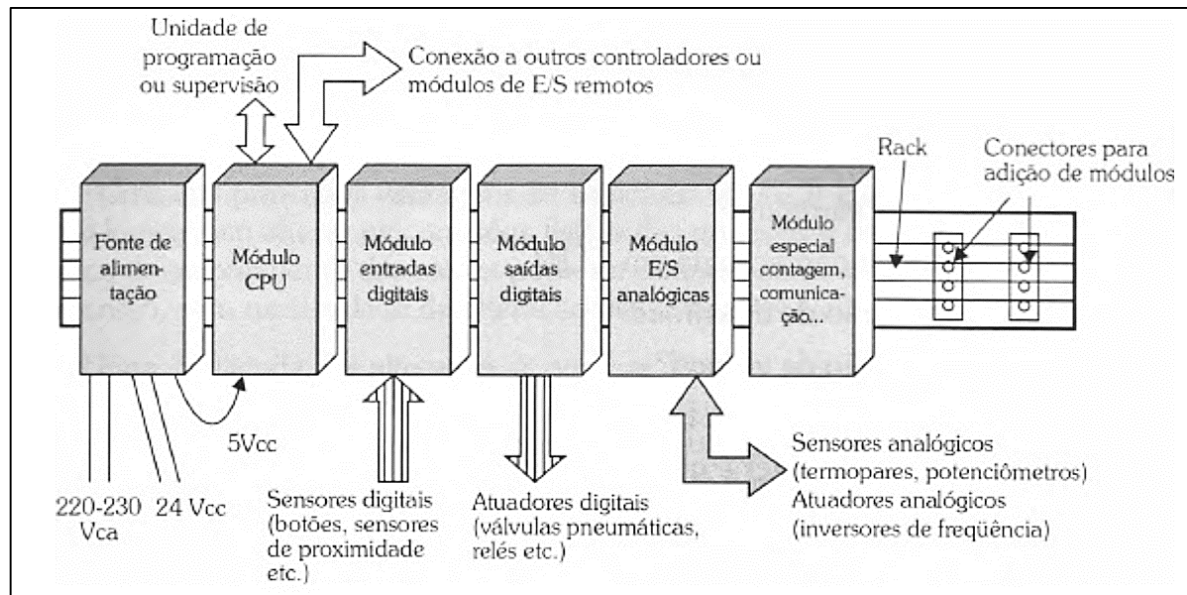


Fonte: Manual de Instruções do LOGO! Siemens (adaptado).

Já os CLPs de médio e grande porte são compostos por uma estrutura modular. Possuem processador e memória localizados no mesmo gabinete, podendo a fonte estar inclusa ou separada. Cada um dos módulos de E/S encontram-se separados da CPU e são montados de acordo com a configuração exigida, em posições pré-definidas (*rack*).

Os CLPs modulares são capazes de controlar desde uma pequena quantidade até milhares de pontos de entradas e saídas. A figura 15 ilustra a arquitetura de um CLP modular.

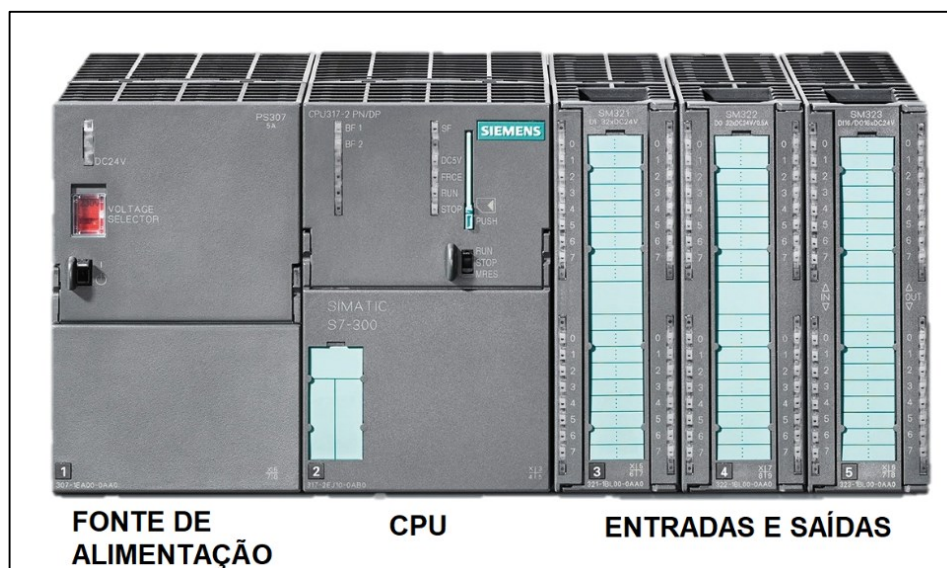
Figura 15 – Arquitetura de um CLP modular.



Fonte: FRANCHI; Claiton Moro, 2008.

Observa-se na figura 16 o Siemens SIMATIC S7-300, um exemplo de CLP modular de médio porte. No módulo à esquerda se localiza a fonte de alimentação e, no centro, a unidade central de processamento. Os três módulos à direita são compostos por entradas e saídas digitais.

Figura 16 – Siemens SIMATIC S7-300.



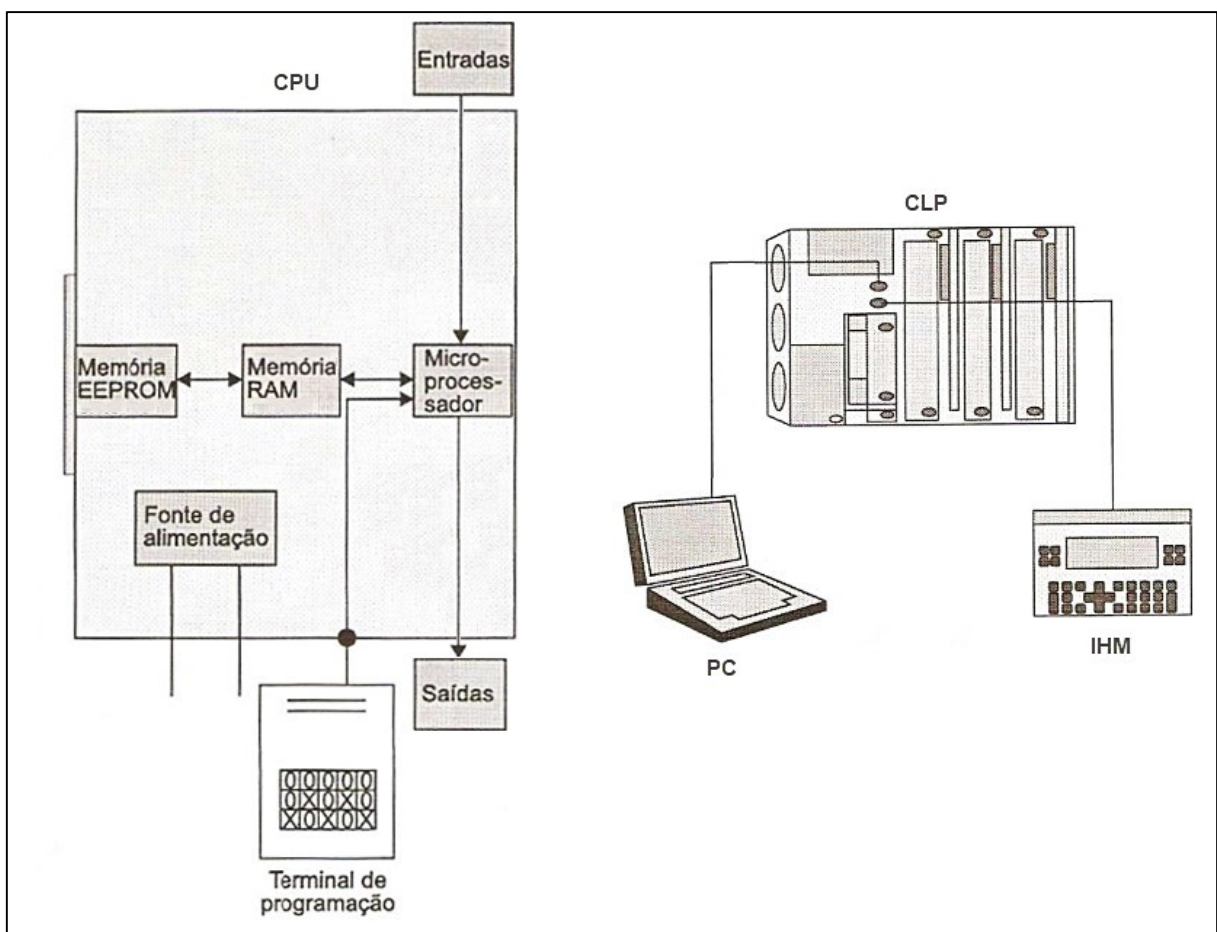
Fonte: Página de Controladores Avançados Siemens Simatic (adaptado).

4.3 Unidade de Processamento Central

A unidade de processamento central (CPU) é considerada a parte mais importante do CLP, pois é ela quem comanda todas as atividades do dispositivo. É responsável por ler os sinais analógicos e digitais presentes nos módulos de entrada, receber comandos da rede de comunicação quando for o caso, executar o programa do usuário e transferir para os cartões de saída as ordens provenientes da programação.

A CPU é composta por três elementos principais: microprocessador, sistema de memórias e fonte de alimentação, quando inclusa. Nessa unidade geralmente encontram-se também terminais de programação, como interfaces homem-máquina (IHM) e/ou portas de comunicação nas quais são conectados computadores pessoais (PCs) para elaborar o programa do usuário, conforme ilustrado na figura 17. Ademais, verifica-se também indicadores de operação e diagnóstico.

Figura 17 – Constituição da CPU.



Fonte: NATALE; Ferdinando, 2008 (adaptado).

Segundo Prudente (2011), o microprocessador é a peça mais complexa do controlador, sendo capaz de realizar operações em alta frequência e com baixo consumo energético. Os fabricantes geralmente especificam o poder de processamento com base no tempo gasto para executar uma instrução.

Os microprocessadores têm como função ler as informações contidas nos módulos de E/S e no sistema de memórias, executar as instruções do programa do fabricante (*firmware* ou sistema operacional), realizar operações de diagnóstico, executar o programa do usuário, atualizar os dados com os resultados das operações, tudo de forma cíclica e contínua.

Ainda segundo Prudente, as memórias são circuitos integrados (chips) que possuem capacidade de armazenar informações, as quais acessadas pelo microprocessador e essenciais para o funcionamento do CLP. As memórias são separadas conforme as seguintes características:

- Volatilidade;
- Possibilidade de leitura e escrita;
- Velocidade de escrita e leitura e
- Capacidade.

De acordo com Franchi (2008), existem seis principais tipos de memórias utilizadas em controladores lógicos programáveis. Cada tipo de memória possui características que afetam em sua forma de operar e a torna ideal para uma determinada aplicação.

- Memória ROM;
- Memória RAM;
- Memória PROM;
- Memória EPROM;
- Memória EEPROM e
- Memória *Flash*.

A sigla ROM, proveniente do inglês *Read Only Memory*, ou memória apenas de leitura, é projetada para ter seu conteúdo gravado uma única vez e nunca mais alterado. Trata-se de um armazenamento não volátil, ou seja, os dados não são perdidos em decorrência de ruídos elétricos ou quedas de energia. Normalmente o fabricante do CLP utiliza a ROM para guardar o *firmware*, que é o *software* mais importante do dispositivo.

A RAM (*Random Access Memory*), por sua vez, possui como característica a possibilidade de escrita e leitura de dados em qualquer posição da memória, o que implica em altas velocidades. Trata-se de uma memória volátil que normalmente é utilizada nos CLPs para armazenar o programa do usuário. Para evitar a perda dos dados em casos de falta de energia, utiliza-se uma bateria de *backup* (figura 18) ou até mesmo capacitores de alta capacitância como no Siemens S7-200. Apresenta bons resultados na maioria das aplicações com CLPs.

Figura 18 – Bateria de *Backup* 3.6V 0,85Ah de Lítio para CLP Siemens.

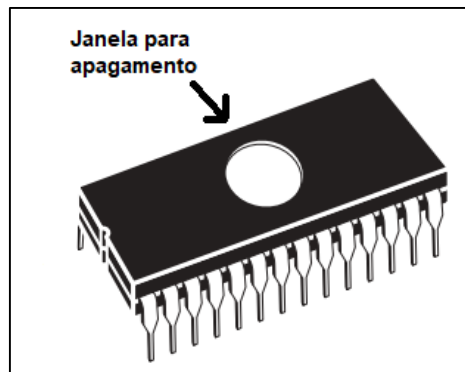


Fonte: Datasheet SIMATIC S7 Backup Battery (adaptado).

A PROM (*Programmable Read Only Memory*) é um tipo raro e especial de ROM. Tem a capacidade de ser programada por uma única vez e, por esse motivo, é conhecida também como memória OTP (*One Time Programmable*). Possui como vantagem a não volatilidade, entretanto apresenta rigidez em programar e exige equipamentos específicos para essa finalidade. É utilizada para a armazenagem de programas que não necessitam de alterações ou inserção de dados durante a operação do CLP.

A EPROM (*Erasable Programmable Read Only Memory*) nada mais é do que uma PROM que pode ser apagada e reprogramada. São capazes de armazenar um programa até que o mesmo precise ser alterado, entretanto não permitem modificações ou escrita de dados durante a operação do CLP. O chip da EPROM possui uma pequena janela (figura 19), a qual deve ser exposta a uma fonte de luz ultra violeta por vários minutos para que seu conteúdo seja apagado. Assim como o modelo anterior, são não-voláteis.

Figura 19 – Memória EPROM.



Fonte: Datasheet ST Electronics EPROM M27C512 (adaptado).

A EEPROM (*Electrically Erasable Programmable Read Only Memory*) é uma evolução do modelo anterior e apresenta como principal vantagem a capacidade de ser reprogramadas através do uso de sinais elétricos. Basta um comando via software e os dados são apagados e reescritos sem a necessidade de remover o chip do controlador. Está presente em muitos CLPs de pequeno e médio porte e fornece armazenamento não volátil para o programa. É muito parecidas com a memória RAM, entretanto possui velocidades de escrita menores. Um novo dado só pode ser escrito após o conteúdo anterior ser apagado, o que a torna lenta para modificações durante a operação do CLP. Ademais, apresenta limitação no número de vezes em que se pode apagar seu conteúdo.

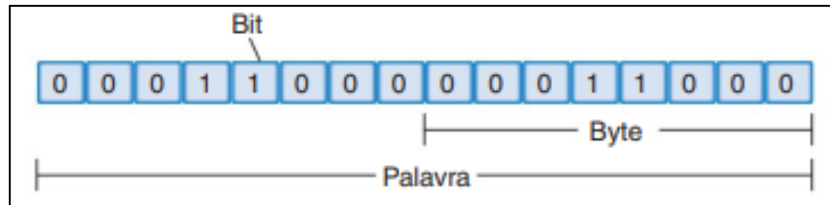
O último tipo, a memória *Flash*, é uma das mais novas no mundo da microeletrônica. Possui todas as vantagens das memórias EEPROM e apresenta maior velocidade de escrita por gravar dados em blocos de *bytes* ao invés de um *byte* por vez. Ademais, possui maior capacidade de armazenamento, confiabilidade, durabilidade e menor consumo energético. Está presente na maioria dos CLPs modernos, como Siemens S7-300.

De acordo com Prudente (2011), a característica mais importante da memória é aquela que indica a quantidade de dados que a mesma pode armazenar. Os computadores em geral tratam de variáveis que possuem apenas dois estados lógicos: ligado ou desligado, conhecidos como 1 e 0. Essas variáveis recebem o nome de *bit* (*binary digit*), que é a menor unidade da informação. O *bit* é capaz de formar dados mais complexos quando em grandes quantidades.

Os *bits* geralmente são organizados em grupos de 8 *bits*, chamados de *bytes*, em grupos de 16 *bits*, chamados de *words* ou em grupos de 32 *bits*, chamados de

double words. A capacidade de uma memória é determinada pelo número de bits que ela pode armazenar. Observa-se na figura 20 uma ilustração que representa algumas dessas unidades de memória.

Figura 20 – Unidades de memória.



Fonte: PETRUZELLA; Frank, 2014.

Nas aplicações em CLP o número de bits seria muito extenso e, por esse motivo, faz-se necessária a utilização de múltiplos. Entretanto, devido à lógica da numeração binária ser representada em potência de 2, não é adequado utilizar os múltiplos definidos pelo Sistema Internacional de Unidades (SI – *Système international d'unités*).

Nesse contexto, no ano de 1998 a Comissão Eletrotécnica Internacional (IEC – *International Electrotechnical Commission*) definiu os nomes e símbolos para prefixos de múltiplos de binários e publicaram-nos como emenda na norma IEC 60027-2.

No nome do prefixo foram mantidas as duas primeiras letras do SI e as duas últimas foram substituídas por “bi”, em referência a palavra “*binary*”. Já o símbolo é o mesmo de seu equivalente no SI, com a adição da letra “i” no final. Observa-se esses detalhes no quadro 2.

Quadro 2 – Prefixos para múltiplos binários

Fator	Nome	Símbolo	Origem	Derivação (SI)	Equivalente Numérico
2^{10}	Kibi	Ki	<i>Kilobinary</i>	Kilo (10^3)	1.024
2^{20}	Mebi	Mi	<i>Megabinary</i>	Mega (10^6)	1.048.576
2^{30}	Gibi	Gi	<i>Gigabinary</i>	Giga (10^9)	1.073.741.824
2^{40}	Tebi	Ti	<i>Terabinary</i>	Tera (10^{12})	1.099.511.627.776
2^{50}	Pebi	Pi	<i>Petabinary</i>	Peta (10^{15})	1.125.899.906.842.624
2^{60}	Exbi	Ei	<i>Exabinary</i>	Exa (10^{18})	1.152.921.504.606.846.976

Fonte: Elaborado pelo autor com base em Tannus, 2019.

Apesar de a nomenclatura para múltiplos binários estar definida por norma, a maioria dos fabricantes não emprega a terminologia. Comumente, utiliza-se a nomenclatura do SI para representar os valores binários.

A quantidade de memória exigida de um CLP está relacionada ao número de entradas e saídas em utilização, ao tipo de sinal trabalhado (analógico ou digital) e ao tamanho do programa do usuário. Dessa forma, quanto mais complexo o sistema controlado, maior a capacidade de memória necessária. Muitos controladores possibilitam a expansão dessa quantidade, o que permite o reaproveitamento do dispositivo em caso de ampliação do sistema sob controle.

Ainda de acordo com Prudente, a memória de um CLP pode ser dividida em quatro partes:

- Memória de programa;
- Memória para as funções internas;
- Memória para o estado das entradas e saídas e
- Memória para o sistema operacional.

A memória de programa é aquela que armazena o programa feito pelo usuário do CLP. As instruções são salvas em uma sequência de bits lida pelo microprocessador, que executa cada uma delas de forma sequencial, da primeira até a última. Após ler e executar a última instrução, o processo recomeça a partir da primeira em um ciclo de *scan*.

O chip da memória de programa é do tipo EEPROM ou RAM, sendo que neste último caso é necessária uma fonte de alimentação extra para a memória com intuito de evitar a perda dos dados quando houver queda de energia. Conforme explicado anteriormente, essa fonte de energia suplementar pode ser uma bateria de *backup* ou até mesmo um capacitor.

A memória para as funções internas é aquela destinada a armazenar valores intermediários decorrentes da lógica da programação do usuário. Trata-se de um resultado que não é transferido para nenhuma saída do CLP, mas é armazenado em um local de memória de forma temporária para que outras equações lógicas possam utilizá-lo. Pode-se citar como exemplo a utilização de funções de temporização.

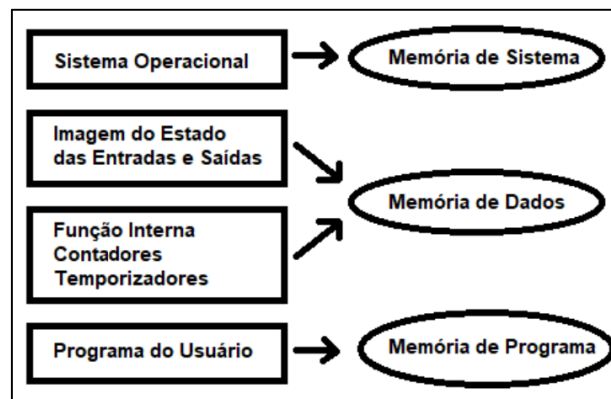
Já a memória para o estado das E/S, como o próprio nome diz, é responsável por registrar o estado de cada uma das entradas e saídas do CLP utilizadas no programa. O microprocessador aloca essa informação em um local fixo da memória e atualiza seu valor de forma cíclica.

Ambas as memórias para as funções internas e para o estado das E/S são conhecidas como memória de dados e o chip utilizado é do tipo RAM com tecnologia CMOS.

Por fim, a memória para o sistema operacional, também conhecida como memória de sistema ou *firmware*, é o local onde é gravado o programa desenvolvido pelo fabricante para garantir o correto funcionamento do dispositivo. Como não se objetiva alterar esses dados, os mesmos são gravados em uma ROM ou EPROM.

Observa-se na figura 21 uma ilustração que resume as partes da memória de um controlador.

Figura 21 – Partes da memória de um CLP.



Fonte: PRUDENTE; Francesco, 2011 (adaptado).

A fonte de alimentação de um CLP é responsável por converter a energia da rede elétrica em níveis de tensão adequados para seu correto funcionamento. Conforme apresentado anteriormente, a fonte pode ser externa ou estar inclusa na CPU do controlador.

Caso a fonte de alimentação seja interna é necessário que o CLP seja diretamente ligado à rede, e o mesmo deve ser capaz de alimentar seu circuito, suas entradas e saídas e também o módulo de comunicação. Dessa forma, se a potência exigida por esse conjunto for superior à especificada no dispositivo, será necessário acrescentar uma fonte complementar.

Já os controladores com fonte de alimentação externa não enfrentam esse inconveniente, uma vez que a mesma pode ser dimensionada para alimentar todo o sistema controlado e suas possíveis expansões.

As fontes utilizadas podem ser do tipo linear ou chaveada. As fontes lineares normalmente utilizam um transformador para abaixar o valor da tensão de entrada, um circuito de retificação, um filtro e um estabilizador. Já as fontes chaveadas

possuem um circuito de conversão CC/CC de alta frequência, o que reduz as dimensões da fonte e aumenta seu rendimento.

Geralmente as fontes aceitam uma ampla faixa de valores de tensão alternada em sua entrada e oferecem tensão contínua de 12V, 24V ou 48V em sua saída, que são os valores mais utilizados pelos sensores e atuadores ligados ao controlador. Ademais, para sistemas com maior potência existem fontes com entrada trifásica e tensões mais elevadas. Observa-se na figura 22 um modelo de fonte externa de CLP.

Figura 22 – Fonte Siemens Simatic PM1207.



Fonte: Datasheet Simatic S7-1200 Power Module PM1207 (adaptado).

Verifica-se no quadro 3 as especificações técnicas da fonte externa apresentada acima.

Quadro 3 – Dados da fonte Siemens Simatic PM1207.

Tipo de alimentação na entrada	Monofásica
Tensão de entrada	Seleção automática 120V/230V
Tensões limite	85V a 132V ou 176V a 264V
Frequência	47 a 63 Hz
Tensão de saída	24V
Ripple máximo	150mV
Curva de tensão de saída	Controlada e isolada
Potência ativa típica na saída	60W
Rendimento	83%
Temperatura de operação	0 a 60°C

Fonte: Elaborado pelo autor com base em Datasheet Simatic S7-1200.

4.4 Entradas e Saídas

Os módulos de entrada e saída são dispositivos responsáveis por realizar a comunicação do controlador lógico programável com o sistema por meio do tratamento dos sinais.

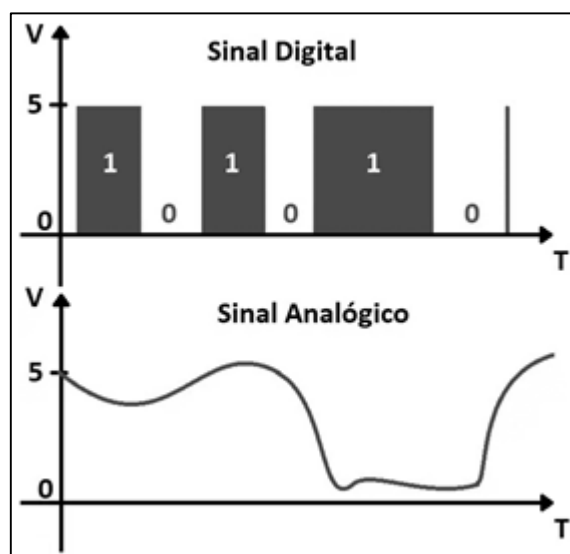
Para que o CLP exerça sua função, é necessário que o mesmo receba informações do processo controlado, tome decisões e envie comandos de atuação. Nesse contexto, os módulos de entrada e saída têm como objetivo tornar compatíveis os níveis de tensão e corrente para que as informações sejam transmitidas entre CPU, sensores e atuadores.

