Engenharia Eletrotécnica e de Computadores

Norma IEC61131-3

Automação avançada

a11611 - Ricardo Rodrigues

a10227 - José Rodrigues

Resumo

Este trabalho consiste na apresentação e estudo da norma de desenvolvimento de código industrial IEC 61131-3.

Neste documento é explicado como a norma foi criada qual o seu objetivo e explicado em que consiste.

Palavras Chave ST, FBD, IL, DL, SFC

Índice

Índice de Figuras

1 Introdução

Este trabalho foi realizado no âmbito da unidade curricular de Automação Avançada presente no plano curricular do Mestrado em Engenharia Eletrónica e de Computadores da Escola Superior de Tecnologia do Instituto Politécnico do Cávado e do Ave. E consiste na análise da norma IEC 61131-3 [3], a sua criação, em que consiste e o que esta pretende normalizar.

Esta norma deverá ser de conhecimento comum para todos os programadores de controladores programáveis de modo a tirar partido de todas as possíveis funcionalidades de cada tipo de linguagem de modo a que seja possível e utilizada a reutilização de código de modo a tornar as funções genéricas.

2 IEC 61131-3

Como a evolução do mercado foram sendo criados vários tipos e modelos de equipamentos dedicados a automação industrial criando assim uma grande variedade de equipamentos e como consequência incompatibilidade de características entro os diferentes modelos.

De modo a resolver o problema de incompatibilidades entre os diferentes equipamentos e marcas, a International Electrotechnical Commission (IEC) decidiu criar um standard de desenvolvimento de modo a normalizar os controladores lógicos programáveis, o hardware a ser utilizado, testes, documentação, programação e comunicação.

Depois de um grande estudo e concordância entre os diferentes desenvolvedores de sistemas automáticos industriais, a secção 3 do IEC 61131 teve o objetivo primaz de desenvolver um novo padrão de linguagens de programação de controladores programáveis. Foi o primeiro esforço internacional que teve resultados ao estabelecer um standard para as linguagens de programação da automação industrial.

Na seguinte tabela 1 é apresentado as diferentes partes constituintes da norma IEC 61131 [3].

Tabela 1 - Partes da norma IEC 61131

Parte	Título	Descrição	Data de publicação
Parte 1	General Information	Definição da terminologia e conceitos	2003 (2ª Ed.)
Parte 2	Equipment requirments and tests	Testes de verificação e produção eletrónica e mecânica	2003 (2ª Ed.)
Parte 3	Progrmmable Languages	Estrutura de software do PLC, linguagens e execução de programas	2003 (2ª Ed.)

Parte 4	User guidelines	Recomendação para a seleção, instalação e manutenção dos PLCs	2004 (2ª Ed.)
Parte 5	Communications	Funcionalidades para comunicação com outros dispositivos.	2000 (1ª Ed.)
Parte 6	Reserved		
Parte 7	Fuzzy Control Programming	Funcionalidades de software, incluindo blocos funcionais padrões para tratamento de lógica nebulosa dentro dos PLCs	2000 (1ª Ed.)
Parte 8	Guidelines for the Application and Implementation of Programming Languages	Orientações para implementação das linguagens IEC 1131-3	2003 (2ª Ed.)

A parte 3 da norma IEC 61131 será aqui descrita, sendo então estudado a Padronização Internacional de Linguagens, estrutura de software e execução de programas em PLCs.

De modo a entender a evolução da norma é apresentado na seguinte figura 1 um cronograma temporal que expõem a evolução da IEC 61131-3.

