PLCopen®
for efficiency in automation

IEC 61131-3: a norma para programação

(this document is based on the 2nd edition of IEC 61131-3)

IEC 61131-3 é o primeiro esforço real para a padronização das linguagens de programação para a automação industrial. Como este é um apelo mundial, esta é uma norma independente de qualquer empresa.

IEC 61131-3 é a terceira parte da família IEC 61131. Esta consiste de:

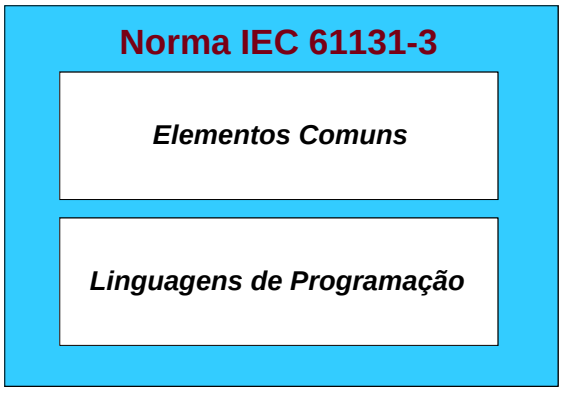
- Parte 1 General Overview
- Parte 2 Hardware
- Parte 3 Programming Languages
- Parte 4 User Guidelines
- Parte 5 Communication

Existem muitas formas de entender a parte 3 da norma. Vamos identificar algumas:

- é o resultado da Força Tarefa 3, Linguagens de Programação, dentro do IEC TC65 SC65B
- é o resultado do trabalho árduo de 7 empresas internacionais somando dezenas de anos de experiência no campo da automação industrial
- aprox. 200 páginas de texto, com cerca de 60 tabelas, incluindo tabelas de características
- é a especificação da sintaxe e semântica de uma suíte unificada de linguagens de programação, incluindo o modelo geral de software e uma linguagem de estruturação.

Outra elegante forma é dividir a norma em duas partes (vide figura 1):

1. Elementos Comuns (Common Elements)
2. Linguagens de programação (Programming Languages)



Vamos olhar para estas partes com mais detalhes:

Elementos Comuns

Tipagem de Dados

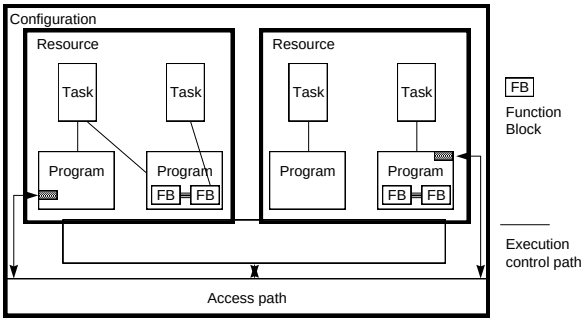
Dentro dos elementos comuns, os tipos de dados são definidos. A tipagem de dados previne erros na fase inicial. É usada para definição do tipo de qualquer parâmetro usado. Isto evita, por exemplo, a divisão de uma data por um inteiro. Os tipos de dados comuns são Boolean, Integer, Real, Byte e Word, mas também Date, Time_of_Day e String. Baseado nisto, é possível definir os nossos tipos de dados pessoais, chamados de tipos derivados. Desta forma, pode-se definir uma entrada analógica como tipo de dado e reutilizá-la inúmeras vezes.

Variáveis

Variáveis são associadas somente para endereços explícitos de hardware (entradas e saídas por ex.) nas configurações, recursos e programas. Desta forma, cria-se um alto nível de independência do hardware, proporcionando a reutilização do software.
O escopo das variáveis é normalmente limitado à unidade de organização nas quais elas são declaradas (escopo local).
Isto significa que os nomes delas podem ser reutilizados em outras partes sem nenhum conflito, eliminando outra fonte de erros muito comum, dados corrompidos pelo programa. Se as variáveis tiverem escopo global, estas devem ser declaradas como tal (VAR_GLOBAL).
A cada parâmetro pode ser atribuído um valor inicial na partida a quente e a frio do sistema, de forma a se garantir os valores corretos.