Os módulos de entrada servem de interface entre os sinais provenientes dos sensores e a CPU, já os módulos de saída adequam os sinais oriundos da CPU e os envia para os atuadores. Esses sinais podem ser digitais ou analógicos (figura 23).

Os sinais digitais, também conhecidos como discretos, são os que assumem apenas dois estados bem definidos: presença ou ausência de tensão, verdadeiro ou falso, ativado ou desativado, 1 ou 0.

Os sinais analógicos são a representação de grandezas contínuas e podem assumir qualquer valor entre um limite inferior e um superior, intervalo conhecido como *range*. Possuem tratamento mais difícil do que o dos sinais digitais.

Figura 23 – Sinais digital e analógico.

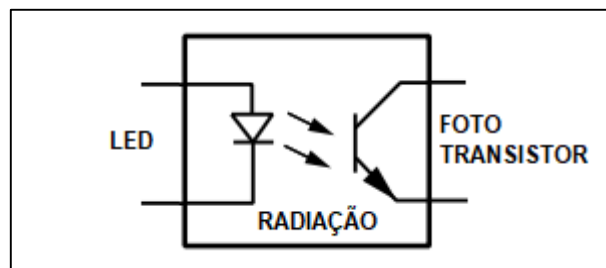


Fonte: Embarcados, 2017.

De acordo com Prudente (2011), o sistema controlado pode apresentar ruídos, variações de tensão e perturbações que podem danificar o CLP. Para evitar que isso aconteça, as placas de E/S geralmente isolam eletricamente a parte lógica da parte de força.

Essa isolamento, na maioria das vezes é feita por meio de um optoacoplador (figura 24). Trata-se de um diodo emissor de luz (LED) que ao ser energizado emite radiação a um fototransistor. Quando este recebe o pulso luminoso, passa a conduzir corrente.

Figura 24 – Optoacoplador.



Fonte: Datasheet Vishay Semiconductor (adaptado).

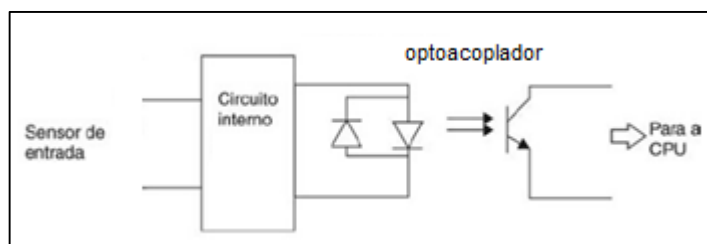
Ainda segundo o autor, em decorrência de existir sinais digitais e analógicos, há quatro módulos de entrada e saída presentes no mercado:

- Módulo de entrada digital;
- Módulo de saída digital;
- Módulo de entrada analógica e
- Módulo de saída analógica.

Os módulos de entrada digital são aqueles que recebem sinais discretos de elementos como botões, chaves, sensores, chaves de fim de curso, pressostatos, entre outros que emitem sinais com apenas dois estados.

O sinal digital proveniente do sensor pode possuir diferentes valores de tensão em corrente contínua ou corrente alternada. Os valores mais comuns em CC são 12V, 24V e 48V; e em CA são 12V, 24V, 120V e 220V. O sinal é recebido no módulo de entrada, onde é transformado, retificado e estabilizado por um circuito interno. O sinal resultante é transmitido através de um optoacoplador e por fim chega à CPU, conforme ilustrado na figura 25.

Figura 25 – Interface de entrada de um sinal digital.



Fonte: PRUDENTE; Francesco, 2011 (adaptado).

Existem módulos com diferentes quantidades de pontos de entrada digitais, podendo ser adquiridos em conformidade com o número de sensores do sistema.

Os módulos de saída digital são responsáveis por enviar comandos discretos a atuadores que trabalhem apenas com dois estados, tais como eletroválvulas, lâmpadas, alarmes, entre outros. Esses cartões eletrônicos possuem três tipos diferentes de saídas: a relé, a transistor e a TRIAC.

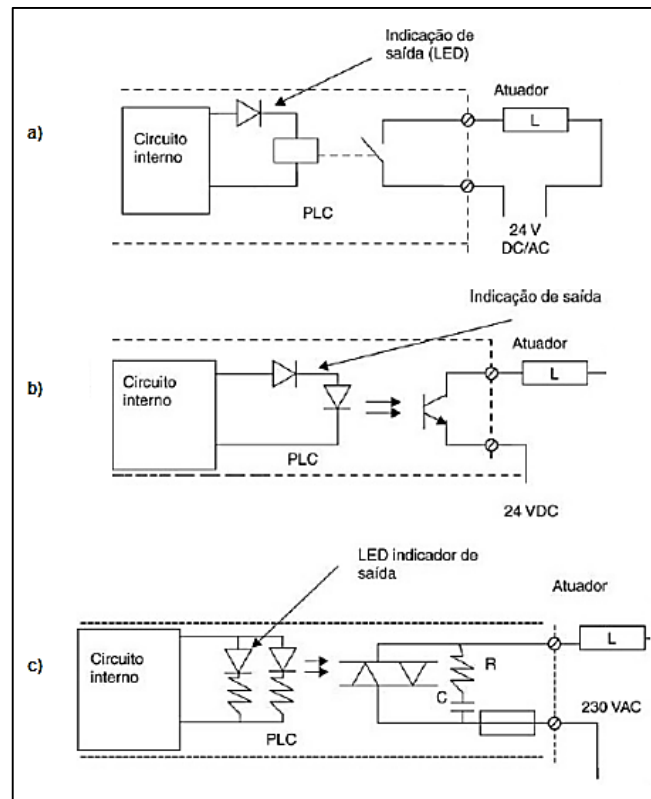
A saída digital a relé é utilizada para acionar atuadores que funcionem tanto em CC quanto em CA, e é compatível com uma ampla faixa de valores de tensão. Entretanto, por ser uma saída eletromecânica, apresenta vida útil menor e maior tempo de resposta se comparado aos semicondutores. A bobina do relé é comandada pelo CLP, que é capaz de energizá-la. Já o atuador fica conectado nos terminais do relé e, conseqüentemente, pode ter seu circuito aberto ou fechado. Nesse caso, não há necessidade de optoacoplador.

A saída digital a transistor possui maior vida útil e resposta mais rápida, entretanto suporta pequenos valores de corrente e é compatível apenas com atuadores que funcionem com tensões contínuas, geralmente de 12V, 24V e 48V. A CPU do CLP envia um sinal para o optoacoplador, e o fototransistor que passa a conduzir corrente no circuito do atuador.

A saída digital a TRIAC possui as mesmas vantagens que a saída a transistor, e é indicada para atuadores que operem com tensão alternada de 115V e 230V. Também possuem pequena capacidade de condução de corrente e opera de forma similar.

Observa-se na figura 26 uma ilustração dos três tipos de saídas digitais explicadas nos parágrafos acima.

Figura 26 – Tipos de saídas digitais a) relé b) transistor e c) TRIAC.



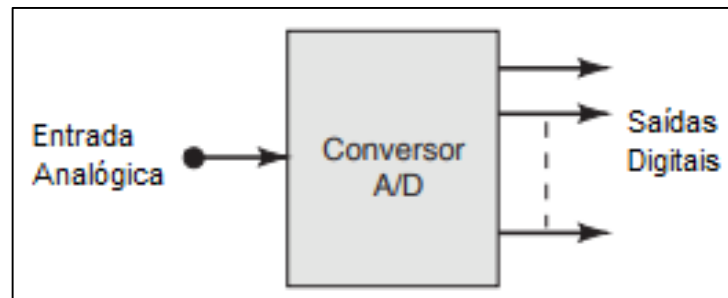
Fonte: PRUDENTE; Francesco, 2011 (adaptado).

Os módulos de entrada analógicos são responsáveis por receber sinais analógicos provenientes dos transdutores e convertê-los em sinais digitais compatíveis com a CPU do controlador. Esses cartões de entrada são projetados para receber valores de tensão contínua geralmente na faixa de 0 a 5V e também valores de corrente contínua na faixa de 4 a 20mA.

Nesse caso, os transdutores são dispositivos que transformam grandezas físicas em elétricas de forma proporcional. Na hipótese de o sinal de saída ser de corrente, 4mA corresponderia a 0% da grandeza física e 20mA equivaleria a 100%. Pode-se citar como exemplo os termopares e encoders.

Quando o sinal elétrico analógico chega ao módulo de entrada, o mesmo é transformado por um conversor A/D. Este entrega em sua saída um sinal digital cuja precisão depende do seu número de bits. Verifica-se na figura 27 uma ilustração de um conversor A/D.

Figura 27 – Conversor A/D.

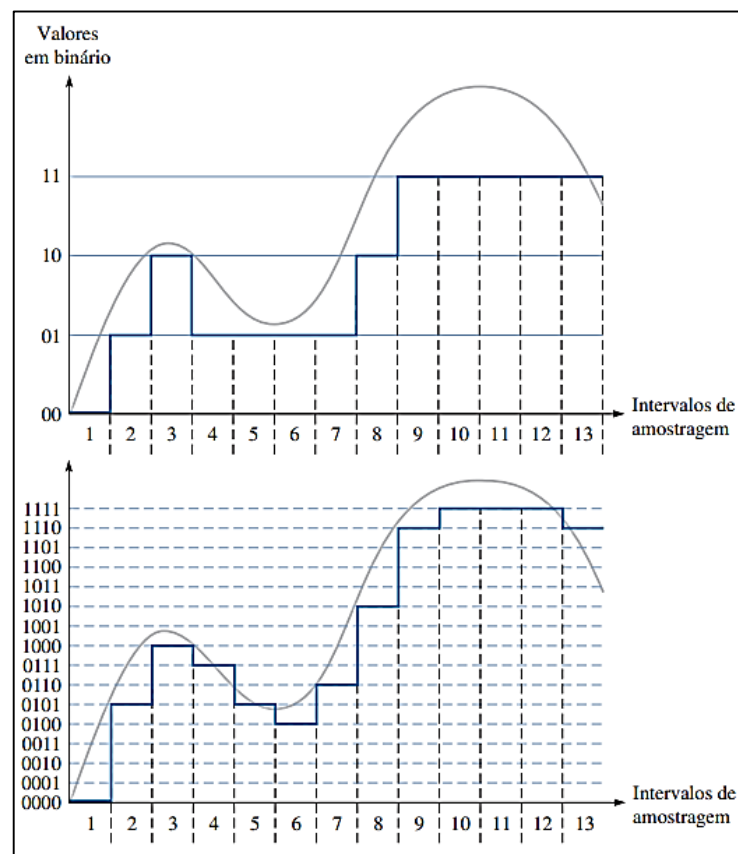


Fonte: TOCCI; Ronald, 2011 (adaptado).

De acordo com Tocci, a quantidade de sinais digitais que os conversores A/D podem formar é dada por 2^N , onde “N” é seu número de bits. Quanto maior essa quantidade, menor será o erro de conversão.

Observa-se na figura 28 os sinais digitais resultantes de conversores A/D de 2 bits e de 4 bits (gráficos de cima e de baixo, respectivamente). O conversor de 2 bits consegue formar quatro valores de sinal digital, enquanto o conversor de 4 bits forma dezesseis e representa o sinal analógico com mais fidelidade.

Figura 28 – Sinais digitalizados.

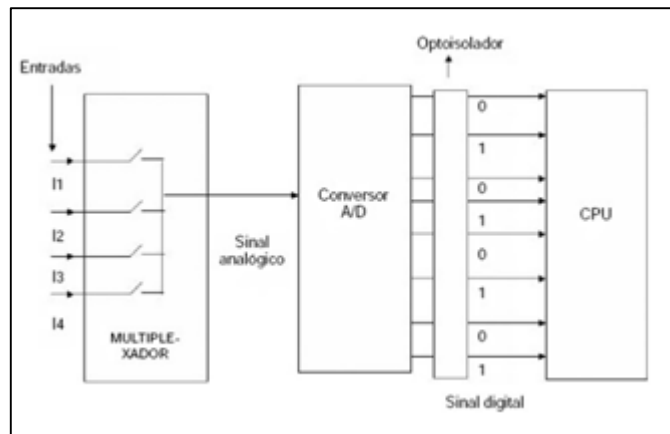


Fonte: FLOYD; Thomas, 2007.

Com objetivo de reduzir custos, os cartões de entrada analógica geralmente são fabricados com apenas um conversor A/D, o qual é compartilhado entre seus pontos de entrada por meio da utilização de um multiplexador. Este dispositivo opera como chave e tem a função de comutar a conexão dos sinais de entrada com o conversor.

A forma de operação explicada no parágrafo anterior tem como ponto negativo o atraso na obtenção dos dados, visto que as entradas analógicas só podem ser lidas uma por vez. A figura 29 ilustra o funcionamento de multiplexador de 4 entradas ligado a um conversor A/D de 8 bits, que comunica com a CPU por meio de um optoacoplador.

Figura 29 – Multiplexador conectado ao conversor A/D.



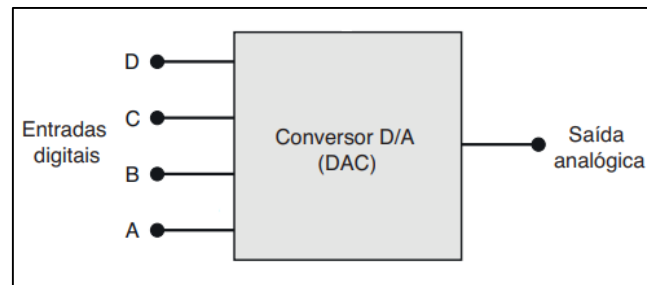
Fonte: PRUDENTE; Francesco, 2011.

Por fim, os módulos de saída analógicos funcionam de forma inversa ao anterior. São responsáveis por converter sinais digitais provenientes da CPU em analógicos. Nesse caso, utilizam conversores digital analógico (D/A).

De acordo com Tocci (2011), a conversão D/A é o processo no qual um valor representado por um código digital é convertido em corrente ou tensão proporcionalmente ao valor digital.

Dessa forma, a CPU envia um valor em binário a ser utilizado na atuação de um elemento. O conversor D/A transforma esses bits em um sinal analógico que é recebido pelo atuador. A depender do valor desse sinal analógico, uma válvula proporcional pode assumir valores intermediários entre a abertura completa e o fechamento total, por exemplo. A figura 30 representa um conversor D/A de 4 bits.

Figura 30 – Conversor D/A.

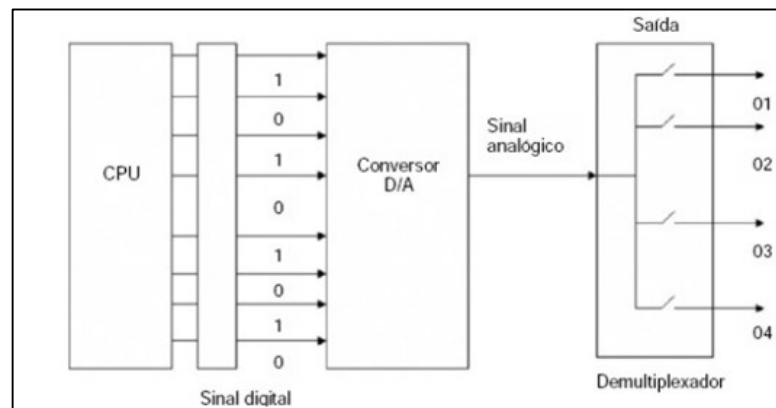


Fonte: TOCCI; Ronald, 2011 (adaptado).

De forma semelhante à anterior, cada módulo de saída analógica geralmente possuem apenas um conversor D/A para reduzir custos. Entretanto, como o objetivo é fazer com que um conversor envie sinais para diferentes saídas, utiliza-se um demultiplexador. Este dispositivo é responsável por comutar a conexão do conversor D/A com os atuadores, um por vez.

A figura 31 ilustra o funcionamento de um módulo de saída analógica com um conversor D/A de 8 bits, um demultiplexador de 4 saídas e um optoacoplador.

Figura 31 – Demultiplexador conectado ao conversor D/A.



Fonte: PRUDENTE; Francesco, 2011.

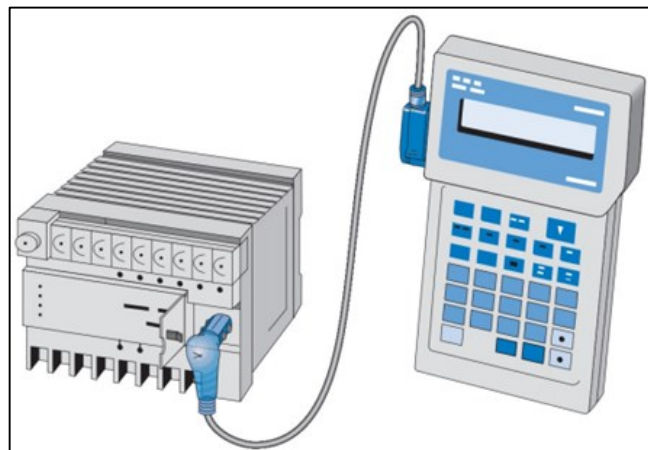
As faixas de valores de tensão e corrente utilizadas pelos módulos de saída analógica são os mesmos dos módulos de entrada analógica, normalmente de 0 a 5V e de 4 a 20mA.

4.5 Dispositivos de Programação

De acordo com Petruzella (2014), os dispositivos de programação são terminais necessários para programar, modificar e encontrar defeitos no programa dos CLPs. Existem diversos equipamentos com essa finalidade, sendo o programador portátil o mais simples deles.

Os programadores portáteis possuem como vantagem o menor custo e a facilidade de utilização. São compostos de um visor de cristal líquido limitado, capaz de mostrar poucas instruções do código, e teclas para navegação e escrita. Geralmente são projetados para funcionar com uma determinada família de CLPs, de um mesmo fabricante. A figura 32 ilustra esse dispositivo conectado a um controlador.

Figura 32 – Programador portátil.

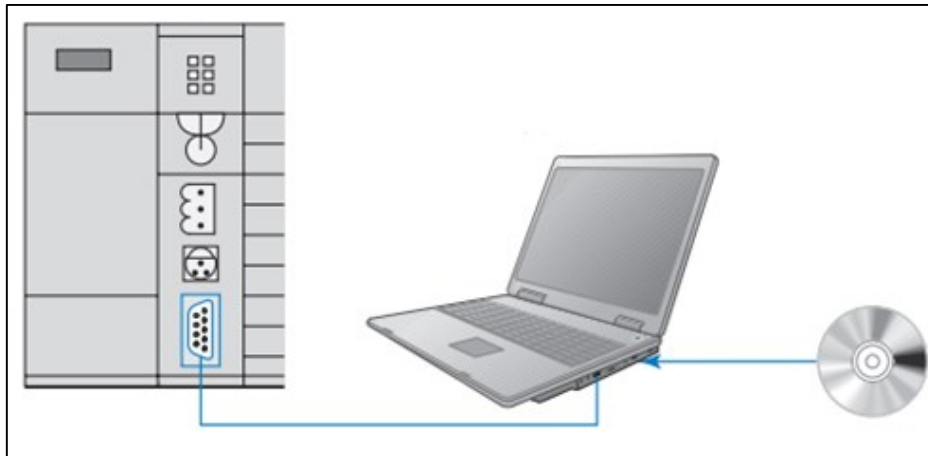


Fonte: PETRUZELLA; Frank, 2014.

Segundo Natale (2008), o dispositivo explicado no parágrafo anterior é atualmente substituído por um PC convencional provido de *softwares* específicos para programar o CLP. Esses *softwares*, também conhecidos como ambientes de programação, são elaborados pelo fabricante do controlador e disponibilizados através de CDs ou DVDs de instalação (figura 33).

Por meio do PC é possível editar o programa do usuário, realizar monitoramentos, diagnósticos de falha, verificação de defeitos e gerar relatórios. Um ambiente de programação elaborado por um determinado fabricante normalmente não é compatível com controladores de outras marcas.

Figura 33 – Utilização de um PC como dispositivo de programação.



Fonte: PETRUZELLA; Frank, 2014.

É importante mencionar também as interfaces homem-máquina (IHM) presentes em muitos CLPs. Trata-se de um periférico que é ligado ao controlador e permite ao usuário visualizar o andamento do programa, as variáveis internas e o estado dos dispositivos de entrada e saída através de uma tela, a qual pode ser sensível ao toque. Em alguns casos também é possível programar o CLP por meio de sua IHM. Observa-se na figura 34 uma ilustração dessa interface.

Figura 34 – IHM Schneider HMIST6200 para CLP.



Fonte: Página de Produtos Schneider.

4.6 Modo de Operação

Os controladores lógicos programáveis possuem diferentes modos de operação, os quais devem ser conhecidos pelo programador para que possam ser realizadas configurações e manutenções no dispositivo. A forma de apresentação desses estados de funcionamento varia conforme o fabricante, entretanto o conteúdo é parecido.

Geralmente os CLPs podem estar nos modos de programação ou execução, e a alternância entre um e outro normalmente é realizada por meio de botões na interface ou através de *software*. Os controladores são capazes de assumir também o estado de falha.

O modo de programação, muitas vezes denominado *prog* ou *stop*, é aquele em que o controlador não executa nenhum código e aguarda alterações em sua configuração. É utilizada para modificar o programa do usuário enquanto o dispositivo está fora de operação.

O envio de um programa elaborado em um terminal de programação para o CLP é conhecido como *download*. De forma similar, a transferência de um programa que esteja armazenado na memória do CLP para o terminal é denominada de *upload*.

O modo de execução, também conhecido como *run*, é aquele em que o controlador executa de forma cíclica o programa do usuário contido em sua memória. De acordo com Franchi, quando o CLP é comutado para esse estado, o mesmo realiza três operações iniciais.

A primeira operação consiste em limpar a parte da memória que armazena os dados referentes às entradas e saídas, a segunda é um teste de funcionamento da RAM e, por fim, a terceira operação verifica se o programa do usuário é capaz de ser executado.

Caso seja constatado algum problema nas operações mencionadas, o controlador entra em estado de falha, sinaliza através de algum LED e interrompe seu funcionamento. Se não forem encontrados erros, o controlador inicia a leitura sequencial das instruções contidas no programa do usuário. Esse fenômeno é denominado de varredura ou *scan*.

Na etapa inicial do processo de *scan*, a CPU do controlador endereça as entradas e saídas. Em seguida é feita uma checagem do estado dos sensores e as

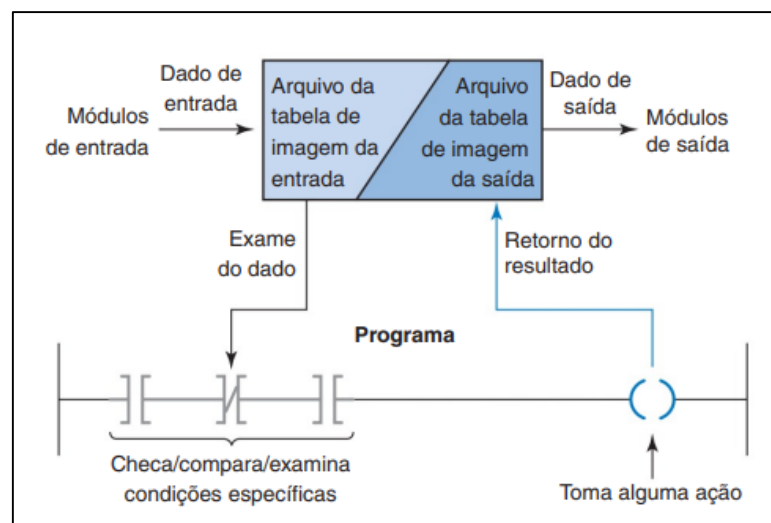
informações são armazenadas, na forma de bits, em um local da memória chamado de tabela de imagem das entradas.

As informações contidas nessa tabela são utilizadas pela CPU ao longo da execução do programa do usuário. O controlador não consulta as entradas diretamente, e sim as informações sobre as mesmas contidas na memória.

À medida que as instruções do programa são executadas, obtém-se resultados que devem ser entregues às saídas. Esses resultados são armazenados em outra região da memória denominada tabela de imagem das saídas. Caso seja necessário consultar as saídas durante a execução do programa, a CPU irá trabalhar com os dados armazenados nessa tabela.

Com o término da execução das instruções, o preenchimento da tabela de imagem das saídas estará completo. Em seguida, os valores armazenados são transferidos às saídas e o ciclo de varredura é finalizado. A figura 35 representa o fluxo dos dados ao longo do processo.