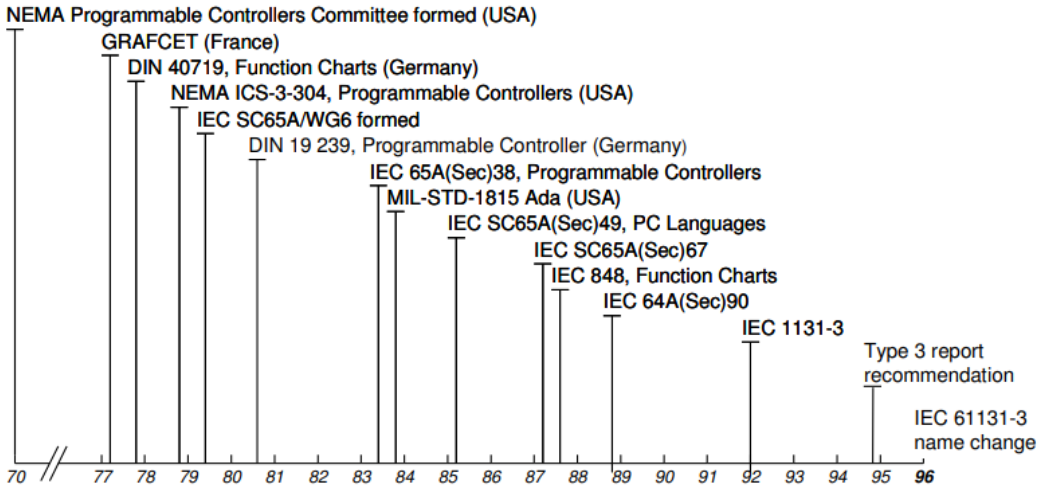


Figura 1 - Evolução do standerd das linguagens de programação IEC 61131-3 [1]

2.1 Linguagens de programação na IEC 61131

A norma IEC 61131-3 apresenta o standard de 5 tipos básicos de linguagens que são encontradas nos controladores programáveis:

- Linguagens Textuais
  - Texto Estruturado (Structured Text – ST)
  - Lista de Instruções (Instruction List – IL)
- Linguagens Gráficas
  - Diagrama Ladder (LD)
  - Diagrama de Blocos Funcionais (Functional Block Diagram -FBD)
- Linguagem Sequencial
  - Sequência gráfica de funções (Sequential Function Chart – SFC)

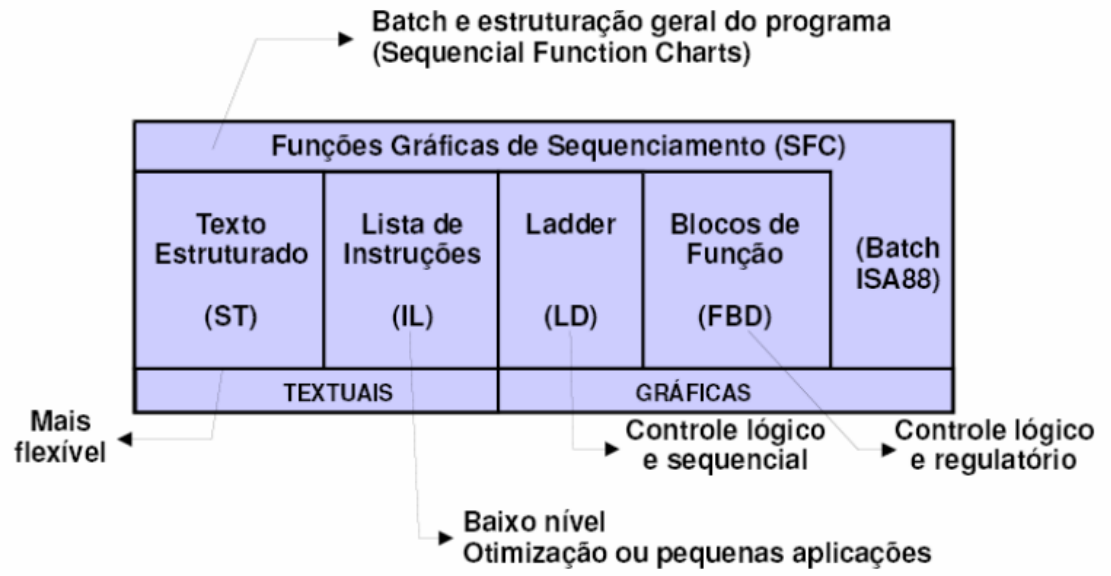


Figura 2 - Linguagens de programação IEC 61131-3 [2]

2.1.1 Texto Estruturado – ST

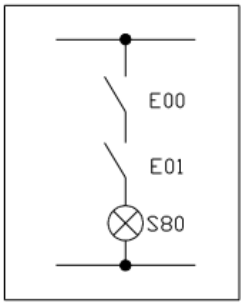
É uma linguagem de alto nível muito poderosa, com raízes em Pascal e "C". Contém todos os elementos essências de uma linguagem de programação moderna, incluindo condicionais (IF-THEN-ELSE e CASE OF) e iterações (FOR, WHILE e REPEAT).

```
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
```

Figura 3 - Exemplo Texto Estruturado [1]

2.1.2 Lista de Instruções – IL

A lista de instruções consiste de uma sequência de comandos padrões correspondentes a funções. Assemelha-se a linguagem Assembler. O programa representado pela linguagem descritiva "Se as entradas E00 e E01 estiverem ligadas, então ligar saída S80" Pode ser representado em lista de instruções por:




```
      A E00      : Contato E00
AND   A E01      : EM SÉRIE Contato E01
      = S80      : = Acionamento de saída S80
```

