

## COMPARAÇÃO ENTRE AS LINGUAGENS DE PROGRAMAÇÃO UTILIZANDO O CLP

SCHWINN, Maicon Adriano<sup>1\*</sup>; TAVARES, Marcelo Sechtich <sup>2\*</sup>; DOS SANTOS,  
Cristiano Rosa<sup>3</sup>

<sup>1,2,3</sup> FAHOR, Curso de Engenharia de Controle e Automação, Faculdade Horizontina,  
Campus Arnaldo Schneider, Avenida dos Ipês, 565, Horizontina, RS, Brasil.

\*Autor Correspondente: ms002186@fahor.com.br; [mt002411@fahor.com.br](mailto:mt002411@fahor.com.br).

### RESUMO

A presente pesquisa permite conhecer alguns dados relevantes para o trabalho com programação de controladores lógicos programáveis (CLP's). Os métodos de programação descritos aqui tem como base a norma regulamentadora vigente, sendo demonstrados os tipos e alguns exemplos de utilização dos mesmos, bem como uma análise de cada tipo e algumas características positivas e negativas quanto á sua utilização. Ainda discorre sobre as funcionalidades do CLP como um todo, descrevendo as funções com as quais o processador executa uma determinada atividade, os métodos de programar em um ambiente gráfico ou textual e a conceituação da forma de programação mais simplificada para cada tipo de atividade que o controlador deverá executar.

**Palavras chave:** CLP's, Programação, Norma.

### COMPARISON BETWEEN PROGRAMMING LANGUAGES USING PLC

#### ABSTRACT

The presente research allows to know some relevant data to work with programming of programmable logic controller (PLC's). The programming methods described here are based on the current regulatory standard, showing the types and some examples of their use, as well as an analysis of each type and some positive and negative characteristics regarding their use. It also discusses the functionalities of the CLP as a whole, describing the functions with which

the processor performs a given activity, the methods of programming in a graphic or textual environment, and the conceptualization of the simplified programming form for each type of activity that the controller should perform.

**Keywords:** PLC's, Programming, Standard.

## **1 INTRODUÇÃO**

O trabalho a seguir mostra as linguagens de programação usadas na norma ICE 61131-3, como suas características e usabilidade, além de exemplos práticos. O objetivo deste trabalho é mostrar, através de experiência vista em aula, qual seria a linguagem de programação de CLP mais adequada, de acordo com sua praticidade para aprendizagem.

## **2 PROGRAMAÇÃO DE CLP**

### **2.1 CLP**

O CLP (controlador lógico programável) foi desenvolvido para comunicar processos industriais visando um desenvolvimento maior da linha de produção e um custo menor quando relacionado à mão de obra. Em suma, o CLP é utilizado para orientar a atividade de uma máquina, podendo haver uma inspeção dos resultados e passando ao operador da mesma uma maior confiança no resultado final. Com a instalação de uma linha automatizada por CLP na indústria, os operadores passaram a necessitar um maior conhecimento quanto à operação do maquinário.

### **2.2 IHM**

A IHM (Interface Homem Máquina) é um dispositivo desenvolvido para enviar e receber informações instantâneas do CLP. Este dispositivo é conectado ao CLP para transmitir ordens sem a necessidade de uma reprogramação do controlador lógico.

Além de enviar informações, esta interface de comunicação ainda recebe dados dos sensores conectados ao CLP e, conforme sua necessidade, produzem gráficos, tabelas e até pode

ser desenhada uma linha gráfica de processos industriais com comandos simplificados, para facilitar a interação com o usuário.

## **2.3 LINGUAGENS**

Visando uma ampla utilização pelos utilizadores de CLP's, foram desenvolvidas várias linguagens de programação, o que facilita a programação do equipamento conforme a facilidade que cada programador tem em uma determinada linguagem. Os tipos de programações utilizados nos CLP's podem ser de duas formas:

- Forma Gráfica: Decorre de imagens inseridas no programa formando a lógica do programa que atenda às necessidades para o qual o mesmo será aplicado. Além de prático, ainda divide-se em três tipos: Ladder, Grafcet e Blocos
- Forma Textual: Descreve uma programação como um algoritmo semelhante à linguagem de programação Phyton, sendo de alto nível pode ser dividida em duas subclasses: Texto estruturado e Lista de Instruções.

## **3 TIPOS DE PROGRAMAÇÃO**

### **3.1 IEC 61131**

Em 1992 o IEC – International Electrotechnical Commission publicou a norma IEC 61131, a qual estabeleceu padrões para Controladores Programáveis. Com isso, esta norma tem influência direta no desenvolvimento de Controladores Lógicos Programáveis e seus periféricos, tais como ferramentas de Programação e Depuração, Equipamentos de Testes e Interfaces Homem-Máquina.