Figura 35 – Fluxo de dados durante a varredura.



Fonte: PETRUZELLA; Frank, 2014.

De acordo com Franchi, o tempo para execução de um ciclo é conhecido como *scan time* e depende de fatores como tamanho do código, *clock* e arquitetura do microprocessador. Caso o programa seja extenso e o controlador tenha pouco poder de processamento, o tempo de varredura será grande.

Segundo Prudente, nessa situação é possível que um sensor mude de estado e retorne ao estado anterior dentro de um mesmo ciclo, sem que o CLP perceba. Tal acontecimento pode acarretar problemas e prejudicar o controle.

Para evitar esse inconveniente, muitos CLPs possuem um *timer* conhecido como *watch dog* (cão de guarda), cuja função é verificar o tempo de varredura. Esse dispositivo geralmente fica ajustado em 150ms e é capaz de interromper o ciclo de funcionamento caso o tempo seja extrapolado. Nessa hipótese, o controlador acusa falha em seus indicadores.

Se o *scan time* estiver dentro da normalidade, o processo é reiniciado a partir da primeira instrução do programa do usuário. Observa-se na figura 36 um fluxograma representado o funcionamento de um controlador no modo de execução.

Figura 36 – Fluxograma do modo de execução do CLP.



Fonte: FRANCHI; Claiton Moro, 2008.

Com as informações do parágrafo anterior, encerra-se a explicação sobre o modo de execução. Entretanto, para concluir a teoria sobre o tema, é importante ilustrar como os controladores indicam seus estados de funcionamento. Nesse intuito, será feita a análise do CLP Rockwell Micro820 apresentado na figura 37.

Esse modelo de controlador possui apenas uma luz indicativa para os modos de execução e programação, denominada *run*. Já a condição de falha é indicada por uma luz exclusiva chamada *fault*.

Segundo o manual do fabricante, caso *run* esteja aceso na cor verde de forma contínua o controlador se encontra em modo de execução. Entretanto, se estiver piscando, o controlador está em modo de programação. Já *fault* pisca na cor vermelha caso haja uma falha recuperável, e fica acesa continuamente caso seja uma falha irreversível.

Figura 37 – Indicadores do CLP Rockwell Micro820



Fonte: Documentação Controlador Rockwell Micro820 (adaptado).

4.7 Programação

Até o início da década de 1990, as empresas enfrentavam dificuldades devido à inexistência de normas acerca dos CLPs. Nessa época havia inúmeras variantes de linguagens de programação diferentes entre si, e os dispositivos de cada fabricante possuíam regras distintas de funcionamento.

Devido à isso, as habilidades desenvolvidas pelos funcionários não podiam ser reaproveitadas caso houvesse a troca de um modelo de controlador por outro, o que implicava em excessiva perda tempo e dinheiro com treinamento por parte das organizações.

Nesse contexto foi elaborada a norma IEC 61131, que visa uniformizar os procedimentos adotados pelos diversos fabricantes. Sua terceira parte, a IEC 61131-3, trata sobre programação e estabeleceu cinco linguagens-padrão para os CLPs:

- Lista de Instruções (IL – *Instruction List*);
- Texto Estruturado (ST – *Structured Text*);
- Linguagem *Ladder* (LD – *Ladder Diagram*);
- Diagrama de Blocos Funcionais (FBD – *Function Block Diagram*); e
- Sequenciamento Gráfico de Funções – SFC (*Sequential Function Chart*).

As linguagens-padrão podem ser divididas em textuais e gráficas, conforme exposto no quadro 4.

Quadro 4 – Divisão das linguagens de programação.

Lista de Instruções (IL)	Linguagem Textual
Texto Estruturado (ST)	
Linguagem <i>Ladder</i> (LD)	Linguagem Gráfica
Diagrama de Blocos Funcionais (FBD)	
Sequenciamento Gráfico de Funções (SFC)	

Fonte: Elaborado pelo autor conforme IEC 61131-3.

A seguir serão apresentadas todas as cinco linguagens estabelecidas pela IEC 61131-3, entretanto o foco será na linguagem *Ladder*, a qual possuirá um subcapítulo dedicado à explicação de seu funcionamento.

A lista de instruções é uma linguagem de programação de baixo nível inspirada em Assembly, baseada em código alfanumérico e com característica sequencial. É pouco intuitiva e requer muito tempo para a pesquisa por falhas no programa. Muitos CLPs não permitem sua utilização.

O texto estruturado, por sua vez, é uma linguagem de alto nível, assim como Pascal, Basic e C++. Contém todos os elementos essenciais da programação moderna, tais como instruções condicionais (if, then, else) e comandos de iteração (for, while, repeat). Entre as linguagens textuais é a mais completa, e é muito utilizada para definir blocos funcionais complexos.

A linguagem *Ladder* é a mais utilizada, sendo encontrada em quase todos os controladores da geração atual. Trata-se de uma representação gráfica que imita a lógica a relés para a realização de circuitos de comando de acionamentos. É composta por degraus lógicos, e por esse motivo recebe o nome de *Ladder*, que do inglês significa "escada". O esquema é composto por duas barras verticais para alimentação e barras horizontais (degraus) com símbolos de contatos e bobinas.

O diagrama de blocos funcionais é uma linguagem gráfica de programação que é composta por blocos interligados utilizando elementos da eletrônica digital, tais como portas lógicas *and*, *or*, *not*, entre outras. Esses blocos geralmente são construídos utilizando a linguagem de texto estruturado.

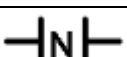
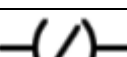
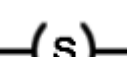
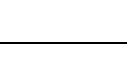
Por fim, o sequenciamento gráfico de funções é uma linguagem que parte da ideia de que qualquer sistema pode ser caracterizado por uma sequencialidade de eventos. Dessa forma, permite a descrições de ações sequenciais, paralelas e alternativas. É constituída por um conjunto de etapas separadas por transições. Em cada etapa há um conjunto de ações a serem executadas e, para que ocorra a transição, é feita uma verificação se as condições foram satisfeitas. É muito similar a um fluxograma.

4.8 Linguagem Ladder

Conforme apresentado anteriormente, a linguagem *Ladder* é uma representação gráfica composta por um conjunto de instruções com símbolos de contatos e bobinas padronizados pela IEC 61131-3. Devido à sua semelhança com diagramas elétricos convencionais, é de fácil compreensão por parte dos profissionais do setor elétrico.

Cada símbolo utilizado nessa linguagem é associado a uma variável contida na memória do controlador, a qual está relacionada com os dispositivos de entrada, de saída ou com a área destinada a armazenar dados de funções especiais. O quadro 5 apresenta os símbolos e suas definições.

Quadro 5 – Simbologia Ladder.

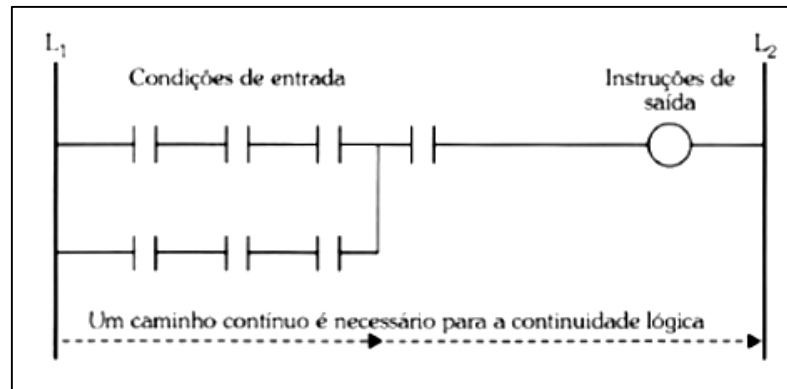
Símbolo	Nome	Descrição
	Contato Normalmente Aberto (NA)	Conduz se a variável booleana associada for “1”
	Contato Normalmente Fechado (NF)	Conduz se a variável booleana associada for “0”
	Contato de Transição Positiva	Conduz até o fim do ciclo de <i>scan</i> ao detectar mudança da variável associada de “0” para “1”
	Contato de Transição Negativa	Conduz até o fim do ciclo de <i>scan</i> ao detectar mudança da variável associada de “1” para “0”
	Bobina	Quando energizada, atribui “1” à variável booleana associada
	Bobina Negada	Quando energizada, atribui “0” à variável booleana associada
	Bobina Set	Uma vez energizada, atribui “1” à variável booleana associada permanentemente, até que esta receba sinal de uma bobina <i>reset</i>
	Bobina Reset	Uma vez energizada, atribui “0” à variável booleana associada permanentemente, até que esta receba sinal de uma bobina <i>set</i>
	Bobina de Detecção Positiva	Quando detecta o evento de energização, atribui “1” à variável associada até o fim do ciclo de <i>scan</i>
	Bobina de Detecção Negativa	Quando detecta o evento de desenergização, atribui “1” à variável associada até o fim do ciclo de <i>scan</i>

Fonte: Elaborado pelo autor com base na IEC 61131-3 e NIST.

Os símbolos são dispostos em série e/ou em paralelo ao longo linhas horizontais (degraus) para formar uma determinada lógica de acionamento. Os degraus são conectados entre duas barras verticais (trilhos), que representam a alimentação elétrica. Os trilhos da esquerda e da direita simbolizam os polos positivo e negativo, respectivamente.

O conjunto de contatos presentes em um degrau impõem a condição de entrada, e são os primeiros elementos à esquerda (zona de *input*). As instruções de saída, representadas pelas bobinas e blocos de funções, são os últimos elementos à direita (zona de *output*). A figura 38 ilustra a estrutura típica de um degrau.

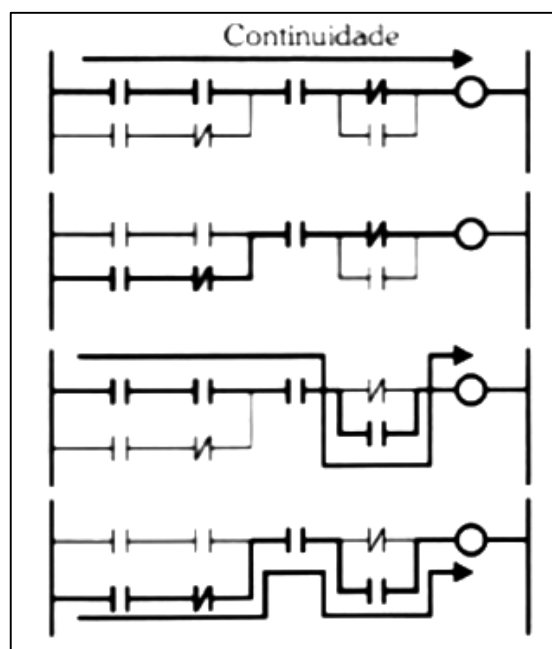
Figura 38 – Estrutura do código *ladder*.



Fonte: FRANCHI; Claiton Moro, 2008.

Quando pelo menos uma combinação de contatos permite continuidade entre os trilhos, a condição de entrada é satisfeita. Nessa situação, flui uma corrente imaginária que energiza a instrução de saída. Na figura 39, observa-se quatro possibilidades de continuidade em uma lógica desenvolvida em Ladder.

Figura 39 – Caminhos possíveis de continuidade.



Fonte: FRANCHI; Claiton Moro, 2008.

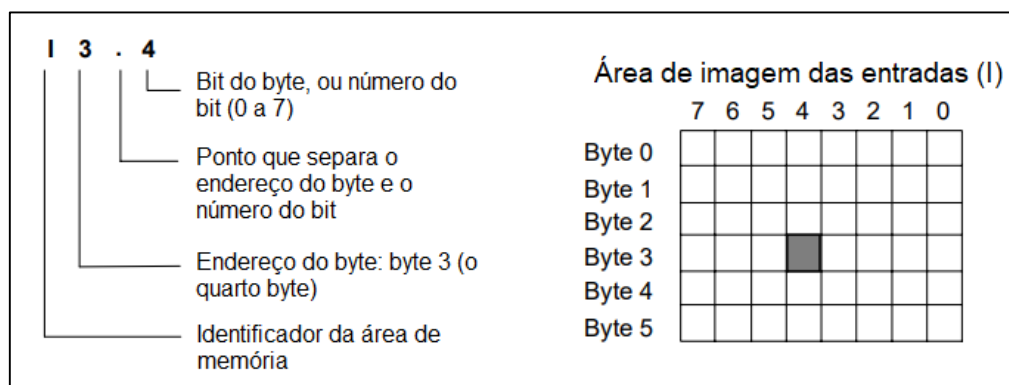
De acordo com o que já foi explicado, cada dispositivo conectado nos cartões do CLP é endereçado a um local de sua memória conhecido como tabela de imagens das entradas e das saídas. Essa área da memória, juntamente com a parte que armazena valores intermediários, precisa ser acessada pelo programa através de seus códigos de endereço.

A notação de um endereço na memória varia conforme o fabricante e o modelo do controlador. Para codificar a região das entradas normalmente utiliza-se “I” de “*Input*”. Já para a área das saídas, geralmente é empregado “Q” de “*Quit*” ou “O” de “*Output*”. Quando se trata da região da memória destinada a armazenar valores intermediários, como um relé virtual, alguns fabricantes denominam com a letra “M” de “*Memory*”.

A maioria dos fabricantes utilizam *words*, *bytes* e variáveis de único *bit* no endereçamento das informações. À frente da letra que designa a região da memória normalmente encontra-se números que podem indicar o cartão de E/S ou o terminal de conexão.

O modelo de controlador Siemens S7-200, por exemplo, emprega as letras “I”, “Q” e “M” para as regiões das entradas, saídas e memória interna, respectivamente. Nessas áreas, as informações são armazenadas em *bytes*, endereçados por um número. À frente deste número, há um ponto (“.”) seguido por um segundo número, que identifica o *bit* dentro do *byte*. A figura 40 ilustra esse endereçamento.

Figura 40 – Endereçamento de memória no Siemens S7-200.

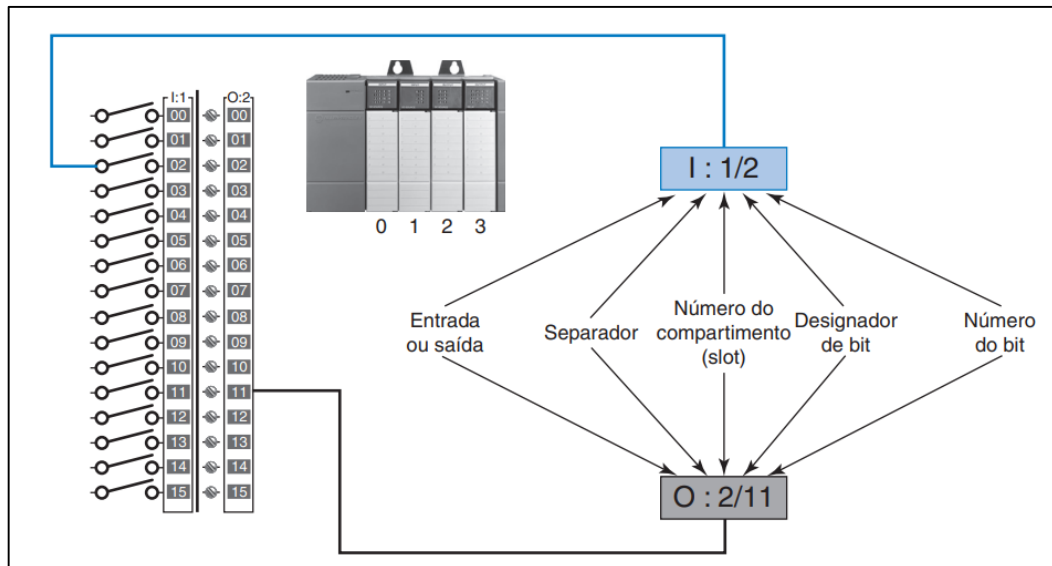


Fonte: Manual do Siemens Simatic S7-200 (adaptado).

Já a família de controladores SLC da fabricante Rockwell endereça as regiões de entrada, saída e memória interna respectivamente com as letras “I”, “O” e “B3”. À frente da primeira letra há um separador de dois pontos (“.”), seguido de um número

que identifica o compartimento em que se encontra o dado desejado. Esses compartimentos estão alocados na memória em forma de *words* (16 bits). Em sequência, há uma barra ("/") seguida pelo número do terminal de conexão, que é o número do *bit* dentro do *word* (figura 41).

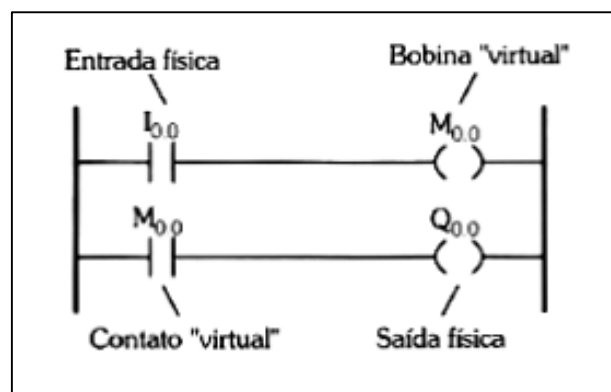
Figura 41 – Endereçamento de memória na família de CLPs Rockwell SLC.



Fonte: PETRUZELLA; Frank, 2014.

Os relés auxiliares, também conhecidos como relés internos ou virtuais, são contatos e bobinas utilizados para armazenar bits temporariamente em um local conhecido como memória interna e auxiliar na lógica da programação. São capazes de acionar outros elementos do diagrama *ladder*, porém não são vinculados aos cartões de E/S. Vide figura 42.

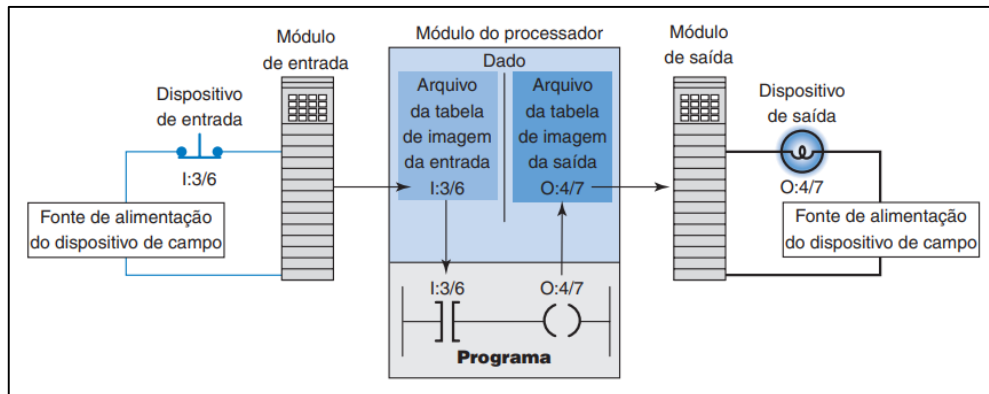
Figura 42 – Contato e bobina virtuais.



Fonte: FRANCHI; Claiton Moro, 2008.

Uma vez conhecidos os endereços de memória desejados, o programador deve associá-los aos elementos do diagrama *ladder*, de forma a construir uma lógica que atenda aos requisitos do sistema controlado. Observa-se na figura 43 um exemplo de aplicação que possui apenas um degrau de programa.

Figura 43 – Etapas de funcionamento com linguagem *ladder*.



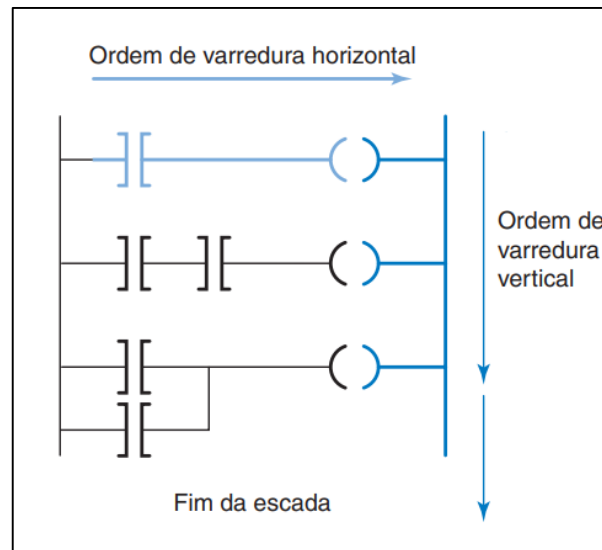
Fonte: PETRUZELLA; Frank, 2014.

Na imagem, verifica-se que uma chave fechada transmite sinal lógico positivo para um ponto de entrada físico, endereçado no local de memória “I:3/6”. Dentro do programa, o contato NA associado a essa endereço muda de estado, passando a conduzir. Devido à isso, a bobina do diagrama é energizada, e transfere sinal lógico positivo para sua posição de memória associada “O:4/7”. Ao final do processo, o valor positivo presente no endereço “O:4/7” é transferido para o ponto de saída físico endereçado, acionando um dispositivo no campo.

A depender da complexidade do sistema controlado, os diagramas *ladder* podem possuir vários degraus. Nesse caso, os mesmos são lidos pela CPU da esquerda para a direita e de cima para baixo, a cada ciclo de varredura, conforme apresentado na figura 44.

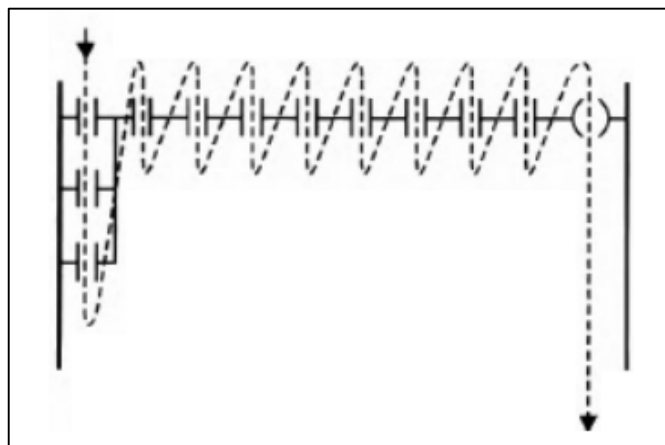
Ademais, dentro de um mesmo degrau que possua elementos em paralelo, a leitura também é realizada de cima para baixo, e não apenas da esquerda para a direita (figura 45).

Figura 44 – Ordem de varredura dos degraus.



Fonte: PETRUZELLA; Frank, 2014.

Figura 45 – Ordem de varredura em cada degrau.

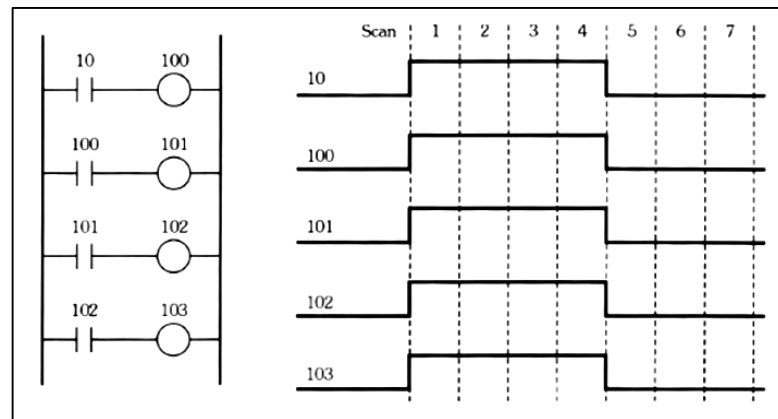


Fonte: FRANCHI; Claiton Moro, 2008 (adaptado).

Vale ressaltar também que todas as mudanças nos cartões de saída provenientes de um ciclo de varredura acontecem simultaneamente. Na figura 46, caso o contato NA de número 10 seja ativado, a bobina 100 será energizada. Em seguida, o contato NA 100 passará a conduzir, e energizará a bobina 101. Após isso, o contato NA 101 será fechado e a bobina 102 acionada, e assim sucessivamente.

Embora as bobinas estejam em degraus diferentes, os dispositivos vinculados a elas são acionados de forma síncrona, conforme gráfico à esquerda. Caso os atuadores associados sejam lâmpadas, por exemplo, todas acenderão no mesmo instante, ao final do ciclo de *scan*.

Figura 46 – Instante de acionamento das bobinas.