2.1.3 Diagrama Ladder – LD

A linguagem Ladder é, conforme mencionado anteriormente, a linguagem de programação de PLCs mais comum e a mais difundida, é também conhecida como lógica de diagrama de contatos, pois se assemelha à tradicional notação de diagramas elétricos e de painéis de controlo a relés. O mesmo esquema elétrico apresentado no exemplo anterior pode ser representado em diagrama Ladder por:

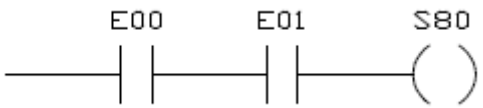


Figura 4 - Exemplo código com Diagrama Ladder [1]

2.1.4 Diagrama de Blocos Funcionais – FBD

O diagrama funcional é uma forma gráfica de representação de instruções ou comandos que devem ser executados. É baseado em blocos funcionais, por exemplo, uma porta AND. Estes blocos são em geral utilizados dentro de lógicas ladder. O programa representado pela linguagem descritiva "Se as entradas E00 e E01 estiverem ligadas, então ligar saída S80" pode ser representado em blocos funcionais por:

Figura 5 - Exemplo código Diagrama de Blocos Funcionais [1]

2.1.5 Sequência gráfica de funções – SFC

O SFC descreve graficamente o comportamento sequencial de um programa de controlo e é derivado das técnicas de modelagem por Redes de Petri e da norma IEC 848 que define o padrão Grafcet.

O SFC tem as características necessárias para uma conversão eficaz de um modelo com um padrão de representação num conjunto de elementos de controlo de execução adequados a projetos de automação.

No global o SFC consiste em passos interligados com blocos de ações e transições. Cada passo representa um estado do sistema. Cada elemento ou programa nesta linguagem pode ser programado em qualquer linguagem textual ou gráfica.

Como a estrutura do SFC é mais adequada a projetos de automação com programas de alto porte, o SFC funciona também como uma ferramenta de comunicação entre as equipas de projeto, fazendo com que pessoas com diferentes formações, departamentos, e até países comuniquem com uma linguagem que é facilmente entendida por todos.

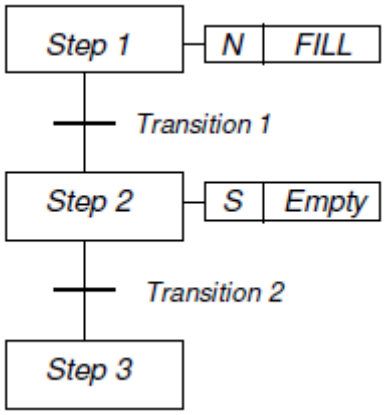


Figura 6 - Exemplo código Sequência gráfica de funções [1]

2.2 Estrutura do software e execução dos programas na IEC 61131

Alguns elementos na norma são usados por todas as linguagens de programação IEC 61131 referidas na seção 2.1. Uma das coisas mais importantes na programação de qualquer sistema é a capacidade de separar o software em vários componentes.

A norma prevê a possibilidade de serem desenvolvidos ambiente de programação ou IDEs capazes de decompor os programas em diferentes módulos, os quais devem seguir um padrão na interface de comunicação entre os vários módulos ou componentes. O modelo de software consiste num conjunto de conceitos que definem uma infraestrutura para a decomposição dos módulos do projeto de automação.

2.3 Modelo de Software

No contexto de um programa para um PLC, todo o programa tem de interagir com o ambiente onde está inserido, ou seja, não é possível definir a estrutura de um PLC sem o conhecimento das interfaces e como vai ser feito o controlo dos atuadores.

Quando o PLC está em modo de execução (run mode), são necessárias as seguintes interfaces:

- **Interfaces de Input/Output**: permitem o acesso aos dispositivos ou cartas de I/O para a leitura de sinais como: pressões, níveis, temperaturas, etc…, assim como comandar os atuadores.
- **Interfaces de comunicação**: utilizadas quando sistemas externos precisão de comunicar com o PLC, como HMI's, camaras, ou outros PLC.
- **Interfaces de sistema**: corresponde à interface entre o programa do PLC e o hardware do mesmo.

A seguinte Figura 7 mostra o modelo do software IEC 61131.