Conforme a norma, os objetivos principais ao criá-la foram:

- Estabelecer critérios e características para seleção e aplicação de Controladores Programáveis (CP's).
- Especificar os requisitos mínimos para funcionalidades, condições de trabalho, características construtivas, segurança geral e testes aplicáveis para os Controladores Programáveis e seus periféricos.
- Definir regras de semântica e sintaxe para as linguagens de programação mais comuns, para que os fabricantes possam expandir e adaptar estas regras para suas próprias implementações de CLP's.

- Definir a comunicação entre CP's e outros Sistemas usando o MMS – Manufacturing Message Specification, conforme norma ISO/IEC 9508.

A norma IEC 61131 por abranger toda a normatização de configuração e desenvolvimento dos controladores lógicos programáveis, visando organizar cada etapa da regularização de CLP's subdivide-se em oito partes:

61131-1 – Informações gerais (General Overview, Definitions)

61131-2 – Requisitos de hardware (Hardware)

61131-3 – Linguagens de programação (Programming Languages)

61131-4 – Guia de orientação ao usuário (User Guidelines)

61131-5 – Comunicação (Message Service Specifications)

61131-6 – Comunicação via Fieldbus (Fieldbus Communication)

61131-7 – Programação utilizando Lógica Fuzzy (Fuzzy Logic)

61131-8 – Guia para implementação das linguagens (Implementation Guidelines)

(Catálogo ABNT-2013)

### **3.2 DIAGRAMA DE BLOCOS**

A lógica dos programas de blocos de função ou FBD (Function Block Diagram) são similares aos programas em logica ladder, exceto que o processo é visualizado na forma de blocos (PETRUZELLA, 2013). A linguagem de blocos é recomendada para quem tem o conhecimento de eletrônica digital, utilizando lógica combinacional.

O que favorece a aplicação do sistema por diagrama de blocos é a simplicidade com a qual se utilizam as propriedades do programa. Apesar deste diagrama não possuir grande versatilidade na solução de programações mais complexas, quando a questão é facilidade de programação este tipo se encaixa muito bem. Na solução de problemas mais complicados ele torna-se inadequado, por utilizar um espaço gráfico grande e necessitar de muito espaço na tela para programação. A utilização se dá pela inserção de blocos lógicos que sugerem a colocação das variáveis necessárias para a compilação, sendo possível alterar o tipo de processamento desejado no próprio bloco.

### **3.3 GRAFCET**

O modelo de programação GRAFCET ou SFC (Sequential Function Chart), é de fácil aplicação e compreensão. A lógica utilizada é de forma sequencial, sendo que ela somente

executará o próximo passo, se a condição anterior já tiver sido executada, ou seja, possui uma linearidade de execução que efetua transições sequenciais em série.

Em sua estrutura básica de programação pode-se definir que a mesma é composta por estado, onde se situa a ação que deverá ser executada no bloco definido, e a transição que é responsável pela verificação de execução da etapa para assim poder seguir com a rotina do programa, uma vez que a transição é ponto de ligação com o próximo bloco, onde será executada a próxima ação.

### **3.4 TEXTO ESTRUTURADO**

O texto estruturado ou ST (structured text) pode ser visto de forma semelhante a programação em linguagem “C”, onde é seguida a lógica do programa dentro de um loop, executando ações conforme determinadas as regras de cada uma delas. Esse modelo é geralmente utilizado quando se faz necessário tomadas de decisões, uma vez que o programa não para de rodar quando executa alguma tarefa e pode executar funções diferentes a cada novo ciclo.

### **3.5 LISTA DE INSTRUÇÕES**

Esta programação, também conhecida como IL (Instruction List) é a que possui menor custo benefício dentre as programações da 61131-3. Assemelhando-se com a estrutura de Assembly, necessita de um acumulador para armazenar as variáveis obtidas ou geradas no algoritmo. É uma linguagem de pouca utilização, para solucionar apenas problemas pequenos e de baixa complexidade com CLP's que não possuem grande capacidade devido á sua limitação.

Para realizar a programação de um componente com esta linguagem, é necessário ter uma pequena instrução sobre Assembly. A dificuldade de alteração em um código já implementado faz a utilização deste tipo ser mais restrita, porém possui uma velocidade de processamento maior devido á sua ligação com o processador do clp ser muito viável e pelo código ser semelhante aos processos executados pelo processador do clp.