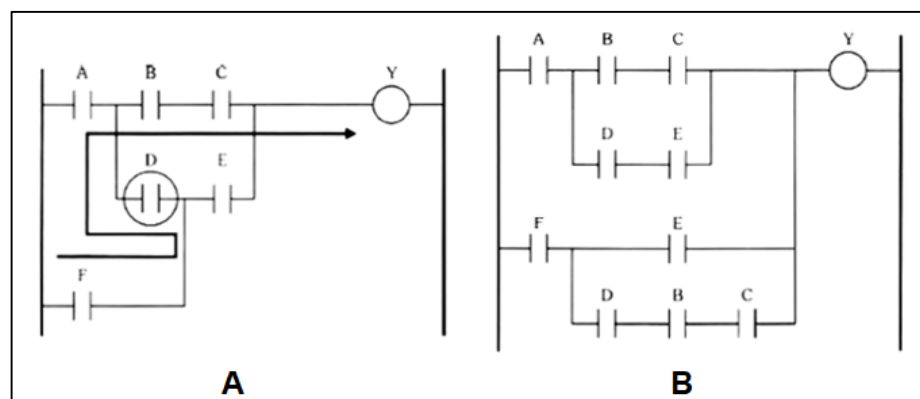
Fonte: FRANCHI; Claiton Moro, 2008.

As fabricantes de controladores geralmente não admitem que haja fluxo de corrente da direita para a esquerda nos diagramas em *ladder*. Devido a isso, alguns circuitos com relés eletromecânicos precisam ser adaptados para funcionarem nessa linguagem gráfica.

No diagrama apresentado no item A da figura 47, caso os contatos F, D, B e C passem a conduzir, a corrente fluiria em sentido reverso por um pequeno trecho e ativaria a bobina Y. Do ponto de vista dos circuitos elétricos, não há problema. Entretanto, no caso da linguagem *ladder*, a bobina não seria energizada.

Para que o diagrama funcione, é necessário que seja feita uma adaptação que permita a corrente fluir entre os mesmos contatos sem assumir o sentido contrário. Uma solução seria repetir os contatos B, C, D e E, criando um caminho paralelo, conforme mostrado no item B da figura. A repetição de contatos é perfeitamente possível nessa linguagem, podendo ser utilizada conforme a necessidade.

Figura 47 – Correção do fluxo reverso com repetição de contatos.

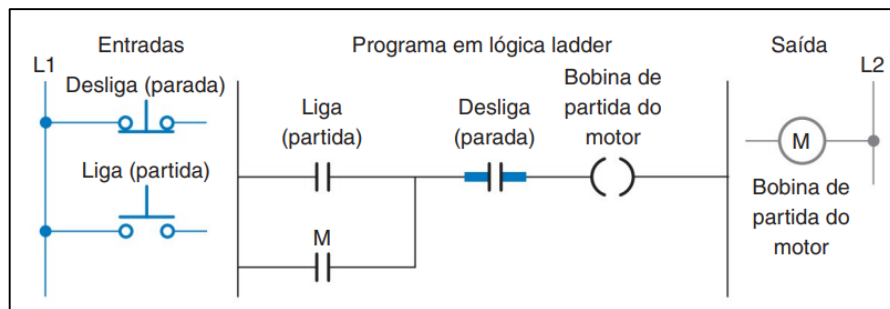


Fonte: FRANCHI; Claiton Moro, 2008 (adaptado).

De forma análoga à lógica com relés, os circuitos com selo são muito utilizados nos diagramas *ladder*. Seu objetivo é manter o caminho de condução em um degrau após um determinado contato se abrir. Trata-se de um segundo contato que é posicionado em paralelo, vinculado à bobina que se deseja manter energizada.

A figura 48 ilustra uma situação em que se deseja partir um motor através de uma chave NA que, ao ser pressionada, fecha o contato associado e energiza a bobina de partida. Uma vez energizada, o contato de selo “M” passa a conduzir, e mesmo que a chave pressionada retorne ao seu estado aberto, a bobina continuará acionada.

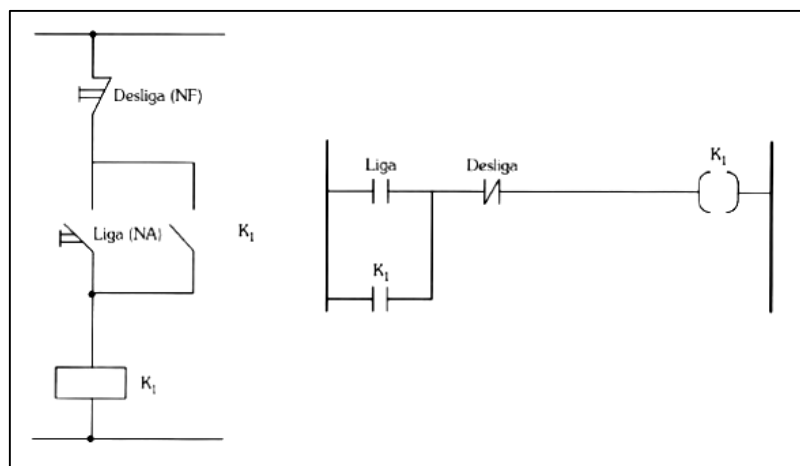
Figura 48 – Diagrama *ladder* com contato de selo.



Fonte: PETRUZELLA; Frank, 2014.

Os diagramas em *ladder* podem ser criados a partir da conversão de diagramas elétricos. Em muitos casos, basta apenas transformar as colunas dos circuitos de acionamento em linhas. A figura 49 ilustra novamente um circuito de comando de partida de motor (à esquerda) e seu equivalente em *ladder* (à direita).

Figura 49 – Diagrama elétrico simples e seu equivalente em *ladder*.

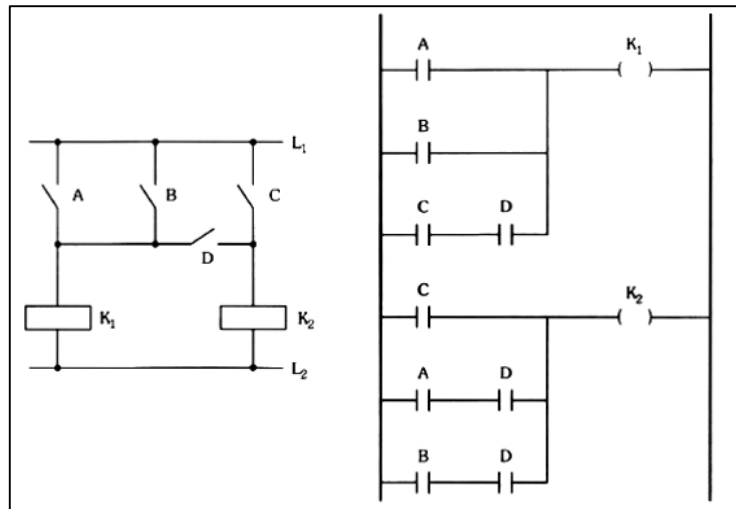


Fonte: FRANCHI; Claiton Moro, 2008 (adaptado).

Entretanto, diagramas elétricos mais complexos não são convertidos com a mesma facilidade. No exemplo à esquerda na figura 50, o contato D faz uma ponte entre os circuitos de comando das bobinas K_1 e o K_2 e já se encontra na posição horizontal.

Para converter esse diagrama elétrico, é preciso observar que a bobina K_1 é acionada pelos contatos A ou B, ou ainda pelo fechamento simultâneo de C+D; e a bobina K_2 é acionada por C ou pela ação simultânea de A+D ou de B+D. Com isso em mente e utilizando a repetição de contatos, é possível elaborar dois degraus em *ladder* que satisfaçam às condições, sendo um degrau para cada bobina, conforme exposto pelo diagrama à direita na figura.

Figura 50 – Diagrama elétrico com contato em ponte e seu equivalente em *ladder*.



Fonte: FRANCHI; Claiton Moro, 2008 (adaptado).

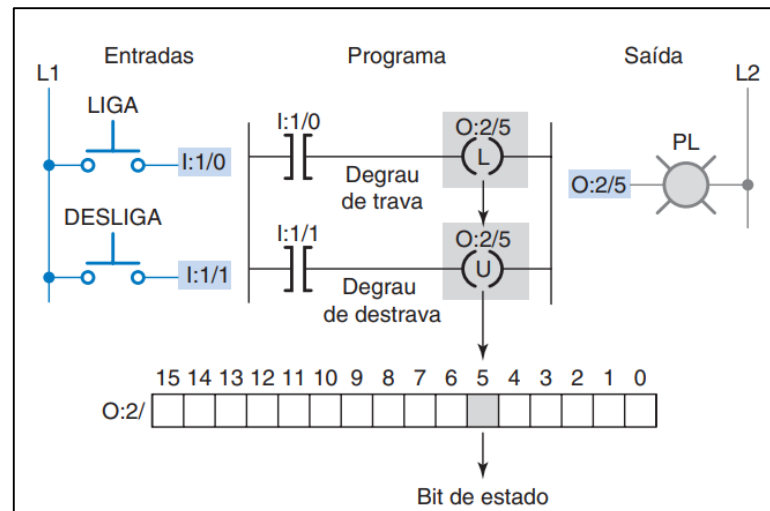
As bobinas *set* e *reset*, também conhecidas como bobinas de trava e destrava ou *latch* e *unlatch*, são instruções retentivas associadas a um determinado endereço de memória. Quando energizadas, mesmo que por um curto período de tempo, gravam um bit na imagem das saídas, o qual permanece salvo após desativadas. Em outras palavras, basta que recebam um degrau de energização para cumprirem sua finalidade.

Quando ativada, a bobina de trava escreve o valor binário "1" e a bobina de destrava o valor binário "0". Possuem diversas aplicações, como o acionamento de uma lâmpada ilustrado na figura 51.

Nesse exemplo, observa-se dois botões pulsantes NA como dispositivos de entrada. Quando botão de ligar é pressionado, seu respectivo contato passa a conduzir no diagrama *ladder*. Dessa forma, a bobina *latch* é ativada temporariamente e transfere sinal lógico "1" para o endereço de saída relacionado à lâmpada PL. Quando o botão retorna ao estado aberto a bobina é desativada, porém a lâmpada continua acesa.

Para que a lâmpada seja apagada, a mesma deve receber a instrução de reset. Isso é possível através do botão de desligar que, de forma análoga, ativa a bobina *unlatch*. Ao ser acionada por um intervalo de tempo, esta escreve o sinal lógico "0" no local de memória vinculado à lâmpada e a apaga.

Figura 51 – Aplicação das bobinas *set* e *reset*.



Fonte: PETRUZELLA; Frank, 2014.

Os contatos de detecção positiva e negativa são aqueles capazes de identificar o evento de mudança de sua variável associada. O primeiro é sensível à borda de subida, ou seja, o momento em que o valor dessa variável muda de "0" para "1". Já o segundo é capaz de detectar a borda de descida, que é quando a variável passa de "1" para "0". Quando o evento necessário acontece, esses contatos conduzem até o final do ciclo de *scan*.

As bobinas de detecção positiva e negativa funcionam de forma parecida. A primeira é capaz de detectar a mudança de seu estado de energização de desligada para ligada. Já a segunda detecta a mudança de seu estado de energização de ligada para desligada.

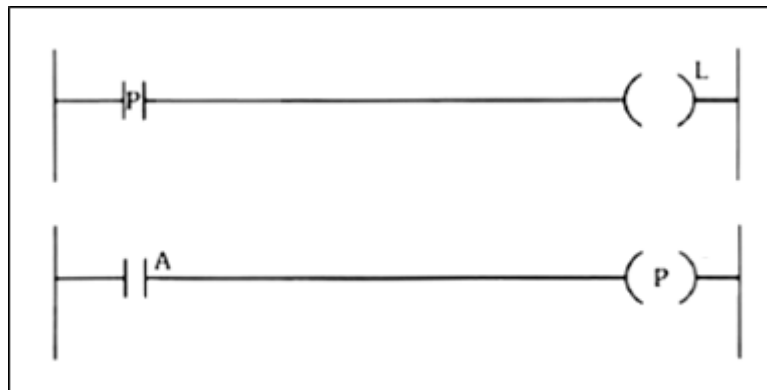
para desligada. Quando o evento necessário acontece, as bobinas atribuem o valor “1” à sua variável associada até o fim do ciclo de *scan*.

Observa-se na figura 52 dois degraus, o de cima com uma aplicação do contato de detecção positiva (contato P) e o de baixo com uma aplicação da bobina de detecção positiva (bobina P).

Na primeira situação, quando a variável associada ao contato P sofrer alteração em seu nível lógico de “0” para “1”, o mesmo irá conduzir até o final do ciclo de *scan*, energizando a bobina L. No ciclo seguinte, mesmo que a variável vinculada ao contato permaneça com nível lógico “1”, este não mais conduzirá, haja vista que não houve detecção de borda de subida. Sendo assim, a bobina L será desligada.

Na segunda situação, quando o contato A passar a conduzir, a bobina P sofrerá alteração em seu estado de energização de “desligado” para “ligado” e atribuirá o valor binário “1” à sua variável associada até o fim do ciclo de *scan*. No ciclo seguinte, mesmo que o contato A permaneça conduzindo, a bobina não irá gravar “1” novamente em sua respectiva variável, uma vez que não foi detectada nenhuma mudança positiva em seu estado de energização.

Figura 52 – Aplicação do contato e da bobina de detecção positiva.

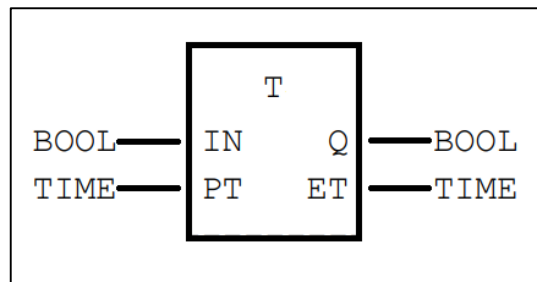


Fonte: FRANCHI; Claiton Moro, 2008 (adaptado).

A instrução em *ladder* mais utilizada, depois dos contatos e bobinas, é o temporizador (*timer*). Trata-se de um recurso que possui a mesma funcionalidade dos relés temporizados eletromecânicos. Esses elementos permitem que o programador configure um valor tempo, o qual começa a ser contado a partir da energização de sua entrada. Quando o tempo decorrido é igual ou superior ao tempo predefinido, o *timer* atua por meio de sua saída.

A norma IEC 61131-3 apresenta o temporizador como um bloco de função com quatro terminais, conforme figura 53. O terminal IN (*Input*) é o local por onde o temporizador é energizado com uma variável booleana, PT (*Preset Time*) é onde se realiza a configuração do tempo, em ET (*Elapsed Time*) é exibido o valor de tempo decorrido e, por fim, Q (*Quit*) é sua saída.

Figura 53 – Bloco do temporizador.

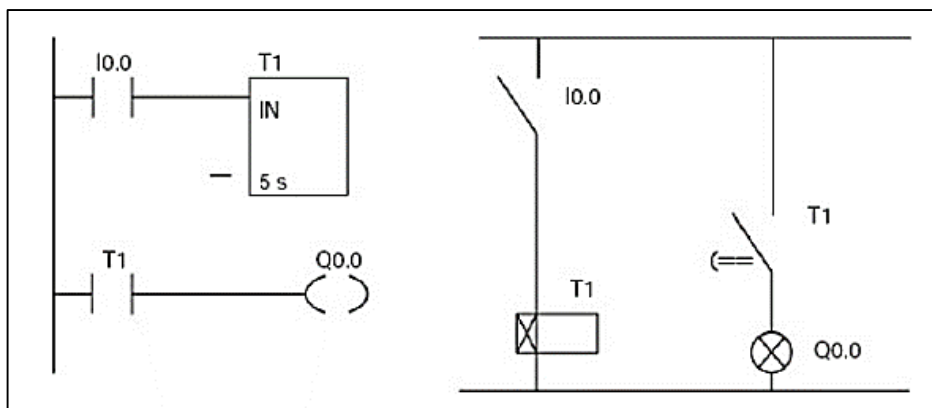


Fonte: Elaborado pelo autor com base na IEC 61131-3.

De acordo com Prudente, em um CLP normalmente há dois ou três tipos de temporizadores disponíveis para utilização. O mais comum é o *timer* de retardo ao ligar (TON – *Timer On Delay*), que é uma função básica. As outras variantes de temporizadores são todas derivadas dessa.

A forma com que o temporizador se apresenta na interface do software de programação varia conforme o fabricante, entretanto funcionam de maneira similar. A figura 54 ilustra um programa *ladder* que contém um *timer* de atraso na ligação e seu diagrama elétrico equivalente.

Figura 54 – Diagrama com *timer* (à esq.) e seu equivalente elétrico (à dir.).

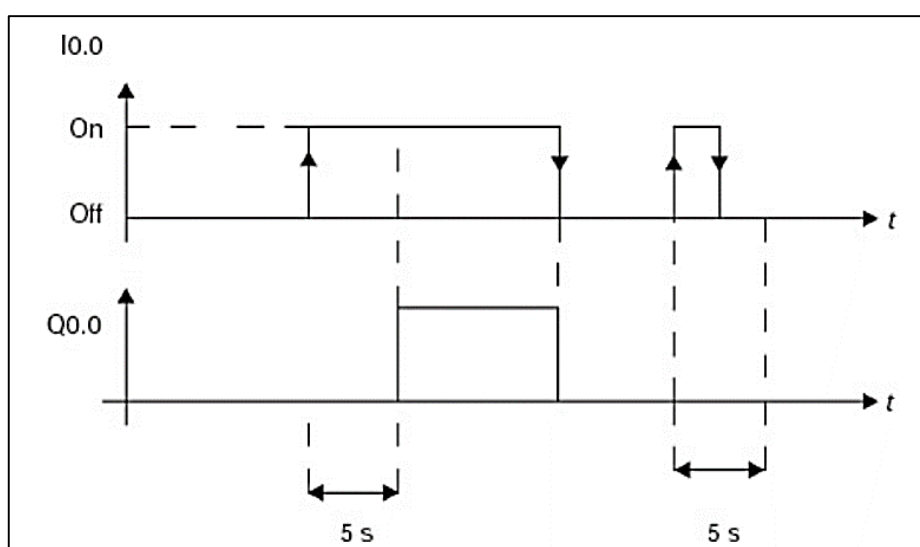


Fonte: PRUDENTE; Francesco, 2011.

Observa-se na figura que o temporizador T_1 está predefinido com tempo de 5 segundos, e sua entrada está sendo controlada pelo contato endereçado em I0.0. Quando este contato passa a conduzir, a contagem do tempo imposto é iniciada. Transcorrido o tempo configurado, o timer ativa o contato T_1 que se fecha e energiza a bobina associada ao endereço Q0.0.

Os gráficos presentes na figura 55 ilustram esses eventos ao longo do tempo. Verifica-se ainda que, caso o contato endereçado em I0.0 seja desativado antes que o timer termine de contabilizar os 5 segundos predefinidos, o contato T_1 não é ativado e o tempo decorrido é zerado.

Figura 55 – Nível lógico da entrada e da saída do TON.



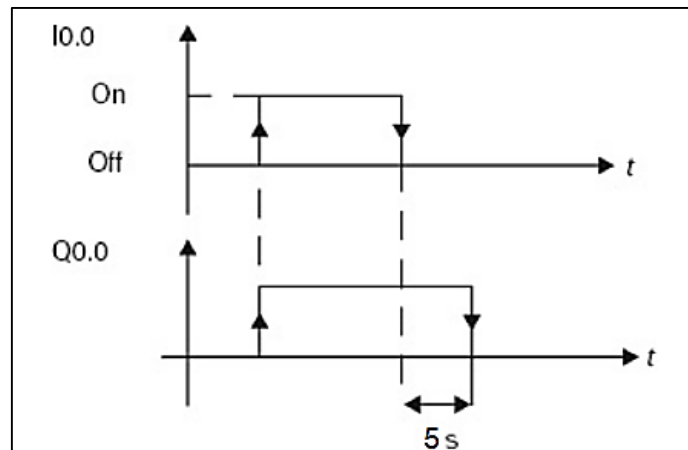
Fonte: PRUDENTE; Francesco, 2011.

O *timer* de atraso no desligamento (*TOF – Timer Off Delay*) funciona de forma muito parecida. Ao invés de temporizar o fechamento do contato T_1 , será temporizada a sua abertura. Para explica-lo, será reaproveitado o conteúdo da figura 54.

Quando a entrada I0.0 é ativada, seu respectivo contato se fecha e energiza o temporizador. Nesse momento, o contato T_1 passa a conduzir e energiza a bobina endereçada em Q0.0. Caso o contato I0.0 se abra, o temporizador iniciará a contagem do valor de tempo predefinido. Transcorrido esse tempo, o contato T_1 se abre e a bobina é desenergizada.

O evento descrito no parágrafo anterior está ilustrado no gráfico da figura 56. Vale ressaltar ainda que, durante a contagem do temporizador, caso o contato endereçado em I0.0 reestabeleça a condução, o tempo decorrido é zerado.

Figura 56 – Nível lógico da entrada e da saída do TOF

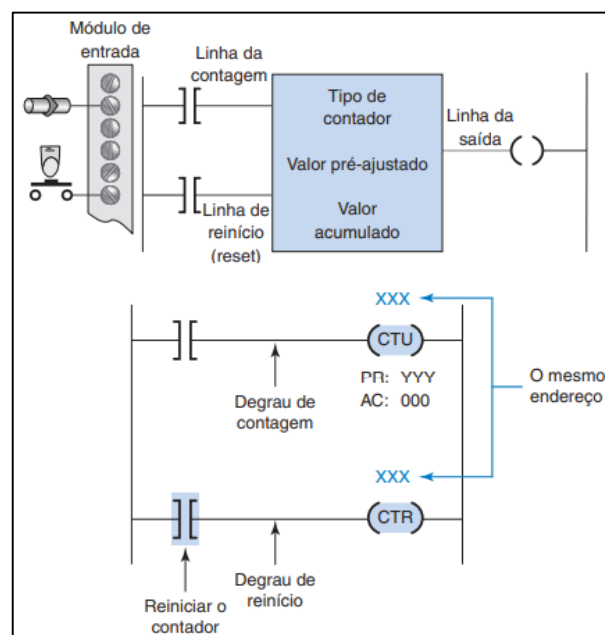


Fonte: PRUDENTE; Francesco, 2011.

Além dos *timers*, os CLPs possuem blocos de grande importância denominados contadores. Estes são responsáveis por realizar a contagem de eventos de todos os tipos, e dependem de pulsos externos ou de programas-fonte para exercerem sua função.

Os contadores geralmente possuem os seguintes parâmetros: tipo, terminal para contagem, terminal de reinício, valor pré-definido, valor acumulado e terminal de saída. São representados nos diagramas *ladder* em forma de bloco ou bobina, conforme ilustrado na figura 57.

Figura 57 – Contador formatado em bloco (cima) e em bobina (baixo)



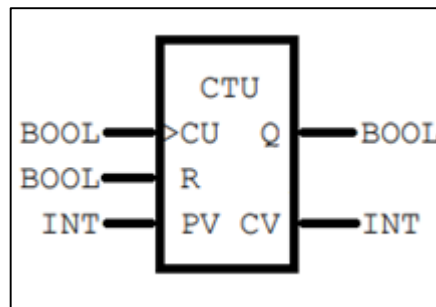
Fonte: PETRUZELLA; Frank, 2014 (adaptado).

Esses dispositivos são projetados para contar, de forma crescente ou decrescente, até que o valor programado pelo usuário seja alcançado, momento em que a saída é acionada. O terminal de reinício é fornecido para restaurar o valor acumulado a um número preestabelecido.

Possuem caráter retentivo, ou seja, não perdem suas informações de contagem quando o processo é desligado por algum motivo. De acordo com a IEC 61131-3, os contadores são divididos em três tipos: crescentes, decrescentes e bidirecionais.

O contador crescente é aquele cuja função é incrementar em uma unidade em seu valor acumulado sempre que sua instrução de entrada mudar de falso para verdadeiro. É apresentado pela IEC 61131-3 na forma de bloco. Vide figura 58.

Figura 58 – Contador crescente.



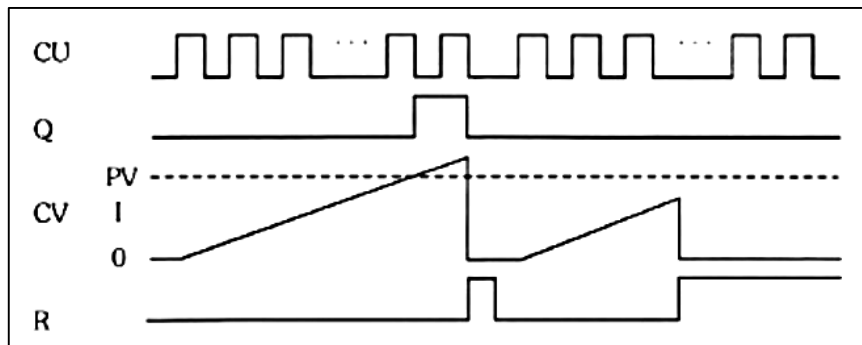
Fonte: Elaborado pelo autor com base na IEC 61131-3.