Configuração, Recursos e Tarefas

Para melhor entendimento, vamos observar o modelo de software, como definido pela norma (veja a seguir):



No nível mais alto, o software deve resolver um problema particular de controle que pode ser formulado como uma Configuração (*Configuration*). Uma configuração é específica para um sistema de controle particular, incluindo a disposição do hardware,

recursos de processamento, endereçamento de memória para I/O e demais capacidades do sistema.
Dentro da configuração pode-se definir um ou mais recursos (*Resources*). Pode-se entender um recurso como um elemento com capacidade de processamento dos programas IEC.
Dentro de um recurso, uma ou mais tarefas (*Tasks*) podem ser definidas. Tarefas controlam a execução de um conjunto de programas ou blocos funcionais. Estas podem ser executadas periodicamente ou quando da ocorrência de um evento específico, tal como a mudança de uma variável.
Programas (*Programs*) são constituídos de um número de diferentes elementos escritos usando qualquer uma das linguagens definidas pela IEC. Tipicamente, um programa consiste de uma rede de Funções (*Functions*) e Blocos Funcionais (*Function Blocks*), os quais são capazes de trocar dados. Funções e Blocos Funcionais são os blocos básicos de construção, contendo uma estrutura de dados e um algoritmo.

Vamos fazer uma comparação com um CLP convencional: este contém um recurso, executando uma tarefa, controlando um programa, processando de forma cíclica. A IEC 61131-3 acrescenta muito mais capacidade, tornando-o aberto para o futuro. Um futuro que inclui multi-processamento e programas disparados por eventos. E este futuro não está longe: basta olhar para os sistemas distribuídos ou sistemas de controle de tempo-real. A IEC 61131-3 é apropriada para uma ampla faixa de aplicações, sem a necessidade de se aprender linguagens de programação adicionais.

Unidades de Organização de Programas

Na IEC 61131-3, os Programas, Blocos Funcionais e Funções são chamadas de Unidades de Organização de Programas (POUs).

Funções

A IEC definiu funções padrões e funções definidas pelos usuários. Funções padrões são, por exemplo, ADD(ition), ABS (absolute), SQRT, SINus e COSinus. Funções definidas pelo usuário, uma vez definidas, podem ser usadas inúmeras vezes.

Blocos Funcionais, FBs

Blocos Funcionais são equivalentes aos circuitos integrados, CIs, representando uma função de controle especializada. Estes contêm dados e um algoritmo, de modo que eles preservam os estados passados (uma das principais diferenças das Funções). Possuem também uma interface bem definida e escondem o seu conteúdo, tais como os CIs. Desta forma, os FBs proporcionam uma separação clara entre diferentes níveis de programadores e equipes de manutenção.

Uma malha de temperatura (PID) é um excelente exemplo de Bloco Funcional. Uma vez definido, este pode ser usado inúmeras vezes, no mesmo programa, em diferentes programas, ou mesmo em diferentes projetos, tornando-o altamente reutilizável.
Blocos Funcionais podem ser escritos em qualquer uma das linguagens IEC, e em muitos casos mesmo na linguagem "C". Neste sentido, eles podem ser definidos pelo usuário. Blocos Funcionais Derivados são baseados nos FBs padrões, mas também nos completamente novos, FBs customizados são permitidos pela norma: isto estabelece um *framework*.