### **3.6 DIAGRAMA LADDER**

O diagrama Ladder é muito utilizado devido a sua semelhança gráfica com plantas de circuitos elétricos. O diagrama Ladder (ou escada) propõe uma programação linear horizontal até o circuito desenhado entrar em contato com uma saída em cada linha, após essa linha terminar, a programação inicia-se e outra linha, semelhante a uma escada.

Esta linguagem de baixo nível possui uma familiaridade com a atuação de relés, o que facilita a sua programação e o entendimento da mesma pelos desenvolvedores de programas de CLP. O diagrama Ladder foi o primeiro dos cinco tipos de linguagens de programação para CLP's abrangidos pela IEC 61131-3 e é o mais utilizado pelos engenheiros pelo fato da linguagem Ladder basear-se em interruptores simples que se conectam a linhas com bobinas de maneira a compor circuitos lógicos. Assim, cada interruptor correspondente a uma entrada recebe uma identificação, assim como as bobinas que designam as saídas. Também é possível utilizar memórias internas, temporizadores, comparadores e blocos lógicos.

#### **4 FUNÇÕES DE PROGRAMA DE CLP**

Na programação de CLP's, deve ser desenvolvida uma solução para o problema apresentado, utilizando as operações fornecidas pelos fabricantes de Controladores Lógicos que se baseiam na norma vigente. Esta solução precisa de algumas variáveis de entrada, que são recebidas pelo CLP, entram em um processamento predefinido e são fornecidas soluções programadas, que são enviadas às saídas do componente.

Entradas são os dispositivos que enviam um sinal ao controlador, como um sensor de temperatura, por exemplo. São processadas as informações pelo controlador lógico utilizando programações baseadas em operações matemáticas em blocos de programação, adição subtração multiplicação divisão, após atingir um nível de aceitação pelo programa em execução, são enviados os sinais para as saídas do componente. O bloco de programação ainda conta com temporizadores e contadores para fazer o processamento dos sinais recebidos pelas entradas.

#### **4.1 APLICAÇÃO DE LINGUAGENS DE PROGRAMAÇÃO DE CLP**

Uma aplicação de um programa para solucionar um problema de um processo pode ser reduzida a uma tela de computador, entradas e saídas físicas, simplificando a instalação de equipamentos em uma linha industrial.

##### **4.1.1 Linguagem de texto estruturado**

Figura 1 - Linguagem ST

```
1 PROGRAM PLC_PRG
2 VAR
3     tempo: TON;
4     ROANDO: BOOL;
5     entrada: BOOL;
6     baixas: BOOL;
7     altas: BOOL;
8
9 IF start= TRUE AND entrada=0 THEN
10 entrada :=1;
11 END_IF
12 IF stop = TRUE AND entrada=1 THEN
13 entrada:=0;
14 END_IF
15 IF entrada = TRUE THEN
16     ROANDO:=TRUE;
17     TEMPO (IN:= ROANDO, PT:=T#10S);
18     IF TEMPO.Q THEN
19         IF s3 =TRUE AND s4 =TRUE THEN
20             WHILE (s1= FALSE)DO
21                 motorLE := TRUE;
22             END_WHILE
23             motorLE:=FALSE;
24         END_IF
25         IF S3= TRUE AND S4= FALSE THEN
26             WHILE (s2= FALSE) DO
27                 motorLD := TRUE;
28             END_WHILE
29             motorLD:=FALSE;
30         END_IF
31     END_IF
32 END_IF
```