Os terminais apresentados na figura tem as seguintes definições:

- CU: entrada de contagem crescente (*count up*);
- R: entrada de reinício (reset);
- PV: entrada para o valor pré-definido (*preset value*);
- CV: saída para o valor acumulado (*count value*); e
- Q: saída para atuação no sistema (*quit*).

Observa-se na figura 59 um diagrama de eventos do contador crescente. Sempre que uma borda de subida é recebida no terminal de contagem CU, o valor acumulado em CV aumenta. Quando o valor acumulado ultrapassa o valor pré-definido PV, a saída Q é acionada. Ao receber um sinal positivo no terminal de *reset*, a saída é desligada e os valores de PV zerados.

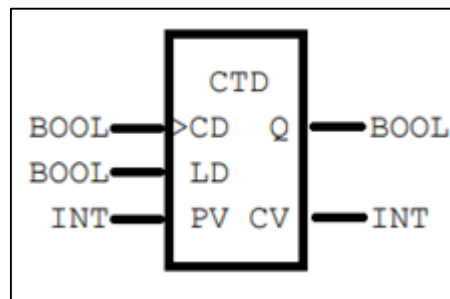
Figura 59 – Diagrama de eventos do contador crescente



Fonte: FRANCHI; Claiton Moro, 2008.

O contador decrescente funciona de forma muito parecida com o anterior. Entretanto, sempre que seu valor de entrada mudar de falso para verdadeiro, o valor acumulado é decrementado em uma unidade. De forma similar, é apresentado pela IEC 61131-3 na forma de bloco, conforme figura 60.

Figura 60 – Contador decrescente.



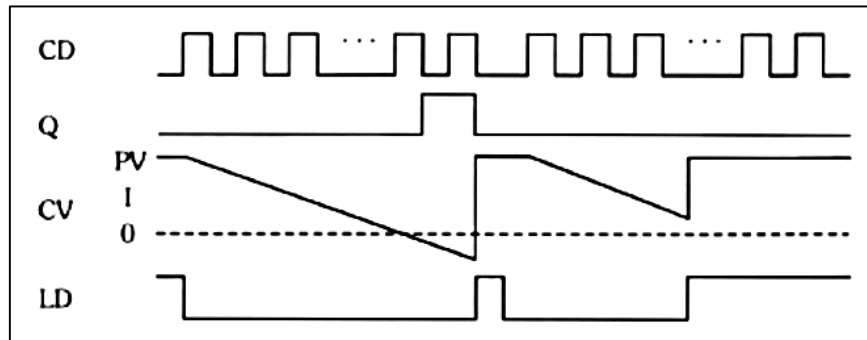
Fonte: Elaborado pelo autor com base na IEC 61131-3.

Se comparado ao contador crescente, as únicas diferenças em seus terminais são as siglas que nomeiam as entradas de contagem e de reinício. Nesse modelo, a primeira é denominada *count down* (CD) ao invés de *count up*, e isso se deve ao fato de se tratar de um contador decrescente. Já a segunda é chamada de *load* (LD) ao invés de *reset*, haja vista que o valor acumulado de contagem não é reiniciado para zero, e sim para um valor pré-definido.

A figura 61 apresenta o diagrama de eventos referente a esse contador. Observa-se que à medida que são identificadas mudanças de “0” para “1” na entrada de contagem CD, o valor acumulado em CV diminui, partindo do valor pré-definido PV até alcançar números negativos. Nesse momento, a saída Q é acionada. Ademais,

caso o terminal de reinício LD seja acionado, a saída é desligada e o CV é reestabelecido ao valor preestabelecido.

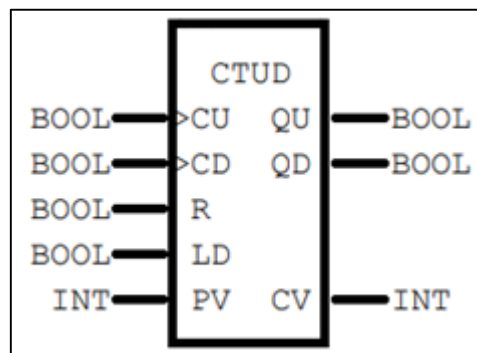
Figura 61 – Diagrama de eventos do contador crescente



Fonte: FRANCHI; Claiton Moro, 2008.

Por fim, o contador bidirecional é a união dos dois tipos apresentados anteriormente. Possui duas entradas de contagem, uma crescente e a outra decrescente, além de uma saída e um reinício para cada. As nomenclaturas são as mesmas já explicadas. É ilustrado pela IEC 61131-3 na forma de um bloco com oito terminais, conforme figura 62.

Figura 62 – Contador bidirecional.



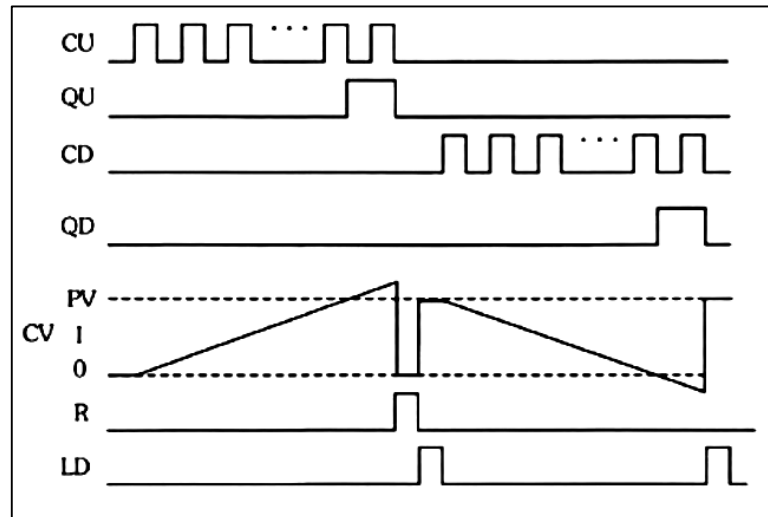
Fonte: Elaborado pelo autor com base na IEC 61131-3.

Nesse dispositivo, o valor acumulado CV pode ser incrementado ou decrementado por meio dos terminais de contagem CU e CD, respectivamente. Caso CV ultrapasse o valor pré-definido, a saída QU é ativada. Entretanto, se o valor acumulado atingir números negativos, a saída QD é ativada.

A entrada R é capaz de zerar o valor acumulado em CV. Já a entrada LD é responsável por igualar CV ao valor pré-definido. Muitos CLPs não possuem esse tipo

de contador, sendo necessário utilizar um modelo crescente e outro decrescente para obter essa função. Observa-se seu diagrama de eventos na figura 63.

Figura 63 – Diagrama de eventos do contador bidirecional.



Fonte: FRANCHI; Claiton Moro, 2008.

Como a IEC 61131-3 não é obrigatória, as fabricantes de CLPs não seguem com muita fidelidade as definições estabelecidas sobre temporizadores e contadores. Cada marca de CLP possui suas particularidades, entretanto esses blocos apresentam funcionamento similar em todos os casos.

4.9 Sistemas Combinacionais

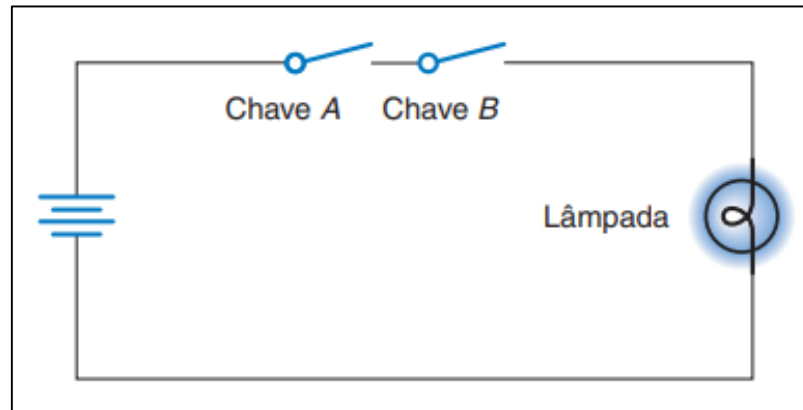
De acordo com Prudente, o maior desafio na configuração de um CLP é a idealização de seu esquema funcional. Para auxiliar nessa etapa, é possível utilizar a álgebra booleana.

A ideia consiste em testar todas as combinações de entrada possíveis a um determinado sistema, registrar os resultados obtidos em uma tabela-verdade e obter sua expressão lógica. Entretanto, o método não serve para modelar sistemas que evoluam em função do tempo ou de eventos externos, ou seja, aqueles que possuem temporizadores ou contadores.

Ao analisar o circuito elétrico da figura 64, por exemplo, observa-se que a energização da lâmpada depende do estado binário das chaves normalmente abertas

A e B. Portanto, trata-se de um sistema lógico com duas variáveis de entrada e uma saída.

Figura 64 – Circuito elétrico com duas variáveis de entrada e uma saída.



Fonte: PETRUZELLA; Frank, 2014.

Segundo Tocci, o número de combinações possíveis em uma situação com N entradas é igual a 2^N . Dessa forma, o esquema elétrico apresentado possui quatro arranjos diferentes.

Considera-se que um contato possui nível lógico “1” quando está acionado, e nível lógico “0” caso contrário. Analogamente o estado da lâmpada é binário, sendo “1” quando acesa e “0” quando apagada.

Nesse exemplo, a saída só será verdadeira quando as entradas A e B também forem. Caso nenhuma ou apenas uma chave seja acionada, a lâmpada não irá acender. Por conseguinte, dentre as quatro possibilidades, apenas uma resultará em nível lógico de saída positivo (quadro 6). Tal comportamento é o mesmo da função lógica *and*, que será apresentada a seguir.

Quadro 6 – Tabela-verdade para o circuito elétrico.

Entrada A	Entrada B	Saída
0	0	0
0	1	0
1	0	0
1	1	1

Fonte: Elaborado pelo autor.

Conforme Petruzella, as operações digitais de modo geral são baseadas nas funções lógicas *and*, *or* e *not*, as quais são conhecidas também como portas lógicas. Cada uma possui regras que determinam o resultado em sua saída.

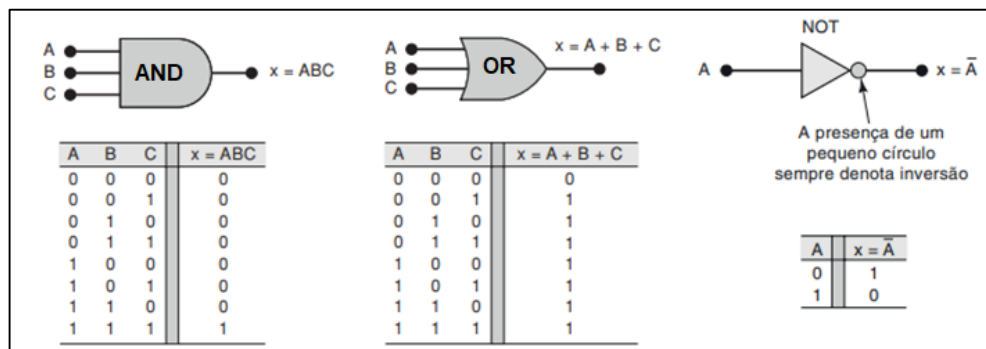
A porta *and* é um dispositivo com duas ou mais entradas, no qual a saída só terá nível lógico alto quando todos os operandos também o possuírem. Nas demais hipóteses, o nível lógico resultante será baixo. Sua expressão booleana é representada com um símbolo de multiplicação (“.”) entre as variáveis de entrada, o qual pode ser omitido.

De forma parecida, a porta *or* possui duas ou mais entradas, porém basta que apenas uma tenha nível lógico “1” para que sua saída seja positiva. Esta só terá nível “0” quando todos os operandos também o possuírem. Sua equação booleana é representada com um símbolo de adição (“+”) entre as variáveis de entrada.

Por fim, a porta *not* apresenta apenas uma entrada e é conhecida como função inversora ou negativa. O resultado de sua operação é sempre o inverso da entrada, ou seja, se o nível lógico de entrada for “1”, o de saída será “0”. A expressão algébrica da variável negada é escrita com uma barra na parte superior (“ $\bar{A}$ ”).

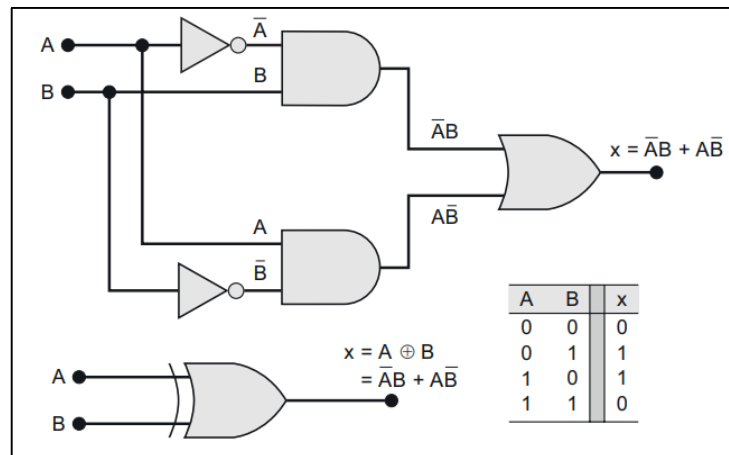
A figura 65 apresenta as portas *and*, *or* e *not*, cada uma com sua tabela-verdade e expressão lógica.

Figura 65 – Portas lógicas básicas.



Fonte: TOCCI; Ronald, 2011 (adaptado).

Com a combinação dessas três funções básicas é possível obter circuitos mais elaborados e também outras portas lógicas. Um modelo muito conhecido é a porta *xor* (*exclusive-or*), a qual produz sinal de saída “1” quando suas entradas apresentarem níveis lógicos opostos. É obtida por meio da combinação de portas *and* e *not*, conforme ilustrado na figura 66.

Figura 66 – Função lógica *xor*.

Fonte: TOCCI; Ronald, 2011 (adaptado).

Quanto mais complexos forem os circuitos lógicos, mais vantajoso será expressá-los na forma de equações booleanas, pois estas podem ser simplificadas utilizando as leis e princípios apresentados no quadro 7.

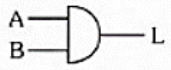
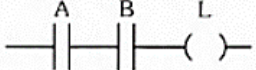
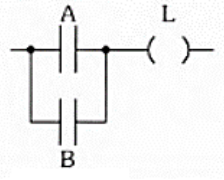
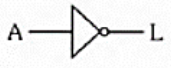
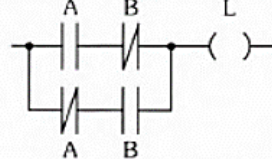
Quadro 7 – Leis da álgebra booleana.

Lei comutativa	$A + B = B + A$
	$A \cdot B = B \cdot A$
Lei associativa	$A + (B + C) = (A + B) + C$
	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$
Lei distributiva	$A \cdot (B + C) = A \cdot B + A \cdot C$
	$(A + B) \cdot (C + D) = A \cdot C + A \cdot D + B \cdot C + B \cdot D$
Lei de De Morgan	$\overline{A + B} = \bar{A} \cdot \bar{B}$
	$\overline{A \cdot B} = \bar{A} + \bar{B}$

Fonte: Elaborado pelo autor com base em TOCCI; Ronald, 2011.

As funções lógicas, além de possuírem seus equivalentes em circuitos elétricos, podem ser representadas por meio de elementos presentes nos diagramas *ladder*. Devido a isso, é possível elaborar programas para CLPs baseados em expressões booleanas obtidas a partir da tabela-verdade. Observa-se na figura 67 as portas *and*, *or*, *not* e *xor* desenvolvidas nessa linguagem.

Figura 67 – Funções lógicas em *ladder*.

Convencional	<i>Ladder</i>
	
	
	
	

Fonte: FRANCHI; Claiton Moro, 2008.

A função *and* é representada por meio de contatos normalmente abertos posicionados em série, cuja quantidade deve ser igual ao número de entradas da porta lógica. Dessa forma, a bobina na saída só será energizada caso todos os contatos assumam o valor binário “1”

A porta *or* é obtida através de contatos NA associados em paralelo. Nesse arranjo, caso qualquer um dos contatos assumam valor “1”, será formado um caminho de condução para acionar a bobina endereçada na saída.

Já a porta inversora equivale a um simples contato normalmente fechado colocado em série com a bobina. Caso esse contato não seja acionado, ou seja, permaneça em nível lógico 0, haverá condução. Entretanto, se o valor de entrada for “1”, o contato NF irá se abrir e a bobina será desligada.

Foi explicado anteriormente que as lógicas mais complexas derivam de combinações das três portas básicas, e o equivalente em *ladder* não é diferente. A porta *xor* consiste em dois ramos paralelos, cada um com dois contatos em série, sendo um NA e o outro NF. Se em um ramo os contatos A e B forem respectivamente NA e NF, no outro ramo devem ser o contrário.

Quando o sistema a ser controlado possuir muitas variáveis envolvidas, o número de combinações será grande e trabalhosa de ser analisada. A depender do caso, convém utilizar o método gráfico do mapa de Karnaugh. Este mapa é uma forma de mostrar a relação lógica entre entradas e saídas de forma minimizada.

O método consiste em transformar cada linha da tabela-verdade em um quadrado, também chamado de célula. Cada quadrado é vinculado a uma combinação de entradas na horizontal e na vertical, que é chamada de endereço. No interior de uma célula é escrito o valor lógico de saída resultante dessa respectiva combinação.

O mapa de Karnaugh é construído de forma que o endereço de cada uma de suas células difere em apenas um *bit* do endereço das células adjacentes. Essa ordenação é conhecida como código de Gray e consiste em inverter a ordem normal binária. Em uma sequência de quatro quadrados, por exemplo, os endereços horizontal e vertical são 00, 01, 11 e 10 ao invés de 00, 01, 10 e 11.

A figura 68 apresenta um mapa de Karnaugh elaborado com base em uma tabela-verdade fictícia com quatro entradas. Observa-se que as variáveis AB e CD foram utilizadas para endereçar respectivamente as linhas e colunas, utilizando o código de Gray. Os devidos resultados estão contidos no interior das células.

Figura 68 – Tabela-verdade e seu respectivo mapa de Karnaugh.

A	B	C	D	X
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

		CD			
		00	01	11	10
AB	00	0	1	0	0
	01	0	1	0	0
	11	0	1	1	0
	10	0	0	0	0

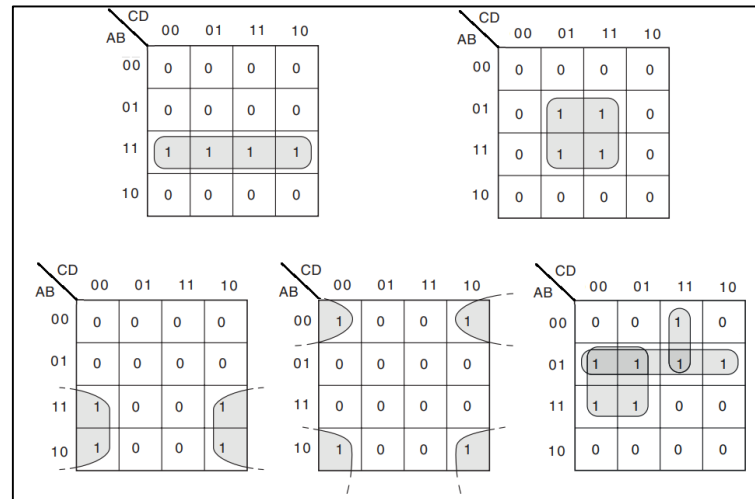
Fonte: TOCCI; Ronald, 2011 (adaptado).

Dentro do mapa de Karnaugh é possível formar grupos de células adjacentes cujo valor binário seja "1" e, a partir da análise dos mesmos, obter expressões lógicas simplificadas de forma mais rápida.

O agrupamento dessas células pode ser realizado em duplas, quartetos, octetos e sucessivas potências de dois. Os grupos assumem diferentes formas, inclusive podem utilizar as células dos quatro cantos do mapa e sobrepor outros

grupos, conforme ilustra a figura 69. O objetivo é encontrar os maiores grupos que cubram todas os casos em que o valor de saída seja positivo.

Figura 69 – Agrupamento de células.



Fonte: TOCCI; Ronald, 2011 (adaptado).

Para a obtenção da equação booleana, algumas regras devem ser observadas. Dentro de um agrupamento de células positivas, as variáveis de endereço que sofrerem alteração de valor não influenciam no resultado e podem ser descartadas da expressão. As variáveis de entrada que permanecerem com valor “0” em todos os quadrados do grupo devem ser escritas de forma negada, e as variáveis que permanecerem com valor “1” devem ser escritas em sua forma original.

Como exemplo de aplicação da álgebra booleana na elaboração de um diagrama *ladder*, considera-se um sistema fictício com três variáveis de entrada. Seu comportamento é ilustrado pela tabela-verdade da figura 70, através da qual se obtém a equação lógica não-simplificada composta pela soma de quatro produtos.

Figura 70 – Tabela-verdade do sistema fictício.

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	1	1
1	0	1	1
1	1	0	0
1	1	1	0

$\bar{A} \cdot \bar{B} \cdot C$
 $\bar{A} \cdot B \cdot C$
 $A \cdot \bar{B} \cdot \bar{C}$
 $A \cdot \bar{B} \cdot C$

$$S = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot C$$

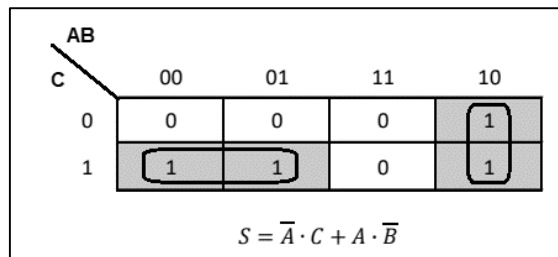
Fonte: Elaborado pelo autor através do software Microsoft Excel.

Ao aplicar o método do mapa de Karnaugh nesse problema, observa-se que as células com valor binário “1” podem ser divididas em dois pares. Por esse motivo a equação lógica é dada pela soma de dois termos, conforme mostrado na figura 71.

No par de células à esquerda, percebe-se que a variável de endereço B sofre alteração, logo a mesma é descartada. Como as variáveis A e B permanecem respectivamente com os valores lógicos “0” e “1”, a primeira é escrita na equação com seu valor negado e a segunda com seu valor original.

No par à direita, a variável C sofre alteração e é descartada. Já as variáveis A e B permanecem respectivamente com os valores lógicos “1” e “0”, logo a primeira é escrita na forma original e a segunda de forma negada.

Figura 71 – Mapa de Karnaugh do sistema fictício.

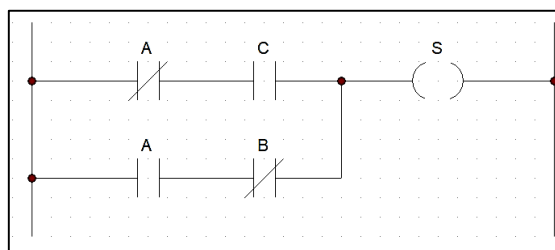


Fonte: Elaborado pelo autor através do software Microsoft Excel.

Com posse da equação lógica simplificada, é possível elaborar com facilidade o diagrama *ladder* adequado para o controle desse sistema. Como a expressão é constituída pela soma de dois termos, haverá dois ramos em paralelo.

Um dos ramos deve ser constituído por um contato NF em série com um contato NA, vinculados respectivamente às variáveis A e C. No outro ramo, coloca-se um contato NA em série com um contato NF, sendo o primeiro vinculado à variável A e o segundo à variável B. Vide figura 72.

Figura 72 – Diagrama *ladder* para o sistema fictício.



Fonte: Elaborado pelo autor através do software CADe_SIMU 3.0.

4.10 Noções de Redes de Comunicação

Conforme explicado nos subcapítulos anteriores, os CLPs possuem cartões de entradas e saídas que fazem sua conexão com sensores e atuadores. Entretanto, em decorrência do grande número de dispositivos presentes em determinadas plantas, pode não ser viável conectá-los separadamente ao controlador. Nesse contexto, foram criadas outras formas de comunicação.

De acordo com Petruzella, as comunicações de dados referem-se aos diferentes modos com que os controladores comunicam entre si e com outros dispositivos. As redes comunicação são muito utilizadas em aplicações industriais e apresentam várias vantagens.

Existem diversos meios físicos para a transmissão de dados em uma rede, os principais são os pares trançados, cabos coaxiais e a fibra óptica. As características que os diferenciam geralmente estão relacionadas à taxa de transferência e a imunidade a ruídos, o que torna cada tipo ideal para uma determinada aplicação.

Os métodos de comunicação em rede minimizam a quantidade de condutores necessária, uma vez que é possível conectar os dispositivos a um único cabo de transmissão. Dessa forma, o tempo de instalação é reduzido e a verificação de defeitos facilitada.