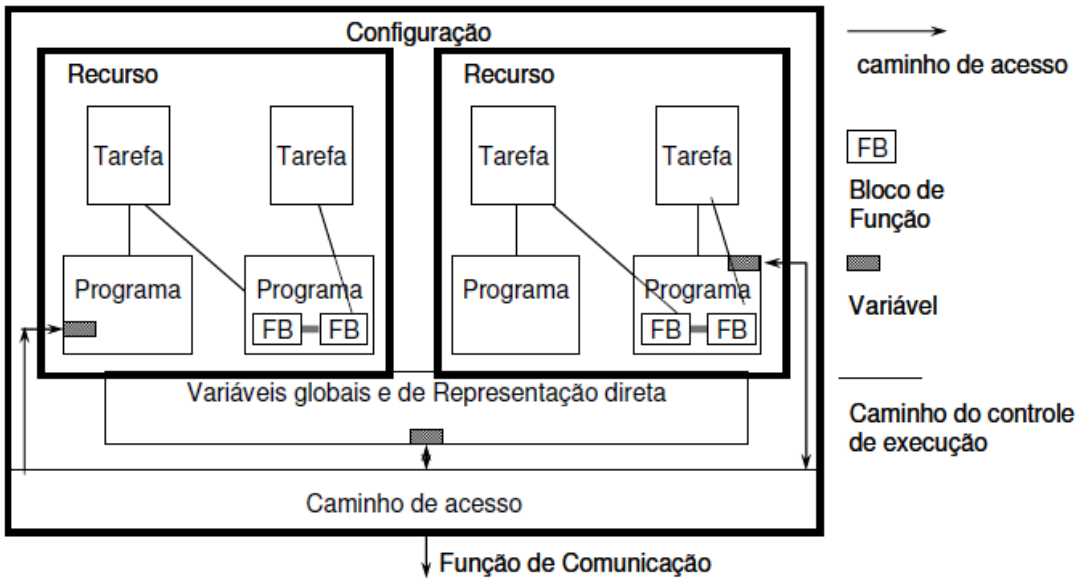


Figura 7 - Modelo de software IEC 61131-3 [1]

2.4 Configuração

Num nível superior, o software de um sistema de controlo tem uma determinada configuração. Cada configuração corresponde ao software necessário para um único PLC. No entanto, nos sistemas mais complexos é possível existir várias configurações, ou apenas uma configuração para vários PLCs, os quais interagem entre si com interfaces de comunicação padrão definido pela norma.

A "configuração" no âmbito da norma IEC não são os passos para a definição de parâmetros de um sistema, ou seja, configuração/setup de um sistema.

2.5 Recursos

Dentro de cada configuração podem existir um ou mais recursos. Um recurso é basicamente qualquer elemento com a capacidade de processamento, responsável pela execução dos programas. Uma característica dos recursos é que eles definem a divisão do software em módulos, mas também podem definir uma divisão no hardware. Cada recurso deve ser independente, não necessitando de outros recursos para funcionar corretamente.

2.6 Programas

Um programa IEC pode ser ter por vários componentes de software estritos em qualquer uma das diferentes linguagens da norma.

Tipicamente, um programa consiste num código executável, capaz de fazer a troca de dados entre outros programas. Um programa pode aceder às variáveis do PLC e comunicar com outros programas. A execução de diferentes partes de um programa pode ser controlada utilizando *Tasks*.

2.7 Blocos Funcionais

O conceito de blocos funcionais é um dos conceitos mais importantes da norma IEC61131, que permite estruturar de forma hierárquica o software.

A utilização de blocos de software facilita a reutilização dos mesmos. As principais características dos blocos funcionais são que os blocos têm um conjunto de dados, os quais podem ser alterados por um algoritmo interno. Apenas alguns dados são mantidos em memória para uma determinada instância do bloco funcional.

2.8 Funções

As funções são elementos do software que não possuem persistência, existindo apenas durante o tempo de execução, ou seja, produzem sempre o mesmo resultado.

As funções têm apenas um output (não considerando a saída ENO (Enable output) para o controlo da execução). O resultado das funções pode ser de um tipo de dados simples, mas com múltiplos elementos, como arrays, vetores ou estruturas.

2.9 Reutilização de Programas, Blocos Funcionais e Funções

Pela norma, programas, blocos funcionais e funções são considerados POUs. A principal finalidade destes elementos é a possibilidade da reutilização através de instâncias. Desta forma, a reutilização pode ser em macro, por programas, ou em microescala por blocos funcionais. A recursividade não é permitida dentro de uma POU por motivos de estabilidade e de segurança da aplicação.

A utilização de Function Blocks e standard functions é feita através de bibliotecas fornecidas pelo fabricante do PLC, ou pela criação de blocos e funções especificas definidas pelo utilizador.

Tabela 2 - Exemplo de Reutilização de POUs

Tipo	POU aplicada como	Descrição
Programa	Instância de um Programa	Permite a reutilização ao novel macro.
Bloco Funcional	Instância de um Bloco Funcional	Permite a reutilização de estratégia de controlo e algoritmos, como o controlo de PID, filtros, motores, etc.
Funçao	Função	Usada para tratamento comum de dados, como a lógica And, Or, seno, cosseno, soma, etc.