As interfaces de funções e blocos funcionais são descritas da mesma forma:

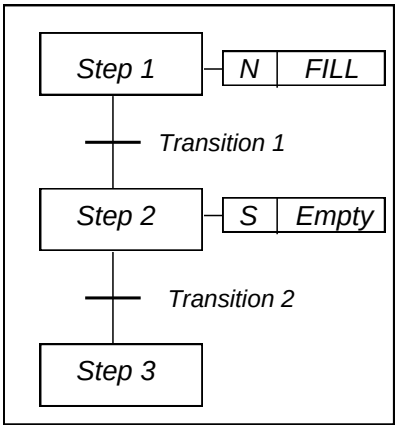
```
FUNCTION_BLOCK Example

VAR_INPUT:
      X :   BOOL;
      Y :   BOOL;
END_VAR

VAR_OUTPUT
      Z :   BOOL;
END_VAR

    (* statements of functionblock body *)

END_FUNCTION_BLOCK
```

As declarações acima descrevem a interface para um bloco funcional com dois parâmetros booleanos de entrada e um parâmetro booleano de saída.

Programas

Com os anteriormente mencionados blocos básicos de construção, podemos dizer que um programa é uma rede de Funções e Blocos Funcionais. Um programa pode ser escrito em qualquer uma das linguagens de programação definidas.

Sequenciamento Gráfico de Funções, SFC



O SFC descreve graficamente o comportamento seqüencial de um programa de controle. É derivado das redes de Petri e da norma IEC 848 Grafcet, com as alterações necessárias para converter a representação de uma documentação padrão para um conjunto de elementos de controle de execução.
O SFC estrutura a organização interna do programa e ajuda a decompor o problema de controle em partes gerenciáveis, enquanto mantém a sua visão geral.
O SFC consiste de Passos, interligados com blocos de Ações e Transições. Cada passo representa um estado particular do sistema sendo controlado. Uma transição é associada com uma condição, a qual, quando

verdadeira, causa a desativação do passo anterior à mesma e a ativação do passo seguinte. Passos são ligados com blocos de ações, desempenhando uma determinada ação de controle. Cada elemento pode ser programado em qualquer linguagem IEC, incluindo o próprio SFC.
É possível o uso de seqüências alternativas e mesmo paralelas, tais como as normalmente usadas em aplicações de bateladas. Por exemplo, uma seqüência é usada para o processo primário, a segunda para a monitoração das restrições operacionais.
Devido a sua estrutura geral, o SFC funciona também como uma ferramenta de comunicação, integrando pessoas de diferentes formações, departamentos e países.

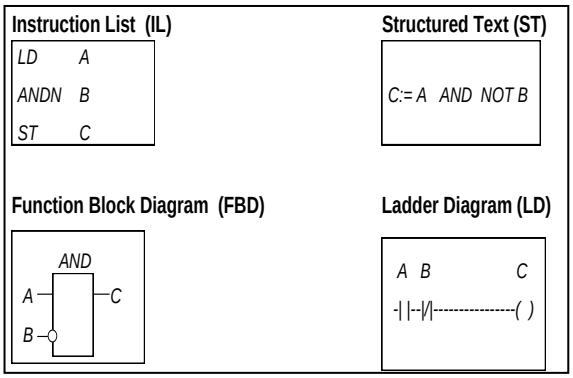
Linguagens de Programação

São definidas quatro linguagens pela norma. Isto significa que suas sintaxes e semânticas foram definidas, eliminando a chance de dialetos. Uma vez aprendidas, é possível o uso de uma variedade de sistemas baseados nesta norma.
As linguagens consistem de 2 textuais e 2 gráficas:
Textuais

- Lista de Instruções, IL
- Texto Estruturado, ST

Gráficas:

- Diagrama Ladder, LD
- Diagrama de Blocos Funcionais, FBD



Na figura anterior, as quatro linguagens descrevem a mesma lógica de programa.
A escolha da linguagem de programação depende:

- da formação do programador
- do problema a resolver
- do nível da descrição do problema
- da estrutura do sistema de controle
- da interface com outras pessoas/departamentos