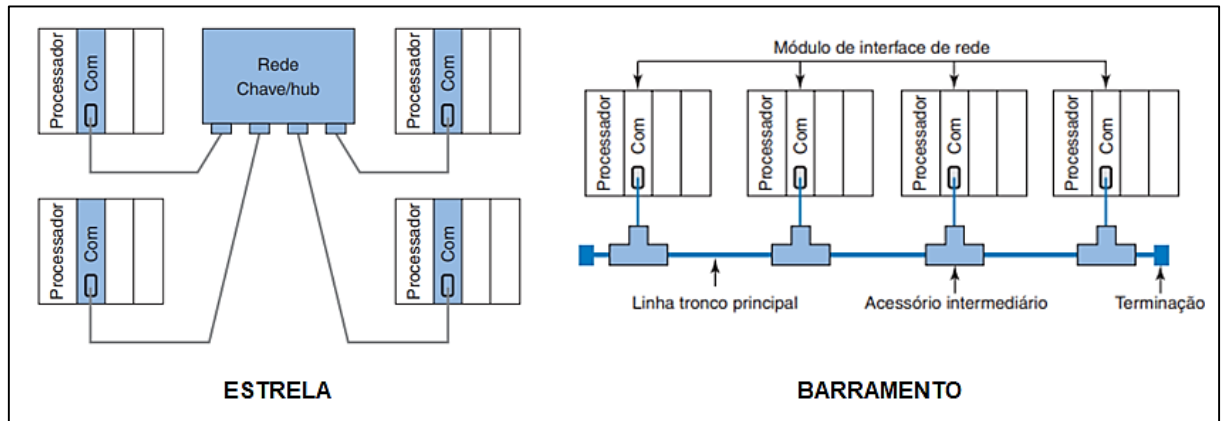
Os dispositivos conectados a uma rede de comunicação são comumente denominados nó ou estação. A forma com que esses elementos são dispostos e interligados determinam a topologia da rede, que pode ser em estrela ou em barramento.

As redes em estrela possuem uma central que é conectada aos vários nós do sistema e faz intermédio da transmissão de dados. Toda informação é enviada para a central, que em seguida a envia para seu destino. Essa topologia é adequada para aplicações onde os dados são enviados de um dispositivo principal aos demais, porém causa lentidão em situações que necessitem de comunicação direta entre os dispositivos secundários.

As redes em barramento são constituídas por um cabo-tronco onde os dispositivos devem ser ligados por meio de conectores de derivação. As informações enviadas ao sistema são recebidas por todos os nós, os quais possuem endereço de identificação. Cada nó aceita apenas os comandos que estejam destinados ao seu endereço.

Observa-se ilustrado na figura 73 as topologias de rede explicadas nos parágrafos anteriores.

Figura 73 – Topologias de rede de comunicação industrial.



Fonte: PETRUZELLA; Frank, 2014 (adaptado).

As redes em barramento necessitam de métodos de controle de acesso para assegurar que os dados sejam transmitidos de forma organizada e evitar que mais de uma mensagem seja enviada simultaneamente. Os métodos mais comuns são os de passagem de ficha e detecção de colisão.

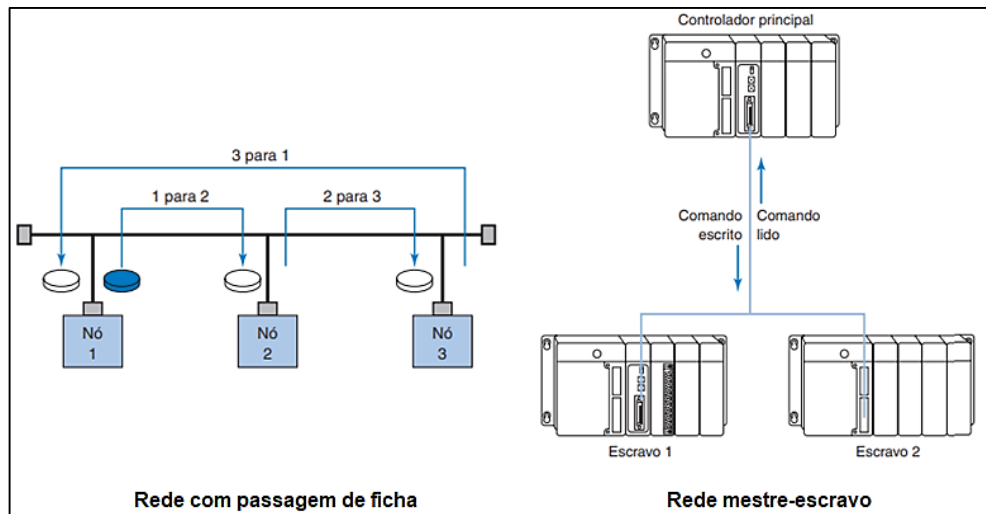
O método da passagem de ficha (*token*) consiste em permitir que um nó transmita dados apenas se ele estiver com posse do *token*. Após o envio dos dados, o *token* é passado para o nó seguinte, o que garante que cada dispositivo tenha a oportunidade de enviar dados na rede sem interferência.

A detecção de colisão é um método no qual os dispositivos detectam atividades na rede e só transmitem dados quando não houver outra mensagem sendo enviada. Nesse método há possibilidade de que mais de uma mensagem seja enviada ao mesmo tempo (colisão). Quando isso acontece, cada nó aguarda um tempo aleatório e reenvia seus dados.

No contexto de controle de acesso, é importante mencionar as redes com comunicação mestre-escravo. Nesses sistemas, toda troca de dados é controlada por um dispositivo mestre. Este envia mensagens aos nós escravos, que são capazes apenas de obedecer aos comandos e enviar respostas quando solicitados.

A figura 74 ilustra os métodos de controle por passagem de ficha e por solicitação do mestre em uma rede mestre-escravo.

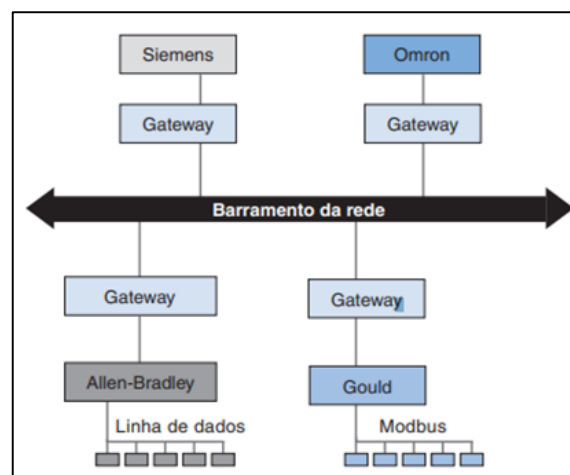
Figura 74 – Métodos de controle de acesso à rede.



Fonte: PETRUZELLA; Frank, 2014 (adaptado).

Para que haja comunicação entre os diferentes elementos que constituem a rede, é necessário que todos sigam um conjunto de regras conhecidas como protocolo. No passado os protocolos de comunicação eram de propriedade dos fabricantes de dispositivos e não podiam ser utilizados livremente. Porém, na atualidade existem diversos protocolos abertos baseados em padrões internacionais.

A comunicação entre protocolos e arquiteturas diferentes é possibilitada por interfaces denominadas *gateways*. A função desses elementos é converter dados e mudar o formato das mensagens de forma que uma rede consiga se comunicar com outra. Observa-se ilustrado na figura 75 dispositivos de diferentes fabricantes conectados à uma mesma rede em barramento através de *gateways*.

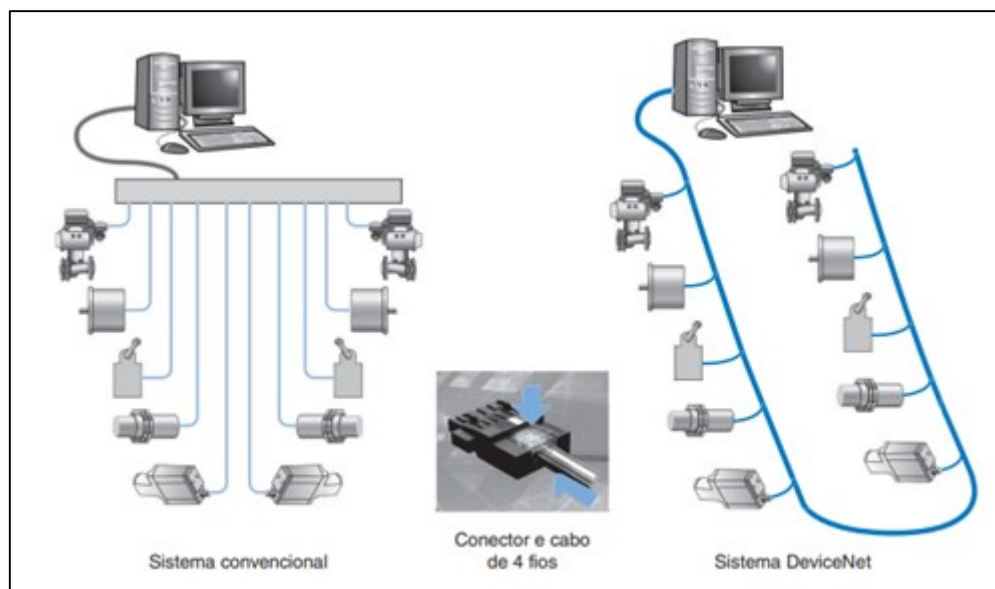
Figura 75 – Utilização de *gateways* para conectar dispositivos.

Fonte: PETRUZELLA; Frank, 2014.

Os protocolos abertos de comunicação de dados mais conhecidos e utilizados são o Modbus, Profibus-DP e DeviceNet. Nos próximos parágrafos será feita a apresentação de cada um deles.

A *DeviceNet* é uma rede composta por um barramento com dois pares de condutores, sendo um par para a transmissão de dados e o outro para a alimentação dos nós. Seu tronco é capaz de suportar até 64 dispositivos, o que reduz o custo e o tempo investido em cabeamento. Observa-se na figura 76 um comparativo entre o sistemas convencional, no qual os dispositivos são conectados diretamente aos módulos de E/S do controlador, e a *DeviceNet*.

Figura 76 – Sistemas convencional e *DeviceNet*.



Fonte: PETRUZELLA; Frank, 2014.

A comunicação do CLP com os nós nesse protocolo é intermediada por um módulo *scanner*, que é responsável por ler e escrever dados e monitorar o estado operacional de cada elemento. Os dispositivos de campo conectados à DeviceNet geralmente são capazes de fornecer informações além da condição ligado/desligado, como diagnósticos sobre seu funcionamento e vida útil.

O Modbus é o protocolo de comunicação livre mais utilizado na indústria, foi criado pela fabricante Modicon e é baseado no sistema mestre-escravo. Em seu funcionamento a comunicação é sempre iniciada pelo Modbus Mestre, que envia comandos aos Modbus Escravos. Estes não comunicam entre si, apenas obedecem ao comando recebido e retornam uma resposta ao mestre.

Cada mensagem enviada pelo dispositivo mestre tem em seu conteúdo um endereço de destino. Todos os escravos recebem a mensagem, porém apenas o que estiver no endereço correspondente obedece ao comando.

Quando a comunicação iniciada pelo mestre é endereçada a um escravo específico, denomina-se *unicast*. Quando o destino for todos os escravos da rede, chama-se *broadcast*. Nesse último caso não há resposta e todos os escravos obedecem ao comando.

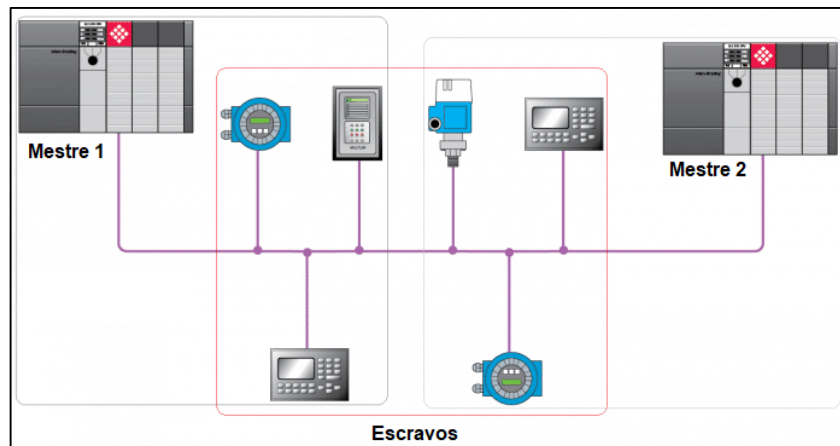
O meio físico de transferência de dados no protocolo Modbus normalmente são os cabos seriais RS-232 ou RS-485, cuja sigla vem do inglês *Recommended Standard* e significa “padrão sugerido”. O primeiro geralmente possui comprimento máximo muito limitado, enquanto o segundo é capaz de transmitir dados a uma distância maior.

Uma rede Modbus normalmente possui 256 endereços, de 0 a 255. O endereço 0 é o destinado ao *broadcast*, enquanto de 1 a 247 é possível endereçar escravos. Já os endereços de 248 a 255 são reservados. Nessa rede, o mestre não possui endereço específico.

O protocolo Profibus-DP, por sua vez, possui como característica a alta velocidade de comunicação. Normalmente é utilizado o meio físico RS-485 e são alcançadas taxas de transferência de dados de 12Mbps (*Mega bits* por segundo), com distâncias de até 1.200 metros. A sigla DP vem de *decentralized periphery* e significa “periférico descentralizado”.

O Profibus-DP é capaz de operar com mais de um mestre conectado ao barramento (rede multimestre), e cada mestre possui seus respectivos escravos. Um mestre pode consultar os parâmetros de todos os escravos da rede, mas só pode enviar comandos aos escravos de seu subsistema. Vide figura 77.

Figura 77 – Rede Profibus-DP multimestre.



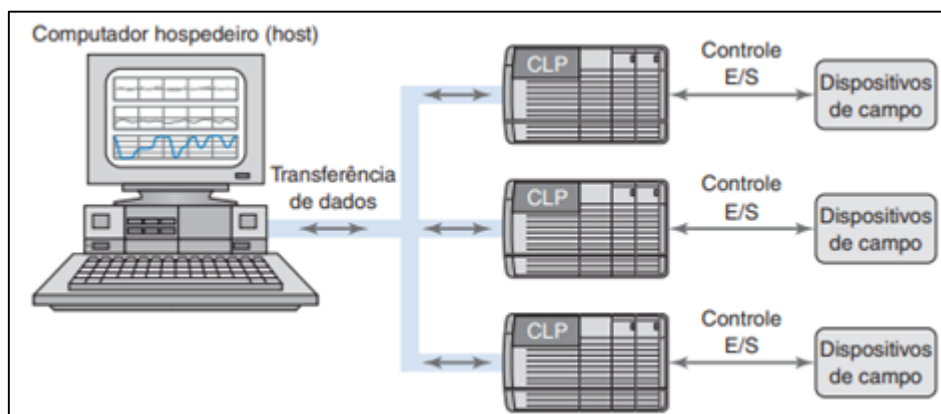
Fonte: Embarcados, 2016.

Por fim, além de realizar as funções de controle, os CLPs são capazes de coletar dados e estruturá-los para a geração de relatórios. Ao integrar o controlador em uma rede de comunicação que contenha um computador provido dos *softwares* necessários, é possível supervisionar o processo controlado em tempo real.

Esses *softwares* são conhecidos como SCADA, sigla que vem do inglês *Supervisory Control and Data Acquisition* e significa “controle de supervisão e aquisição de dados”. O computador que o executa é denominado *host* e geralmente se localiza em uma sala de controle. Vide figura 78.

Com o sistema SCADA é possível registrar dados, observar tendências e ajustar parâmetros e processos para obter eficiência máxima. É muito utilizado pelas indústrias para monitorar e controlar o processo produtivo. As informações são exibidas em telas e normalmente ficam sob cuidado de operadores humanos.

Figura 78 – Sistema SCADA.



Fonte: PETRUZELLA; Frank, 2014.

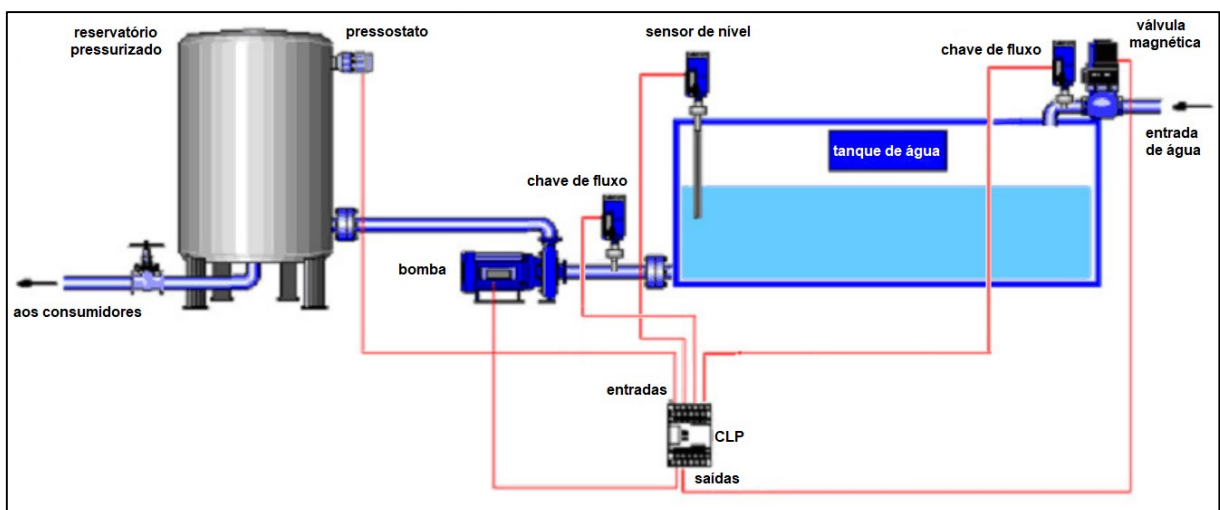
5 APLICAÇÃO

Neste capítulo será apresentada a simulação via *software* da aplicação de um controlador lógico programável em um sistema de suprimento de água. O CLP será programado em linguagem *Ladder*, fará a leitura de sinais de sensores e acionará atuadores, de forma a garantir o correto funcionamento do conjunto.

5.1 Sistema de Bombeamento

O sistema ilustrado na figura 79 é composto por um tanque, um reservatório pressurizado, tubulações, quatro sensores e dois atuadores. Seu objetivo é receber água na entrada, armazená-la no tanque e bombeá-la de forma a manter a pressão dentro do reservatório adequada. Este possui uma saída responsável por alimentar os consumidores.

Figura 79 – Sistema de suprimento de água.



Fonte: Disponibilizado pelo orientador.

Para controlar o ingresso de água no tanque, a tubulação de entrada é provida de uma válvula magnética e uma chave de fluxo que operam com sinais digitais. A primeira é capaz de abrir ou fechar a passagem de água quando recebe os sinais lógicos 1 ou 0, nessa ordem. De forma análoga, a segunda indica se há ou não fluxo de líquido respectivamente pelos sinais 1 e 0.

Quando a válvula é aberta e a chave de fluxo não detecta passagem de líquido, o sistema ativa um alarme para sinalizar a falta de abastecimento de água e incrementa uma unidade ao contador presente no programa do CLP. O valor acumulado na contagem pode ser zerado a partir de um botão de impulso.

O tanque tem como objetivo criar uma reserva de água para alimentar a bomba, que fica conectada à sua base. Em seu interior há um sensor de nível instalado, o qual indica de forma digital a necessidade de abrir ou fechar a válvula de entrada. Quando o nível de água armazenada é suficiente, o sensor emite sinal lógico 0 e, caso contrário, indica sinal 1. A bomba não é ligada quando o nível de água no tanque for baixo.

Para evitar bruscas acelerações no fluido, o motor associado à bomba utiliza a chave de partida estrela-triângulo. Conforme explicado nos capítulos antecedentes, esta reduz o conjugado acelerante no início do movimento e previne danos ao sistema.

A tubulação que liga a base do tanque até a bomba possui outra chave de fluxo, cujo objetivo é garantir que não ocorra operação a seco. Trata-se de uma proteção adicional que desliga o motor da bomba quando a passagem de água não é identificada por mais de um segundo durante a operação do mesmo.

A situação de falha apresentada no parágrafo anterior ocorre apenas quando o sensor de nível, por algum defeito, não indica que o tanque está com pouca água. Nessa condição, caso a bomba seja acionada e o nível do tanque chegue a zero, o motor é desligado e um alarme ativado. Uma vez que isso aconteça, o funcionamento do sistema é interrompido e só se reestabelece quando o operador pressiona um botão de impulso.

Ademais, a tubulação de saída da bomba possui uma válvula de retenção que não permite o retorno da água enviada até o reservatório pressurizado. A pressão no interior deste é monitorada por um pressostato. Quando a pressão está baixa, o pressostato emite sinal lógico 1 para ligar a bomba. Caso contrário, emite 0 para desligá-la.

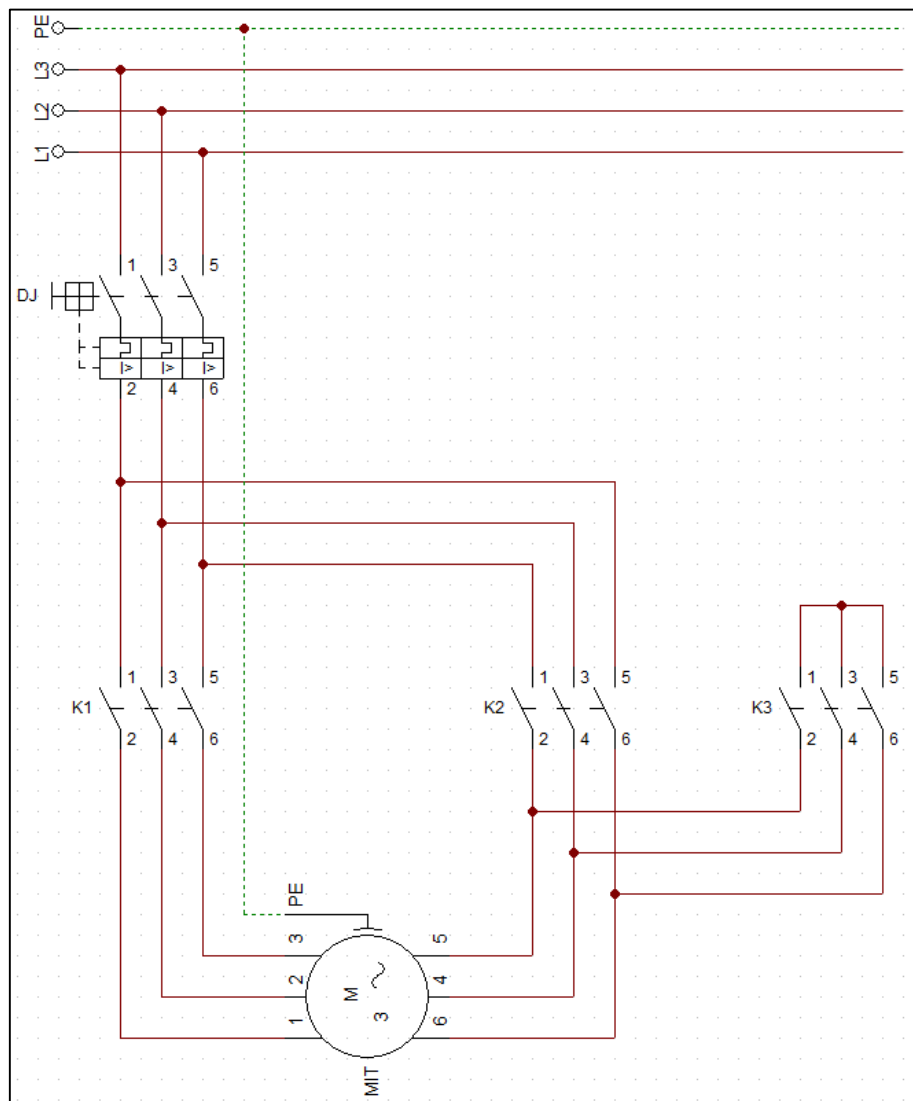
O sistema em questão é relativamente simples, com poucos dispositivos envolvidos. Dessa forma, basta um CLP compacto com poucos pontos de E/S para realizar o controle do mesmo.

5.2 Programação em Ladder

A programação em *ladder* foi desenvolvida através do *software* CADe_SIMU 3.0, no qual foram utilizados contatos, bobinas e blocos de funções para criar uma lógica que atenda às necessidades do sistema. Os sensores e atuadores foram endereçados nas tabelas de entrada e de saída e vinculados ao *software* PC_SIMU.