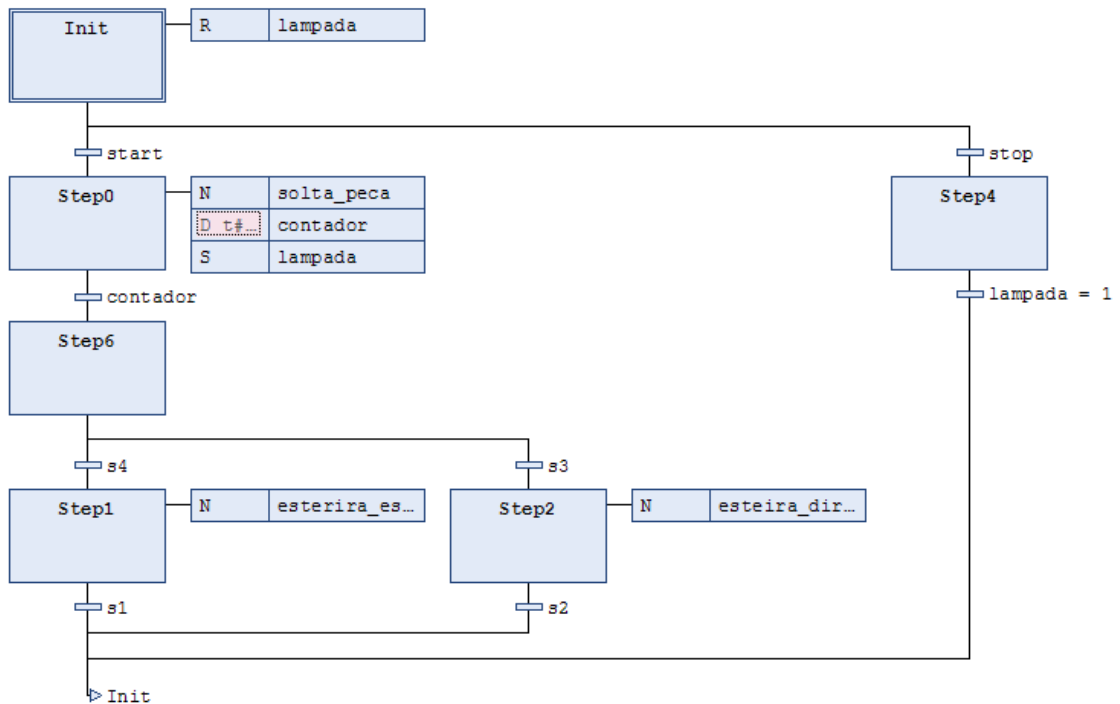
Fonte: Os Autores (2019).

A figura 1 reproduz um código do tipo texto estruturado produzido para realizar o controle de uma esteira, diferenciando os produtos que nelas são dispostos e assim alocar eles em seus locais específicos.

No modelo as condições são propostas pelo IF onde todas tem que ser satisfeitas para poder executar as ações neles dispostas. Os laços de repetições são executados pelo WHILE, para as funções serem executadas quantas vezes necessárias, até satisfazer às regras impostas por ele.

#### 4.1.2 Linguagem SFC

Figura 2 - Linguagem SFC



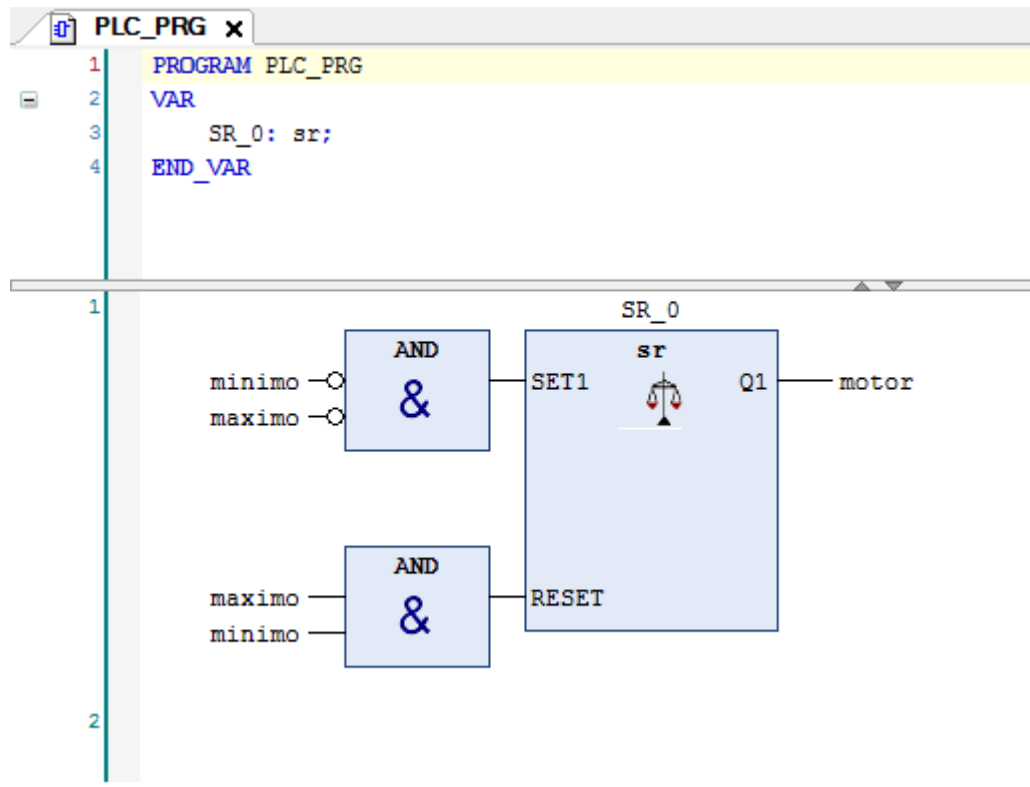
Fonte: Os Autores (2019).