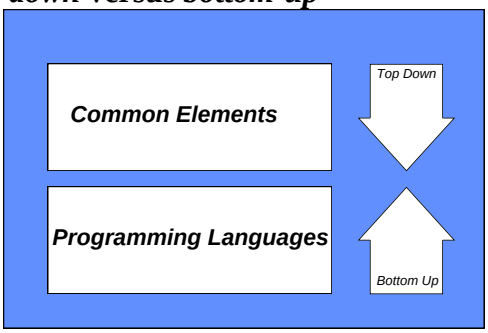
Todas as quatro linguagens são interligadas: elas proporcionam uma plataforma comum, com uma ligação com a experiência existente. Neste sentido, elas também funcionam como uma ferramenta de comunicação, integrando pessoas de diferentes formações.

Diagrama Ladder tem sua origem nos EUA. É baseada na representação gráfica da Lógica de Relés.
Lista de Instruções é a contraparte européia. Como uma linguagem textual, se assemelha ao assembler.
Diagrama de Blocos Funcionais é muito usada na indústria de processos. Expressa o comportamento de funções, blocos funcionais e programas como um conjunto de blocos gráficos interligados, como nos diagramas de circuitos eletrônicos. Se parece com um sistema em termos do fluxo de sinais entre elementos de processamento.
Texto Estruturado é uma linguagem de alto nível muito poderosa, com raízes em Ada, Pascal e "C". Contém todos os elementos essenciais de uma linguagem de programação moderna, incluindo condicionais (IF-THEN-ELSE e CASE OF) e iterações (FOR, WHILE e REPEAT). Estes elementos também podem ser aninhados. Esta linguagem é excelente para a definição de blocos funcionais complexos, os quais podem ser usados em qualquer outra linguagem IEC.

Exemplo em ST:

```
I:=25;
WHILE J<5 DO
        Z:= F(I+J);
END_WHILE

IF B_1 THEN
        %QW100:= INT_TO_BCD(Display)
ENDIF

CASE TW OF
        1,5:    TEMP := TEMP_1;
        2:      TEMP := 40;
        4:      TEMP := FTMP(TEMP_2);
ELSE
        TEMP := 0;
        B_ERROR :=1;
END_CASE
```

Top-down versus *bottom-up*



A norma também permite duas formas para o desenvolvimento de programas: *top-down* e *bottom-up*. É possível especificar toda uma aplicação dividindo-a de cima para baixa em parte menores, declarando as variáveis, e assim por diante. Ou pode-se começar de baixo, a partir de funções e blocos funcionais derivados, por exemplo. Em ambos os caso, o ambiente de desenvolvimento irá auxiliá-lo durante todo o processo.

Implementações

O atendimento completo às exigências da norma IEC 61131-3 não é simples. Por esta razão, a norma permite

a implementação parcial em vários aspectos. Isto cobre o número de linguagens, funções e blocos funcionais suportados. Também dá uma liberdade maior para os fabricantes, mas os usuários devem ficar atentos durante o processo de seleção. Novas versões podem também ter um nível de implementação dramaticamente alto.

Muitos ambientes de programação IEC atuais oferecem tudo esperado para os modernos ambientes: uso do mouse, menus, telas gráficas, suporte para múltiplas janelas, uso de hipertexto, verificação durante a programação, etc.
Fique atento, pois estas são características não definidas pela norma: é um ponto onde os fabricantes podem se diferenciar.

Conclusão

As implicações técnicas da norma IEC 61131-3 são muitas, deixando bastante espaço para crescimento e diferenciação. Isto torna esta norma propensa para evoluir muito neste século.
A norma IEC 61131-3 causará um grande impacto em toda indústria de controle industrial. Certamente a norma não ficará restrita para o mercado de CLPs convencionais. Atualmente, a norma já é adotada no mercado de Motion Control, sistemas distribuídos e sistemas de controle baseados em PC/Softlogic, incluindo pacotes SCADA. E as áreas de aplicação continuam crescendo.
Ter uma norma sobre uma ampla área de aplicação proporciona muitos benefícios para os usuários e programadores. Os benefícios da adoção da norma são vários, dependendo da área de aplicação. Alguns exemplos são:

- redução do desperdício de recursos humanos, no treinamento, depuração, manutenção e consultoria
- destinar maior atenção para a solução de problemas através da reutilização de software em alto nível
- eliminação de erros e dificuldade de entendimento
- utilização de melhores técnicas de programação em um ambiente mais amplo: indústria de controle em geral
- combinação de diferentes componentes de diferentes programas, locais, empresas e mesmo países

Melhorias da norma pela PLCopen

A **PLCopen** é uma associação mundial independente de produtos e fabricantes que suporta a norma IEC 61131-3. Através da implementação desta norma em muitos ambientes de desenvolvimento, os usuários poderão escolher entre diferentes marcas e tipos de controladores com pouquíssimo treinamento e intercambiar aplicações com o mínimo de esforço.
A organização PLCopen trabalha com a promoção da adoção da norma, assim como realçar a norma no sentido técnico. Este último inclui certificações e intercâmbio.
Membros da PLCopen podem participar dos comitês, e como tal ter informações em avanço, uma forte identidade, assim como influenciar nos resultados.
Além disso, eles podem usar a logomarca da PLCopen para a sua promoção. Os comitês que trabalham com a PLCopen e seus resultados são descritos resumidamente a seguir.

Resultados Técnicos

Os comitês técnicos, TCs, com representantes dos membros da PLCopen, trabalham em itens específicos.
Dentro do **TC1** – **Normas**, a PLCopen coleta propostas de seus membros para o grupo de trabalho IEC 65B WG7, desenvolve uma posição conjunta, e distribui as informações relativas. Isto foi objetivado especificamente na segunda edição da norma, a qual foi publicada no início de 2003.

TC2 - Funções define bibliotecas comuns de Funções e Blocos Funcionais para áreas de aplicação específicas. Um exemplo é a biblioteca de Blocos Funcionais para aplicações de Motion Control. Esta padronização integra os aspectos de segurança e controle de movimento no controle industrial. Como tal, proporciona uma visualização e entendimento comum para os usuários: programadores assim como equipes de implantação e manutenção. Com múltiplas implementações desta biblioteca, a reutilização de software e a escala dos sistemas de controle são muito mais fáceis, mesmo através de diferentes arquiteturas e marcas de controladores. O intercâmbio de partes de programas através do Nível Reutilização da PLCopen (veja a seguir para maiores detalhes) desempenha um importante papel aqui.

Certificação e Testes de Conformidade

TC3 – Certificação define um sistema de certificação para os Ambientes de Suporte a Programação, PSE (ambientes de desenvolvimento). Cada PSE pode ser testado para mostrar que é compatível com um subconjunto da norma IEC 61131-3 especificado pela PLCopen. A norma possui um grande número das chamadas Tabelas de Características e a PLCopen definiu quais elementos destas tabelas devem ser suportados para atender às exigências de compatibilidade. Adicionalmente, a PLCopen estende a norma para suportar a reutilização dos blocos funcionais definidos pelo usuário (derivados) entre diferentes PSEs.

Nível Conformidade, CL. Com a ampla faixa de áreas de aplicação da norma IEC 61131-3, nem todas implementações utilizam exatamente os mesmos tipos de dados. Para comportar isto, a certificação de acordo com o Nível Conformidade, CL, implica que o fornecedor da PSE selecione os tipos de dados pelo seu produto em concordância com a declaração de conformidade. Todas as características da IEC são testadas. Isto significa que embora o teste seja to tipo Sim/Não (conformidade e não-conformidade), podem existir diferenças entre dois produtos certificados. Estas diferenças podem influenciar a reutilização dos blocos funcionais definidos pelo usuário.