Conforme ilustrado na figura 80, a chave de partida do motor requer o uso de três contatores: K1, K2 e K3. Consequentemente, é necessário que cada um deles seja associado a um endereço de saída do controlador, o qual é responsável por acioná-los de forma sequencial e temporizada para partir a bomba de forma adequada.

Figura 80 – Circuito de potência MIT.



Fonte: Elaborado pelo autor no CADe_SIMU 3.0.

Na tabela posicionada à esquerda na figura 81, observa-se que o sensor de nível, as chaves de fluxo do tanque e da bomba, o pressostato e os botões de *reset* do contador e de reestabelecimento do sistema estão vinculados respectivamente aos endereços de entrada I0.0, I0.1, I0.2, I0.3, I0.4 e I0.5.

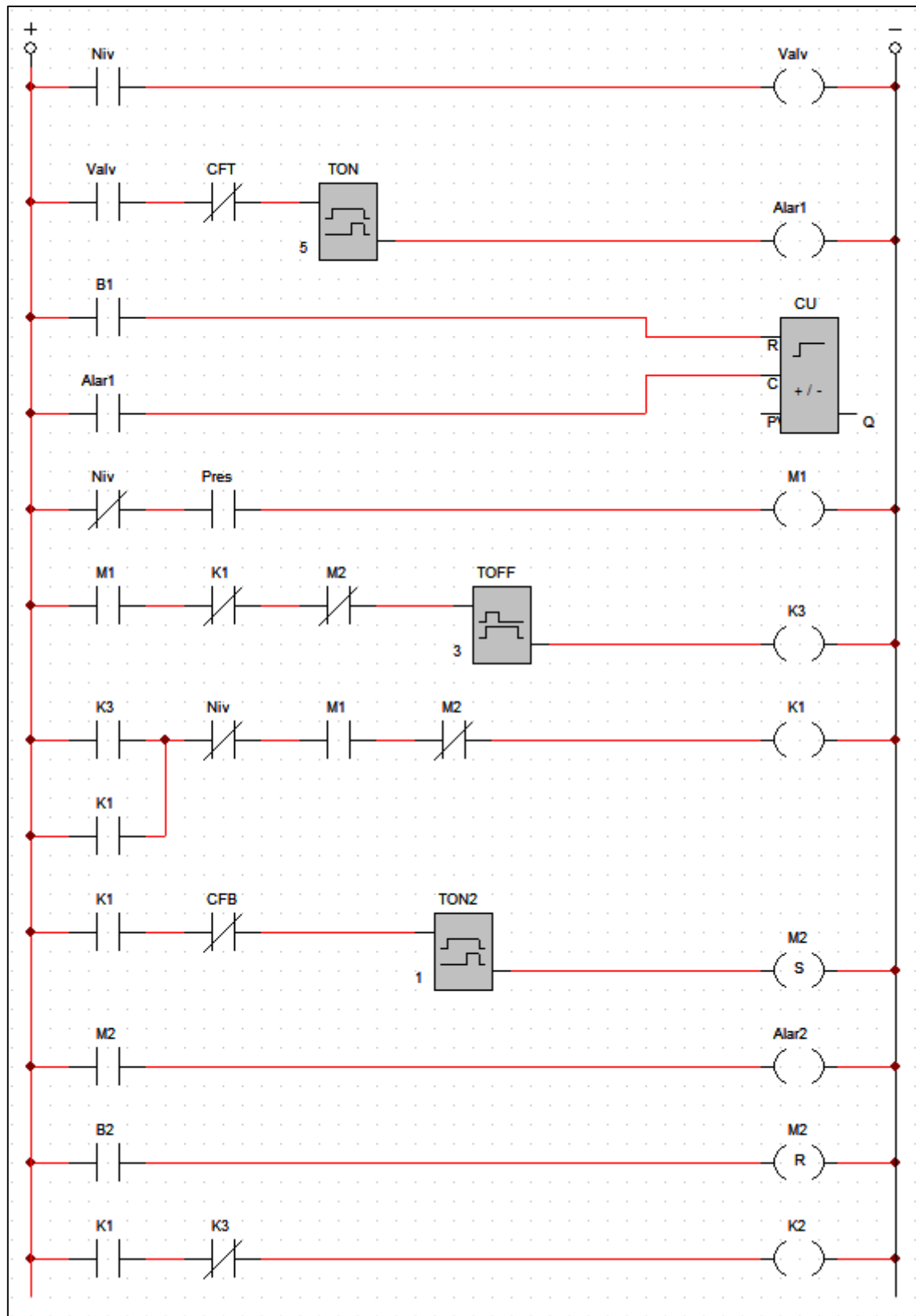
Já na tabela à direita na figura, verifica-se que a válvula magnética e os alarmes de falta de abastecimento e de operação a seco estão endereçados aos pontos de saída Q0.0, Q0.1 e Q0.2. Ademais, os contadores K1, K2 e K3 da chave-estrela triângulo foram associados a Q0.3, Q0.4 e Q0.5, nessa ordem.

Figura 81 – Tabelas de entrada e de saída com variáveis associadas.

I N P U T		O U T P U T	
Niv	I0.0	Valv	Q0.0
CFT	I0.1	Alar1	Q0.1
CFB	I0.2	Alar2	Q0.2
Pres	I0.3	K1	Q0.3
B1	I0.4	K2	Q0.4
B2	I0.5	K3	Q0.5
	I0.6		Q0.6
	I0.7		Q0.7

Fonte: Elaborado pelo autor no CADe_SIMU 3.0.

Com as variáveis definidas e associadas, os degraus do programa foram desenvolvidos conforme apresentado na figura 82. Os resultados da simulação em cada etapa de funcionamento do sistema estão detalhados no apêndice deste trabalho.

Figura 82 – Lógica em *ladder*.

Fonte: Elaborado pelo autor no CADe_SIMU 3.0.

No primeiro degrau está definido que a condição de acionamento da válvula magnética é o sinal positivo proveniente do sensor de nível. O segundo estabelece que, quando a válvula estiver aberta e a chave de fluxo não indicar a passagem de líquido dentro de cinco segundos, o alarme de falta de abastecimento é acionado.

Já o terceiro e quarto degrau definem os parâmetros de entrada de um contador crescente. O alarme mencionado no parágrafo anterior é responsável por incrementar o contador, e o botão B1 pelo *reset*.

O quinto degrau implementa uma bobina auxiliar M1, que é acionada quando o nível de água no tanque é suficiente e a pressão do reservatório está baixa. O contato normalmente aberto de M1 ativa um temporizador que energiza o contator K3 por três segundos no sexto degrau. Nesse período, contato associado a K3 fecha o circuito até a bobina de K1 no sétimo degrau, momento em que o motor parte com ligação em estrela.

O oitavo degrau, por sua vez, analisa se há fluxo de água chegando até a bomba após o início de seu movimento. Caso negativo, uma bobina de *set* vinculada a M2 é acionada após um segundo. Nessa situação, os contadores K1 e K3 são desativados e o alarme de operação a seco é ligado no nono degrau.

Para que a M2 tenha seu valor reestabelecido, é necessário que a bobina de *reset* no décimo degrau seja energizada pelo fechamento do contato vinculado ao botão de reestabelecimento do sistema B2.

Caso a condição de falha representada por M2 não seja ativada, quando ocorrer a desenergização de K3 por parte do temporizador de desligamento, a bobina de K2 é acionada no décimo primeiro degrau. Nesse momento, a ligação do motor é trocada de estrela para triângulo.

Quando a pressão no reservatório atingir o valor adequado, o contato associado ao pressostato deixará de conduzir no quinto degrau e a bobina M1 será desativada. Consequentemente, K1 será desenergizado e o motor desligará.

6 CONSIDERAÇÕES FINAIS

Com a realização deste trabalho foi possível complementar os conhecimentos adquiridos ao longo do curso de graduação em engenharia elétrica. O estudo sobre CLPs permitiu a compreensão de como a eletrônica e a informática ganharam espaço em aplicações de controle e substituíram o uso da lógica baseada em interligações elétricas de dispositivos.

Os aspectos construtivos e funcionais dos controladores explanados ao longo da redação provaram a flexibilidade, robustez, confiabilidade, agilidade e economicidade dos mesmos em relação aos métodos utilizados no passado, fato que os tornaram importantes no campo de atuação profissional e justificou a presente pesquisa.

O aprendizado acerca das linguagens de programação evidenciou que as funções existentes nos CLPs permitem elaborar de forma simples operações complexas do ponto de vista dos diagramas elétricos convencionais. No que se refere aos dispositivos de programação, constatou-se a facilidade de configuração e modificação da lógica de controle.

Por fim, aplicação de um controlador na resolução de um problema por meio de simulação computacional possibilitou a vinculação entre o objeto do estudo e situações que podem ocorrer no exercício da profissão. Dessa forma, consolidou-se a aprendizagem sobre o tema.

7 REFERÊNCIAS

- [1] Da revolução industrial à conectividade sem fio: o que é automação industrial. **Novus**, 2021. Disponível em: <https://www.novus.com.br/blog/da-revolucao-industrial-a-conectividade-sem-fio-o-que-e-automacao-industrial>. Acesso em: 7 de abr. de 2022.
- [2] NATALE, Ferdinando. **Automação Industrial**. 8ª ed. São Paulo: Editora Érica, 2008.
- [3] PRUDENTE, Francesco. **Automação Industrial: PLC Teoria e Aplicações**. 2ª ed. Rio de Janeiro: Editora LTC, 2011.
- [4] FRANCHI, Claiton Moro. **Acionamentos Elétricos**. 4ª ed. São Paulo: Editora Érica, 2008.
- [5] FRANCHI, Claiton Moro; CAMARGO, Valter Luís Arlindo. **Controladores Lógicos Programáveis: Sistemas Discretos**. 1ª ed. São Paulo: Editora Érica, 2008.
- [6] LAMB, Frank. **Automação Industrial na Prática**. São Paulo: Editora AMGH, 2015.
- [7] SOUZA, Neemias S. **Apostila de Acionamentos Elétricos**. 2009. Disponível em: <https://docente.ifrn.edu.br/heliopinheiro/Disciplinas/maquinas-acionamentos-eletricos/apostila-basica>. Acesso em: 10 de abr. de 2022.
- [8] O que é um sistema automatizado. **Mobiler**, 2019. Disponível em: <https://mobiler.com.br/mobilidade/2019/03/06/o-que-e-um-sistema-automatizado>. Acesso em: 10 de abr. de 2022.
- [9] Manual de Instruções Siemens LOGO!. Disponível em: https://cache.industry.siemens.com/dl/files/461/16527461/att_82574/v1/Logo_pt.pdf. Acesso em: 29 de abr. de 2022.

[10] TANNUS, Alexandre Moraes. **Sistemas de Medidas**. 2019. Disponível em: <http://repositorio.aee.edu.br/bitstream/aee/1851/3/Sistema%20de%20Medidas.pdf>
Acesso em: 30 de jun. de 2022.

[11] Datasheet ST Electronics EPROM M27C512. 2007. Disponível em: <https://www.tme.com/Document/43a6b59f99f54c7938aa1e9c681af6fd/109568.pdf>
Acesso em: 30 de jun. de 2022.

[12] Datasheet SIMATIC S7 Backup Battery 3.6V for S7-300 CPUs. Disponível em: <https://mall.industry.siemens.com/mall/pt/pt/Catalog/Product/6ES7971-1AA00-0AA0>
Acesso em: 1º de jul. de 2022.

[13] Datasheet Simatic S7-1200 Power Module PM1207 Stabilized. Disponível em: <https://mall.industry.siemens.com/mall/en/ww/Catalog/Product/6EP1332-1SH71>
Acesso em: 2 de jul. de 2022.

[14] Página de Controladores Avançados Siemens Simatic. Disponível em: <https://new.siemens.com/br/pt/produtos/automacao/controladores.html>
Acesso em: 2 de jul. de 2022.

[15] Manual Siemens Simatic S7-200. Disponível em: https://cache.industry.siemens.com/dl/files/582/1109582/att_22063/v1/s7200_system_manual_en-US.pdf. Acesso em: 8 de jul. de 2022.

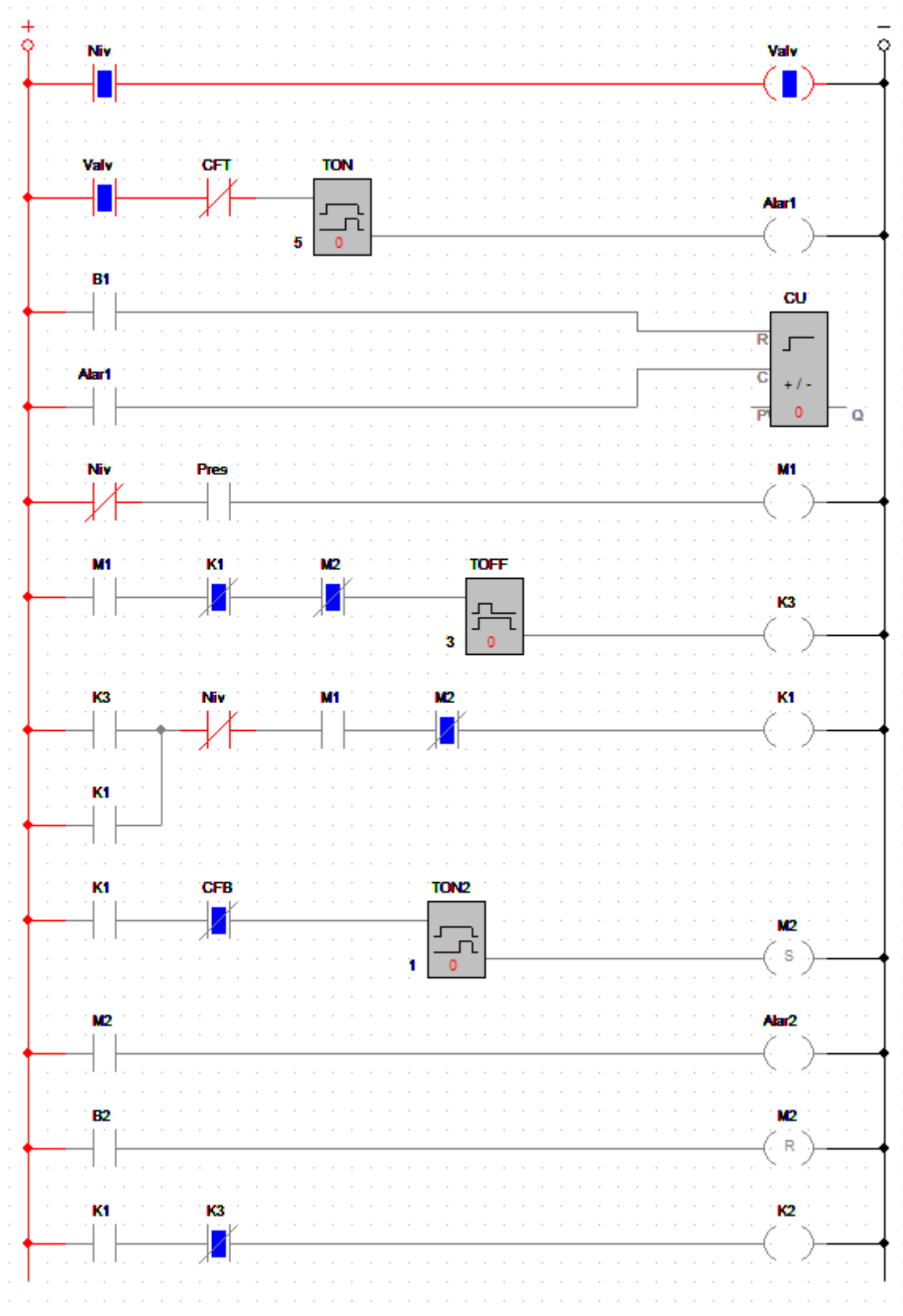
[16] Datasheet Optocoupler Phototransistor Output. Vishay Semiconductor. Disponível em: <https://www.vishay.com/docs/83673/sfh618a.pdf> Acesso em: 2 de jul. de 2022.

[17] Documentação Controlador Rockwell Micro820. Disponível em: <https://www.rockwellautomation.com/pt-br/products/hardware/allen-bradley/programmable-controllers/micro-controllers/micro800-family/micro820-controllers.html>. Acesso em: 8 de jul. de 2022.

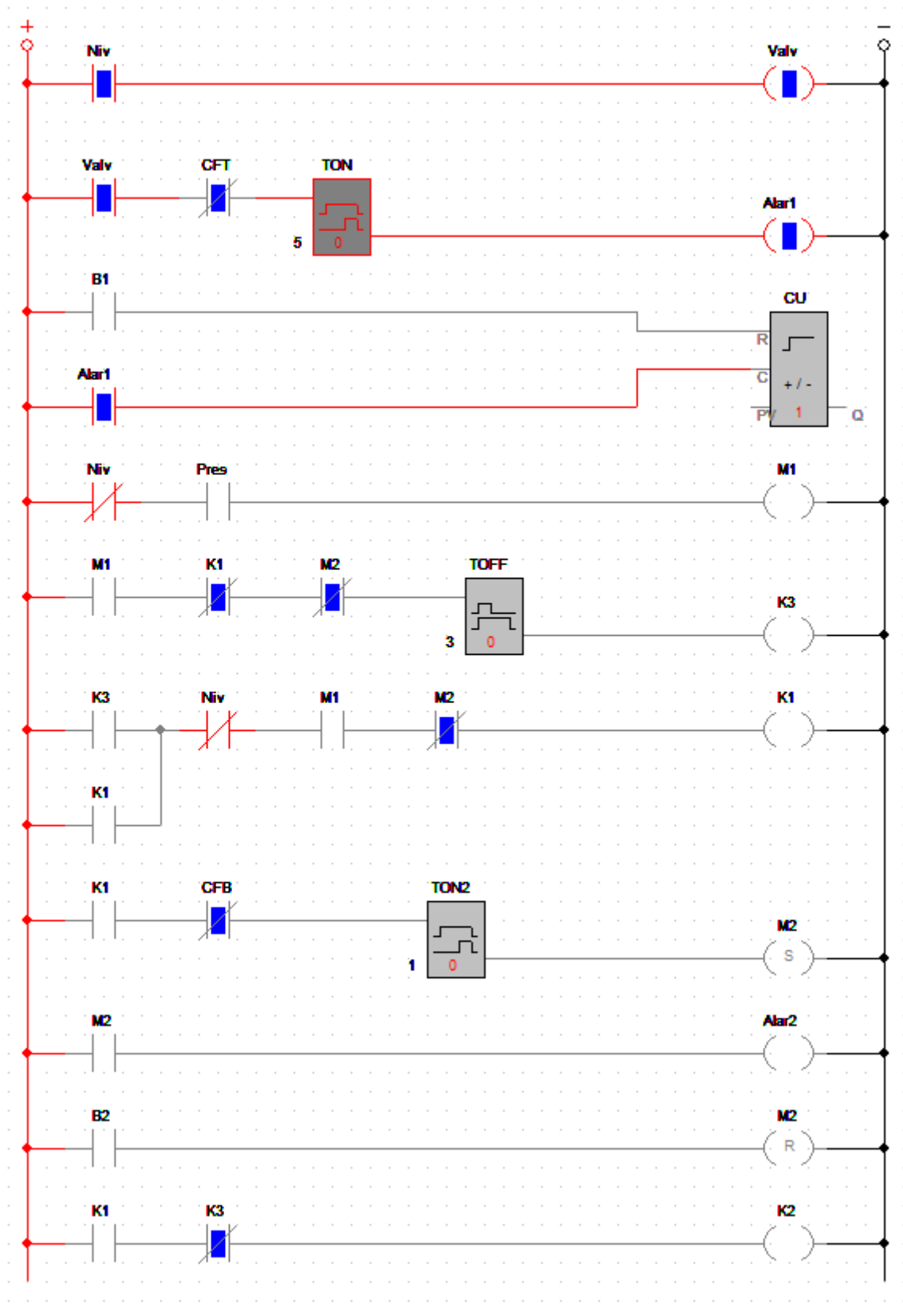
- [18] Prefixes for binary multiples. **NIST**, 2021. Disponível em: <https://physics.nist.gov/cuu/Units/binary.html>. Acesso em: 10 de jul. de 2022.
- [19] International Electrotechnical Commission. **IEC 61131-3: Programmable controllers - Part 3: Programming Languages**. Switzerland, 2003. Disponível em: https://d1.amobbs.com/bbs_upload782111/files_31/ourdev_569653.pdf. Acesso em: 10 de jul. de 2022.
- [20] Sinal Analógico x Sinal Digital. **Embarcados**, 2017. Disponível em: <https://embarcados.com.br/sinal-analogico-x-sinal-digital>. Acesso em: 10 de jun. de 2022.
- [21] Como Funciona a Linguagem LADDER. **Citisystems**, 2016. Disponível em: <https://www.citisystems.com.br/linguagem-ladder/>. Acesso em: 15 de jul. de 2022.
- [22] FLOYD, Thomas L. **Sistemas Digitais: Fundamentos e Aplicações**. 9ª ed. Porto Alegre: Bookman, 2007. Tradução de José Lucimar do Nascimento.
- [23] TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L. **Sistemas Digitais: Princípios e Aplicações**. 11ª ed. São Paulo: Pearson Education, 2011. Tradução de Jorge Ritter.
- [24] PETRUZELLA, Frank D. **Controladores Lógicos Programáveis**. 4ª ed. Porto Alegre: Bookman, 2014. Tradução de Romeu Abdo.
- [25] Página de Produtos Schneider IHM Touch 4" HMIST6200. Disponível em: <https://www.se.com/br/pt/product/HMIST6200/ihm-touch-4-1com-1ethernet-usb-24vdc>. Acesso em: 8 de jul. de 2022.
- [26] Protocolo Modbus: Fundamentos e Aplicações. **Embarcados**, 2014. Disponível em: <https://embarcados.com.br/protocolo-modbus>. Acesso em: 17 de jun. de 2022.

- [27] Fundamentos e Aplicações do Protocolo Profibus. **Embarcados**, 2016.
Disponível em: <https://embarcados.com.br/fundamentos-protocolo-profibus/>. Acesso em: 17 de jun. de 2022.
- [28] FITZGERALD, A. E.; KINGSLEY, Charles Jr.; UMANS, Stephen D. **Máquinas Elétricas**. 7ª ed. Porto Alegre: AMGH Editora Ltda, 2014. Tradução de Anatólio Laschuk.
- [29] MAMEDE FILHO, João. **Instalações Elétricas Industriais**. 9ª ed. Rio de Janeiro: LTC, 2017.
- [30] Catálogo Comercial Motor Trifásico Weg W22. Disponível em:
<https://static.weg.net/medias/downloadcenter/hb7/h10/WEG-w22-motor-trifasico-50025536-brochure-portuguese-web.pdf>. Acesso em: 20 de jul. de 2022.
- [31] Guia de Seleção de Partidas Weg. Disponível em:
<https://static.weg.net/medias/downloadcenter/haa/h53/WEG-guia-de-selecao-de-partidas-50037327-manual-portugues-br-dc.pdf>. Acesso em: 20 de jun. de 2022.