A figura 2 reproduz um código do tipo GRAFCET. Este exemplo é responsável por separar peças de tamanhos diferentes, em sua estrutura as transações são para progredir de estado, podendo ser impostas condições do tipo AND, OR e outras. Assim podendo ter um melhor controle sobre o código e uma melhor eficiência, nos blocos está anexado às ações que deverão ser realizadas, em bifurcações existentes, o programa deverá realizar verificações para saber o caminho correto a seguir. O modelo é de fácil implementação e implementação lógica, mas o limitante é o fato de ser sequencial, ou seja pode ocorrer travamento caso alguma ação não seja realizada, desse modo deve-se criar o código com atenção e sempre pensando no modelo de aplicação para definir se esse é o melhor na ocasião desejada.

### 4.1.3 Linguagem de diagrama de blocos

Figura 3 - Linguagem FBD





Fonte: Os Autores (2019).

O programa mostrado na figura 3 foi desenvolvido para aplicação em uma caixa d'água, onde os sensores máximo e mínimo fazem o controle de um motor para reabastecimento da caixa. Ao receber sinal lógico 1 de ambos os sensores, o CLP seta a saída do motor como nível lógico 1, ligando o motor. Ao receber sinal lógico 0 de ambos os sensores e apenas de ambos, na entrada de sinal do CLP, o programa reseta o bloco de função SET/RESET desligando o motor.

Como pode ser notado no programa de linguagem em blocos, o desenvolvimento do software de controle é muito fácil e simples entendimento.

#### 4.1.4 Lista de instruções

Figura 4 - Linguagem IL

```

1  PROGRAM PLC_PRG
2  VAR
3  END_VAR

1  LD      START
   S      M1
   LD      S2
   AND    M1
   R      M1
   S      C1
   LD      S1
   R      C1
   LD      S3
   AND    S1
   ANDN   S2
   S      M2
   LDN    S1
   AND    S2
   AND    S3
   R      M2
  
```

Fonte: Os Autores (2019)

A figura 4 reproduz um código elaborado no formato da linguagem Lista de Instruções, o problema predisposto para ser solucionado seria de uma esteira onde haverá o corte de chapas metálicas e posteriormente as mesmas vão ser deslocadas por outra esteira. Ao pressionar Start o processo é iniciado executando todos os processos até o R(reset) do Motor 2. Esse modelo de programação é diferenciado dos demais, por ser similar ao Assembly, dessa forma tornando-se mais rápido, mas ao mesmo tempo mais complexo de ser aplicado.

### 4.1.5 Ladder

Figura 5 - Linguagem LD (Ladder).



Fonte: Os Autores (2019).

A figura 5 sugere a ligação de um sistema por um contato de entrada, fazendo um selo digital, ou seja, fazendo com que o equipamento permaneça acionado apenas a partir de um toque no botão start, sendo este sistema interrompido por um botão stop. A fácil aplicação e entendimento desta linguagem faz dela a mais utilizada pelos desenvolvedores de programas para CLP.

## CONCLUSÃO

Conforme decorreram os desenvolvimentos dos exemplos de aplicação supracitados, definimos que a aplicação dos programas em LADDER e em GRAFCET são os melhores aplicáveis às operações com CLP, devido às suas simplicidades, desenvolvimento em ambiente gráfico e grande capacidade de processamento de informações, além de ter proporcionado uma lógica fácil e rápida de ser compreendida, podendo ser adequada para a maioria dos programadores de CLP.

Na verdade, a linguagem apropriada depende do programador, principalmente a sua formação, e a funcionalidade do programa que irá fazer.

## REFERÊNCIAS

DESCONHECIDO, Autor. **Programação do CLP**. Disponível em: <<https://ensinandoeletrica.blogspot.com/2017/07/programacao-do-clp.html>> Acesso em: 22 de abril de 2019.

MORAES, Everton. **As 5 formas de programação do CLP**. Disponível em: <<https://www.saladaeletrica.com.br/programacao-do-clp/>> Acesso em: 24 de abril de 2019.

PETRUZELLA, Frank D. **CONTROLADORES LÓGICOS PROGRAMÁVEIS**. 4º Ed. Amgh Editora, 2013.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Norma Técnica IEC 61131-3.2013**. Disponível em: <<https://www.abntcatalogo.com.br/norma.aspx?ID=195941>> Acesso em: 24 de abril de 2019.