IEC 61131-3: un recurso de programación estandar

En la actualidad aún siguen persistiendo sistemas de control específicos del fabricante, con programación dependiente y conexión compleja entre distintos sistemas de control. Esto significa para el usuario costos elevados, escasa flexibilidad y falta de normalización en las soluciones al control industrial.

IEC 61131 es el primer paso en la estandarización de los autómatas programables y sus periféricos, incluyendo los lenguajes de programación que se deben utilizar. Esta norma se divide en cinco partes:

- Parte 1: Vista general.
- Parte 2: Hardware.
- Parte 3: Lenguaje de programación.
- Parte 4: Guías de usuario.
- Parte 5: Comunicación.

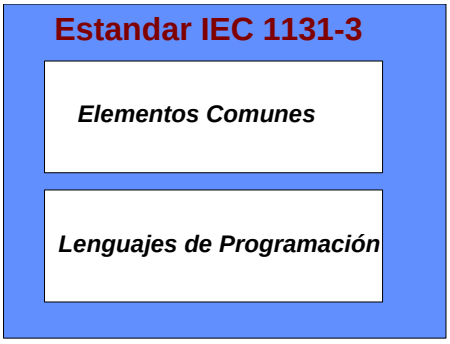
IEC 61131-3 pretende es la base real para estandarizar los lenguajes de programación en la automatización industrial, haciendo el trabajo independiente de cualquier compañía.

Hay muchas maneras de describir el trabajo desarrollado en la tercera parte de esta norma, indicaremos algunas de ellas son:

- IEC 61131-3 es el resultado del gran esfuerzo realizado por 7 multinacionales a los que se añaden muchos años de experiencia en el campo de la automatización industrial.
- Incluye 200 páginas de texto aproximadamente, con mas de 60 tablas.
- IEC 61131-3 son las especificaciones de la sintaxis y semántica de un lenguaje de programación, incluyendo el modelo de software y la estructura del lenguaje.

Otra visión distinta es dividir el estándar en dos partes: (ver figura 1):

- Elementos comunes.
- Lenguajes de programación.



Elementos Comunes

Tipos de datos

Dentro de los elementos comunes, se definen los tipos de datos. Los tipos de datos previenen de errores en una fase inicial, como por ejemplo la división de un dato tipo fecha por un número entero. Los tipos comunes de datos son: variables booleanas, número entero, número real, byte y palabra, pero también fechas, horas del día y cadenas (strings).

Basado en estos tipos de datos, el usuario puede definir sus propios tipos de datos, conocidos como tipos de datos derivados. De este modo, se puede definir por ejemplo un canal de entrada analógica como un tipo de dato.

Variables

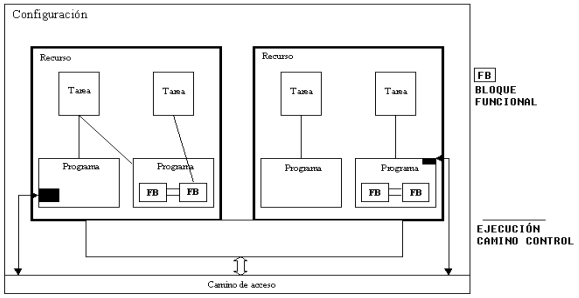
Las variables permiten identificar los objetos de datos cuyos contenidos pueden cambiar, por ejemplo, los datos asociados a entradas, salidas o a la memoria del autómata programable. Una variable se puede declarar como uno de los tipos de datos elementales definidos o como uno de los tipos de datos derivados. De este modo se crea un alto nivel de independencia con el hardware, favoreciendo la reusabilidad del software.

La extensión de las variables está normalmente limitada a la unidad de organización en la cual han sido declaradas como locales. Esto significa que sus nombres pueden ser reutilizados en otras partes sin conflictos, eliminando una frecuente fuente de errores. Si las variables deben tener una extensión global, han de ser declaradas como globales utilizando la palabra reservada VAR_GLOBAL.

Pueden ser asignados parámetros y valores iniciales que se restablecen al inicio, para obtener la configuración inicial correcta.

Configuración, recursos y tareas

Para entender ésto mejor, vamos a ver el modelo de software, que define IEC 61131-3 (ver figura 2).