2.10 Tasks

Uma Task é um mecanismo muito útil para sistema de tempo real, onde os programas ou blocos funcionais são executados periodicamente, ou em resposta a um evento (mudança de estado de alguma variável), permitindo a execução de programas com diferentes ciclos.

O objetivo de executar programas com taxas diferentes é atender às exigências do tempo de resposta de um processo controlado e de otimizar o uso da capacidade de processamento do PLC.

A norma IEC assume que as tasks em diferentes recursos são executadas de forma independente. No entanto, em alguma implementação pode ser necessário a utilização de mecanismos de sincronização.

A norma IEC não define nenhum mecanismo implícito para a execução de programas, ou seja, um programa só é executado se for associado a alguma Task ativa que corra periodicamente, ou por um determinado evento (trigger).

Existem 2 tipo de Tasks:

- Não-preemptiva

  É uma task que termina sempre o seu processamento independentemente se sofre interrupções ao longo da execução. O intervalo entre execução destas tasks pode variar.

- Preemptiva

  É uma Task recomendada para sistemas que necessitam de apresentar comportamentos determinísticos no tempo.

  Neste tipo de sistema de Tasks quando o intervalo de uma Task de maior prioridade vence, a Task em execução é suspensa e a nova Task de maior prioridade passa a ser executada de imediato. Logo que a Task de maior prioridade termine a Task suspensa continua a sua execução.

2.11 Declaração de Variáveis

Podem ser declaradas variáveis Locais ou Globais, A norma exige a declaração de variáveis dentro de diferentes elementos de software, como Programas e Blocos Funcionais. As variáveis podem utilizar nomes com significado idênticos e serem de diferentes tipos de dados.

Podem ser de alocação dinâmica ou associadas a posições de memória (representação direta). O scope das variáveis é local ao elemento de software que as tem declarada permitindo acesso dentro do próprio elemento que pode ser uma Configuração, Recurso, Programa, Bloco Funcional ou Função.

As variáveis também podem ser de scope global, sendo acedidas por todos os elementos.

2.12 Caminhos de Acesso

Os caminhos de acesso permitem a transferência de dados entre diferentes configurações. Cada configuração pode definir um número de variáveis para acesso por configurações remotas. A norma assume que estarão disponíveis mecanismos de comunicação para troca de informações, não abordando a forma a ser adotada.

2.13 Fluxo de Controlo

A norma IEC não define os mecanismos para o controlo de execução dos elementos de software, os quais são dependentes da implementação. Entretanto, são definidos os comportamentos na partida e parada do sistema:

- Partida:

Quando uma configuração parte, todas as variáveis globais são inicializadas e todos os recursos são iniciados.
Quando um recurso parte, todas as variáveis dentro do recurso são inicializadas e todas tarefas são habilitadas.
Uma vez ativadas as tarefas, todos os programas e blocos funcionais associados às mesmas executarão quando a tarefa estiver ativa.

- Parada:

Quando uma configuração para, todos os recursos da mesma param.
Quando um recurso para, todas as tarefas são desabilitadas, interrompendo a execução dos programas e blocos funcionais.
Deve ser observado que um programa somente controla a execução dos blocos funcionais associados à mesma tarefa. Entretanto, os blocos funcionais podem ser associados a tarefas distintas, não sendo necessariamente sincronizados com os programas.

3 Conclusões

No final desta pesquisa foram aqui identificados os principais pontos focados na norma IEC 61131-3. Verificou-se que a norma 61131-3 identifica as linguagens de programação a serem utilizadas, o tipo de variáveis que deve ser utilizado, a estrutura do software e execução dos programas, o modelo de Software entre outros aspetos relacionados com o desenvolvimento de software para PLC.

A realização deste trabalho prático mostrou-se bastante útil para saber que existe uma entidade que se preocupou em normalizar o modo de programação de controladores lógicos programáveis, que existe uma normalização de modo a que seja possível alternar entre diferentes produtores e o modo de programação e comunicação seja idêntico.

Bibliografia

[1] Descrição Norma IEC 61131, shorturl.at/sBKP3D, USP. [Acedido em 12/01/2020].

[2] Especificação e Implantação IEC 61131, shorturl.at/qrHJP, UFES. [Acedido em 12/01/2020].

[3] Norma IEC 61131 Second edition 2003-01, https://d1.amobbs.com/bbs_upload782111/files_31/ourdev_569653.pdf, [Acedido em 03/01/2020].