O número total de tipos de dados definidos pela norma é 26 (tabelas 10 e 12 da norma). Esta faixa vai desde sinais digitais simples ON/OFF (BOOL) até estruturas potencialmente complexas. Portanto, CL tem 26 opções: tipo de dado suportado e não-suportado. Somente os tipos de dado suportados são usados para teste.

Adicionalmente, o **Nível Reutilização, RL**, é dedicado a fazer com que as unidades de programação Função e Blocos Funcionais sejam reutilizáveis em diferentes PSEs. Isto é feito através do intercambio em uma forma de texto simples da linguagem ST. Para representação em outra linguagem, uma ferramenta de conversão de/para ST pode ser incluída. Esta é a maior, mas natural, contribuição da PLCopen para a norma IEC 61131-3.

Historicamente, existe um nível de entrada chamado de **Nível Básico, BL**, para mostrar o comprometimento com a norma. Embora mais limitado, é possível desenvolver aplicações baseadas nele. BL proporciona um ponto de entrada para os fabricantes, mostrando o comprometimento deles com a norma. Para os usuários, este nível proporciona uma interpretação uniforme da norma, especialmente importante se eles tem que trabalhar com sistemas de diferentes fabricantes.

Especificações detalhadas da maioria das linguagens da norma IEC 61131-3 estão quase prontas. O trabalho continua para as linguagens restantes. Os procedimentos de teste de compatibilidade e credenciamento para os

laboratórios de teste foram estabelecidos. Laboratórios independentes têm sido credenciados e um crescente número de produtos tem sido certificados. Para um lista completa consulte o website www.plcopen.org.

TC4 – Comunicações trabalha na relação entre a comunicação e as linguagens de programação, tal como o mapeamento do Profibus e CANopen via IEC 61131-5 sobre a IEC 61131-3.

TC5 – Software Seguro prepara recomendações para a aplicação da norma IEC 61131-3 numa adaptação para aplicações de alta confiabilidade (Safety Systems), especialmente para as novas normas para sistemas seguros IEC 61508 e 61511.

TC6 – XML trabalha na especificação de esquemas XML para todas as linguagens, assim como projetos inteiros. Esta especificação irá proporcionar a base para intercâmbio, assim como a integração com outras ferramentas de software, incluindo ferramentas de desenvolvimento de alto nível, ferramentas de documentação e verificação.

Eventos Promocionais

Uma importante tarefa da PLCopen é informar usuários/programadores sobre os benefícios da padronização da programação de controle industrial. Isto é feito via:

- o website da PLCopen: www.plcopen.org ;
- publicação de um folheto gratuito (*newsletter*) impresso e eletrônico, chamado "PLCopening";
- publicações na imprensa;
- participação em conferências e mostras;
- organização de conferencias e seminários próprios, como o ICP em Outubro;

Os Comitês Promocionais **PC1, PC3 e PC4** suportam estas atividades.
Membros da PLCopen são mais bem vistos no mercado como apoiadores de padrões abertos.

PC2 – Treinamento Comum definiu uma base comum para treinamento. Esta inclui certificação. Neste sentido, centros de treinamento certificados para treinamento na norma IEC 61131-3 podem ser facilmente identificados.

Benefícios da Afiliação

A afiliação na PLCopen tem muitos benefícios para fabricantes e institutos. A PLCopen suporta fortemente a comunidade de usuários. Para isto, foram criadas categorias de afiliação adicionais. Através da afiliação com a PLCopen, faz-se uma declaração clara do seu comprometimento com a norma IEC 61131-3, se tornando mais visível como tal, podendo-se usar o logo da PLCopen, e ter acesso a informações em avanço assim como influenciar no trabalho realizado.

Para maiores informações, favor acessar o website www.plcopen.org, assim como a revista eletrônica, para a qual você pode se inscrever no website.

Tradução:
Marcos Fonseca
Diretor da Divisão de Tecnologia da Automação – ATAN Sistemas
www.atan.com.br