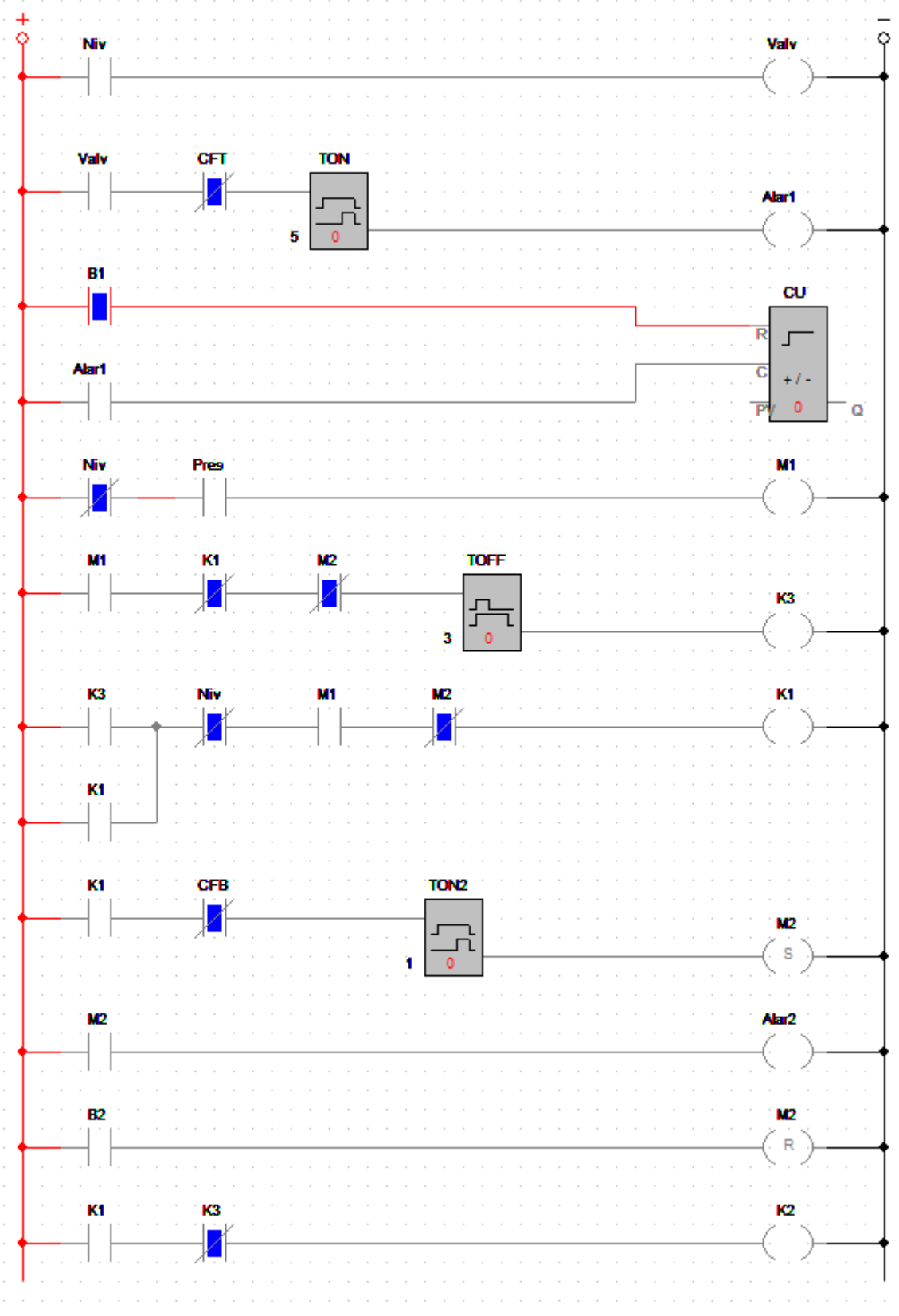
Apêndice A – Reabastecimento do tanque em condições normais.



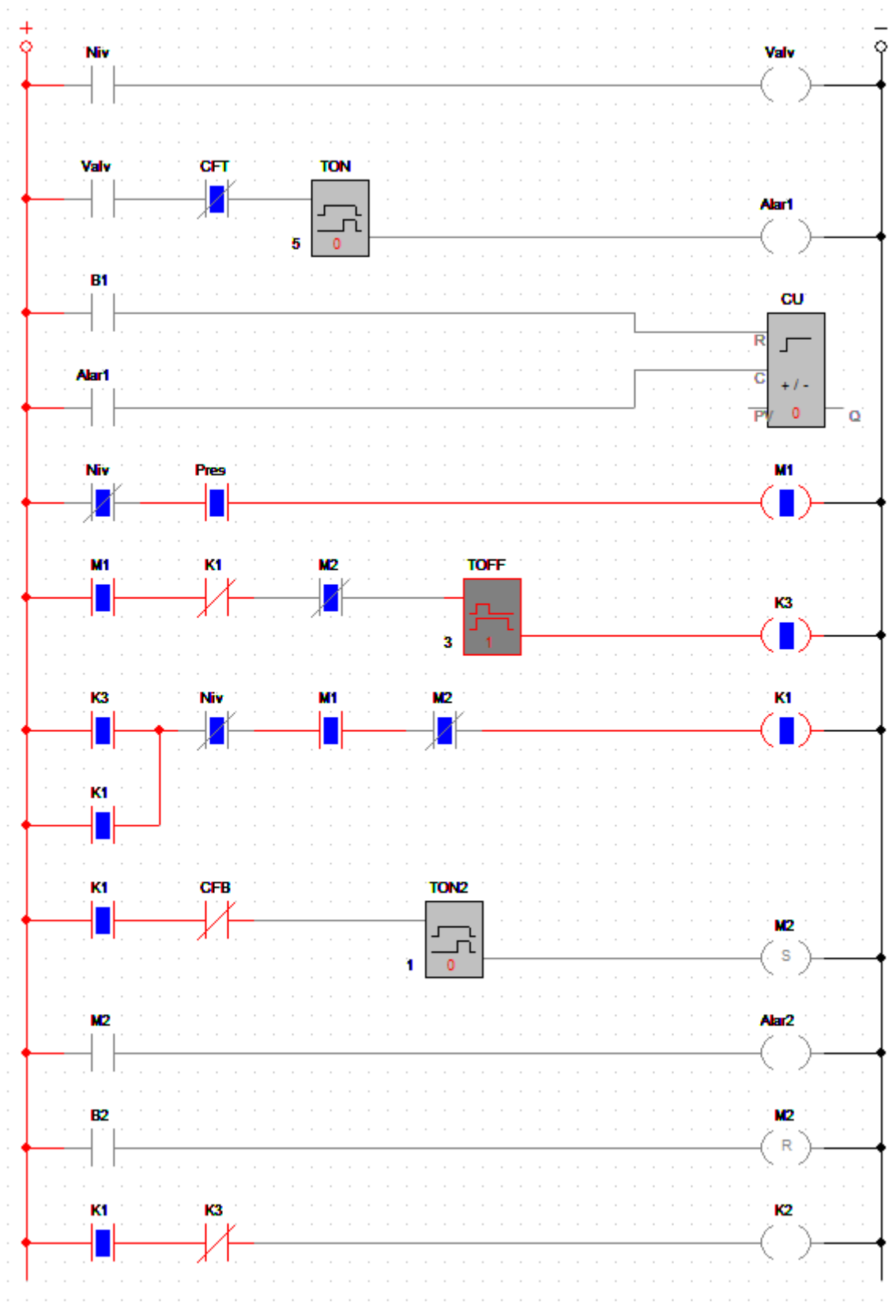
Apêndice B – Alarme de falha no reabastecimento do tanque.

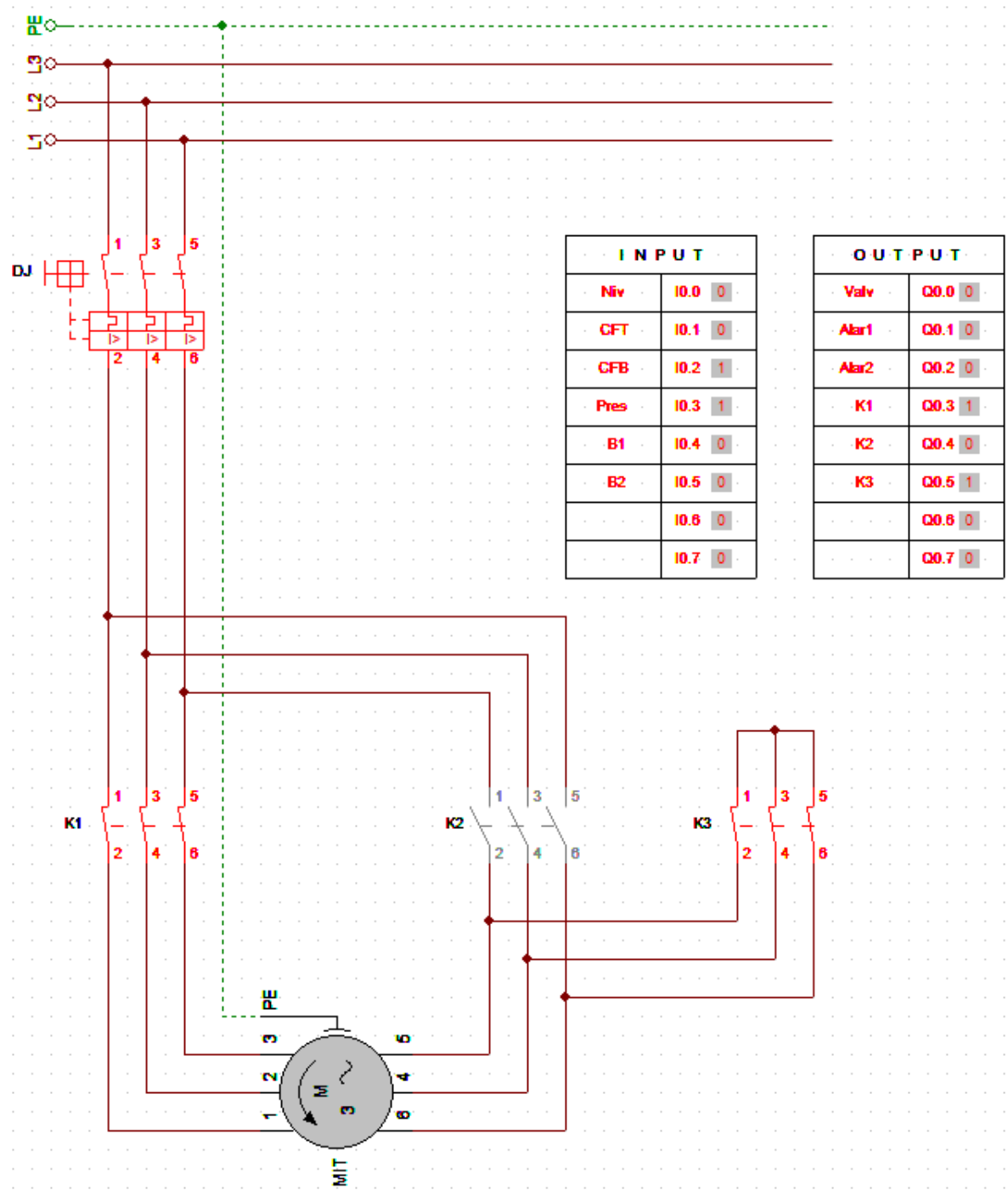


Apêndice C – Reset do contador de falhas de reabastecimento do tanque.

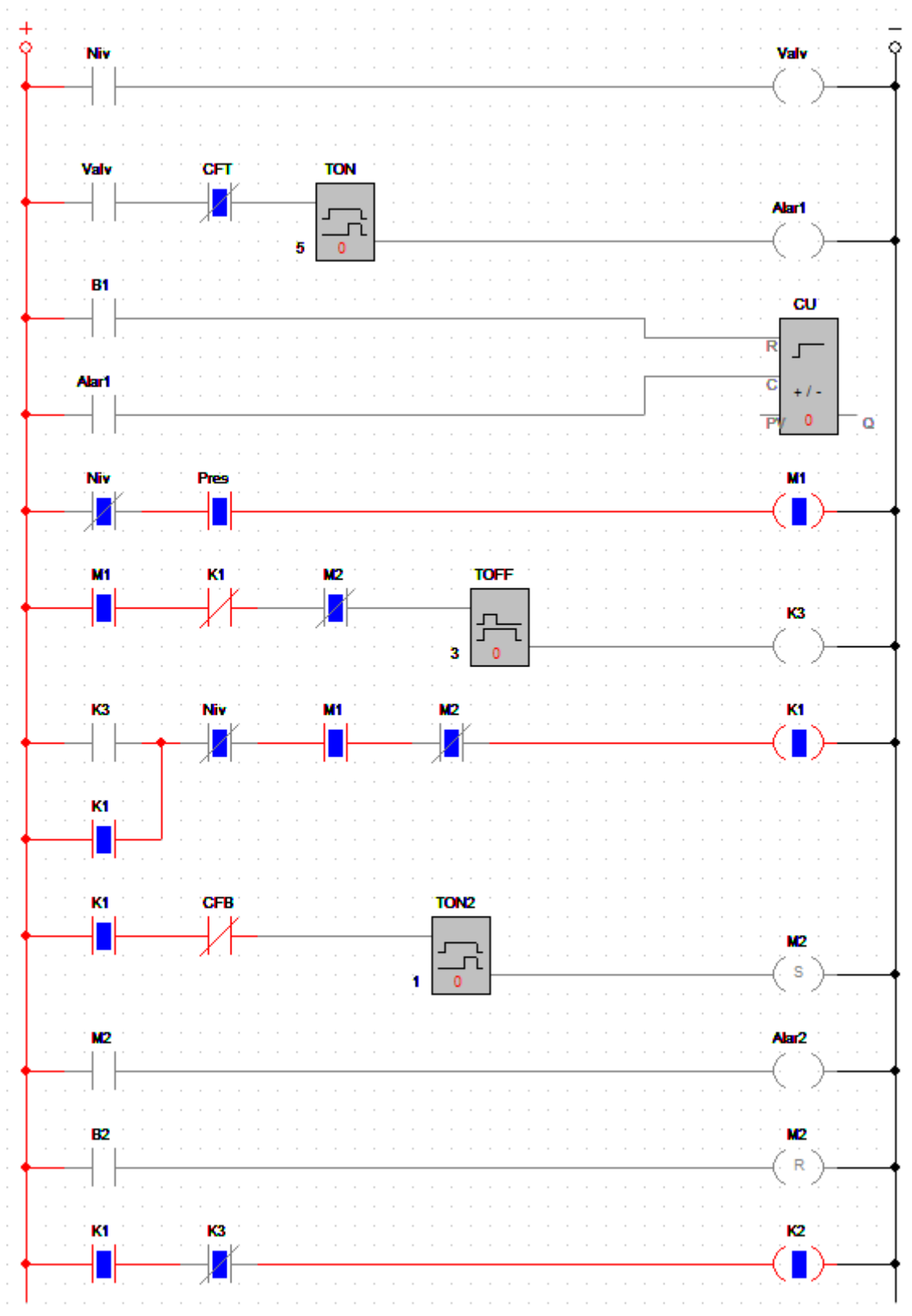


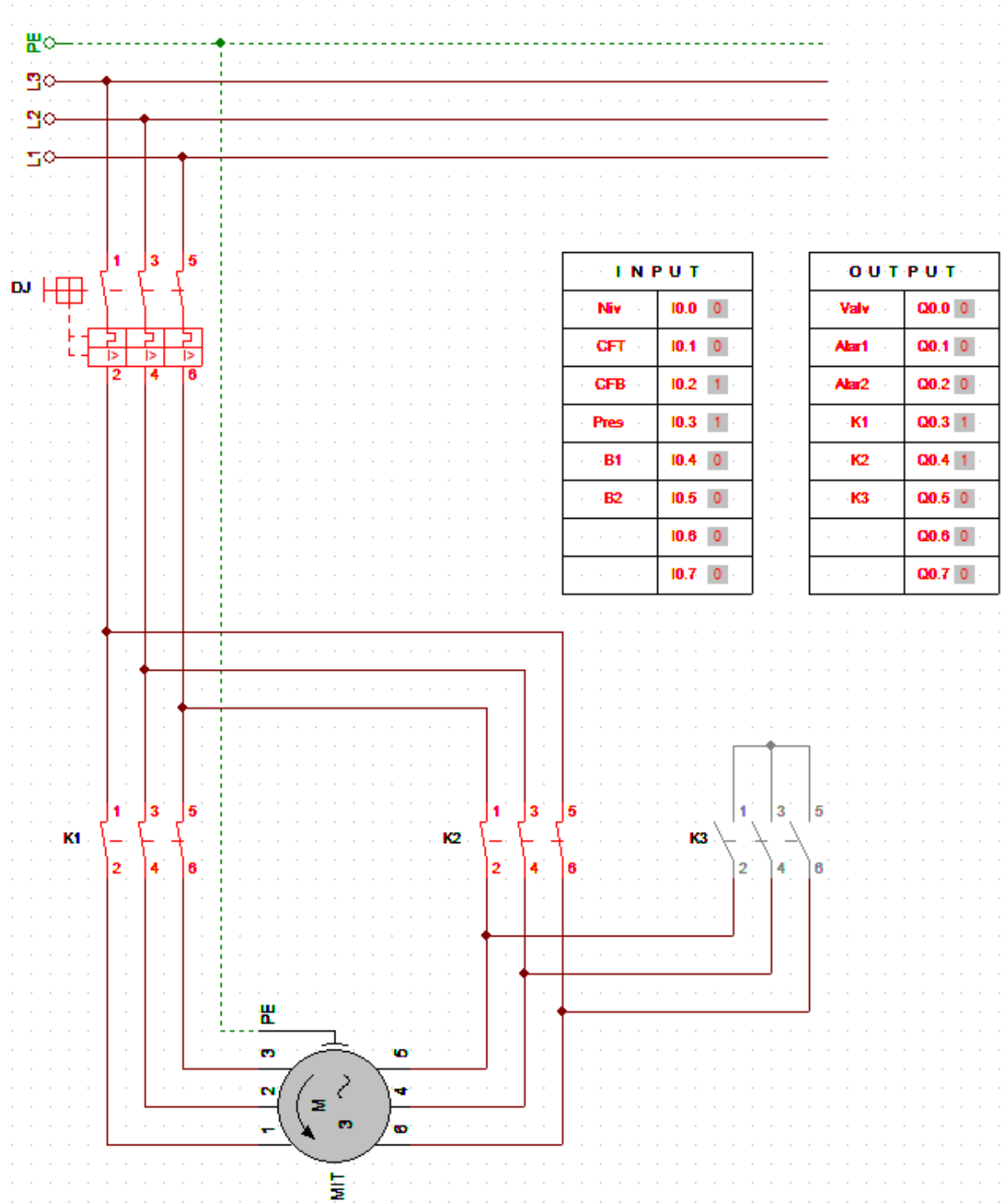
Apêndice D – Partida do motor em estrela.



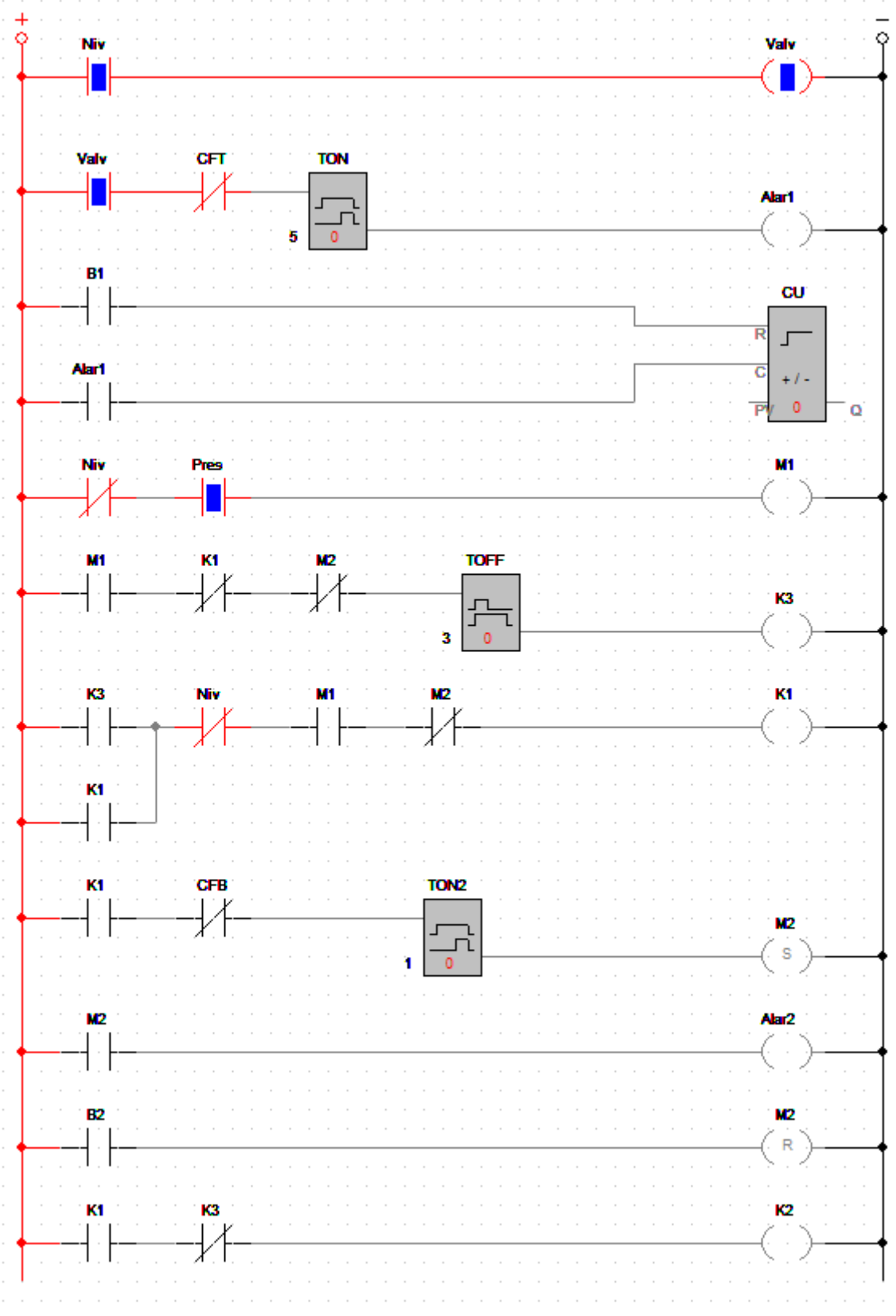


Apêndice E – Comutação do motor para triângulo.

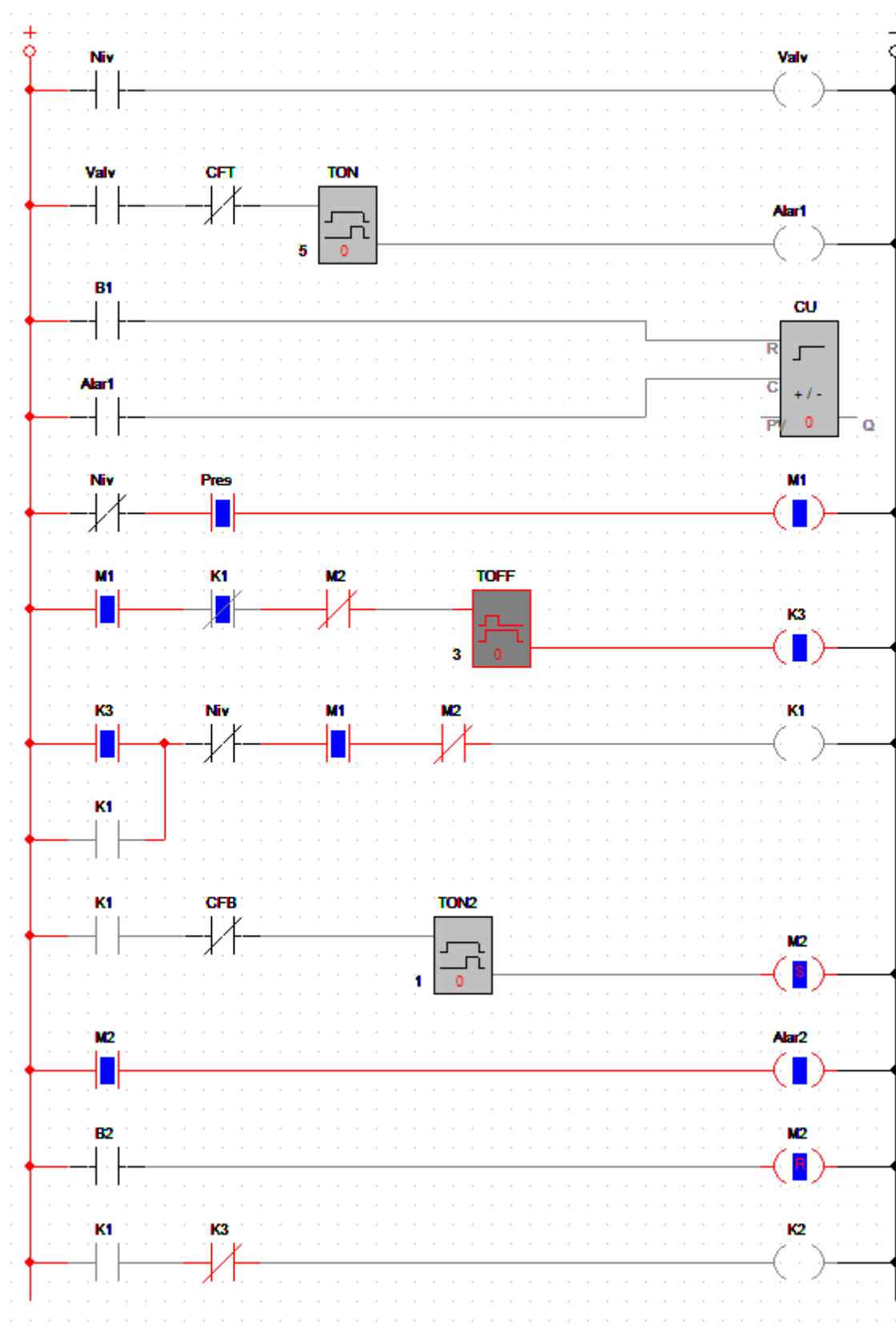




Apêndice F – Impossibilidade de partida devido ao baixo nível do tanque.



Apêndice G – Alarme e proteção contra operação a seco.



Apêndice H – Reset da falha decorrente da operação a seco.