Al más alto nivel, el elemento software requerido para solucionar un problema de control particular puede ser formulado como una *configuración*. Una configuración es específica para un tipo de sistema de control, incluyendo las características del hardware: procesadores, direccionamiento de la memoria para los canales de I/O y otras capacidades del sistema.
Dentro de una configuración, se pueden definir uno o más *recursos*. Se puede entender el recurso como un procesador capaz de ejecutar programas IEC.
Con un recurso, pueden estar definidas una o más *tareas*. Las tareas controlan la ejecución de un conjunto de programas y/o bloques de función. Cada una de ellos puede ser ejecutado periódicamente o por una señal de disparo especificada, como el cambio de estado de una variable.
Los *programas* están diseñados a partir de un diferente número de elementos de software, escrito en algunos de los distintos lenguajes definidos en IEC 61131-3. Típicamente, un programa es una interacción de *Funciones* y *Bloques Funcionales*, con capacidad para intercambiar datos. Funciones y bloques funcionales son las partes básicas de construcción de un programa, que contienen una declaración de datos y variables y un conjunto de instrucciones..
Comparado esto con un PLC convencional, éste contiene un solo recurso, ejecutando una tarea que controla un único programa de manera cíclica. IEC 61131-3 incluye la posibilidad de disponer de estructuras más complejas. El futuro que incluye multi-procesamiento y gestión de programas por eventos ¡Y no está muy lejos!, observar simplemente las características de los sistemas distribuidos o los sistemas de control de tiempo real. IEC 61131-3 está disponible para un amplio rango de aplicaciones, sin tener que conocer otros lenguajes de programación adicionales.

Unidades de Organización de Programa

Dentro de IEC 1131-3, los programas, bloques Funcionales y funciones se denominan Unidades de Organización de Programas, *POU´s*.

Funciones

IEC 61131-3 especifica funciones estándar y funciones definidas por usuario. Las funciones estándar son por ejemplo ADD (suma), ABS (valor absoluto), SQRT (raíz cuadrada), SIN (seno), y COS (coseno). Las funciones definidas por usuario, una vez implementadas pueden ser usadas indefinidamente en cualquier POU. Las funciones no pueden contener ninguna información de estado interno, es decir, que la invocación de una función con los mismos argumentos (parámetros de entrada) debe suministrar siempre el mismo valor (salida).

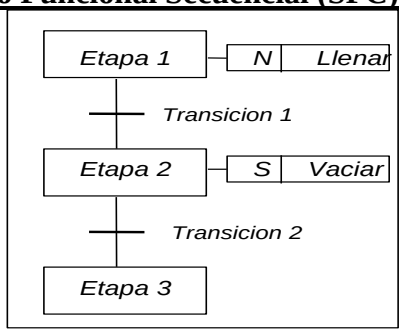
Bloques Funcionales, FB´s

Los bloques funcionales son los equivalentes de los circuitos integrados, IC´s, que representan funciones de control especializadas. Los FB´s contienen tanto datos como instrucciones, y además pueden guardar los valores de las variables (que es una de las diferencias con las funciones). Tienen un interfaz de entradas y salidas bien definido y un código interno oculto, como un circuito integrado o una caja negra. De este modo, establecen una clara separación entre los diferentes niveles de programadores, o el personal de mantenimiento. Un lazo de control de temperatura, PID, es un excelente ejemplo de bloque funcional. Una vez definido, puede ser usado una y otra vez, en el mismo programa, en diferentes programas o en distintos proyectos. Esto lo hace altamente reutilizable.
Los bloques funcionales pueden ser escritos por el usuario en alguno de los lenguajes de la norma IEC, pero también existen FB´s estándar (biestables, detección de flancos, contadores, temporizadores, etc.). Existe la posibilidad de ser llamados múltiples veces creando copias del bloque funcional que se denominan *instancias*. Cada instancia llevará asociado un identificador y una estructura de datos que contenga sus variables de salida e internas.

Programas

Los *programas* son "un conjunto lógico de todos los elementos y construcciones del lenguaje de programación que son necesarios para el tratamiento de señal previsto que se requiere para el control de una máquina o proceso mediante el sistema de autómata programable". Un programa puede contener, aparte de la declaración de tipos de datos, variables y su código interno, distintas instancias de funciones y bloques funcionales.

Gráfico Funcional Secuencial (SFC)



SFC describe gráficamente el comportamiento secuencial de un programa de control. Esta definición deriva de las Redes de Petri y Grafcet (IEC 848), con las modificaciones adecuadas para convertir las representaciones de una norma de documentación en un conjunto de elementos de control de ejecución para una POU de un autómata programable.
SFC ayuda a estructurar la organización interna de un programa, y a descomponer un problema en partes manejables, manteniendo simultáneamente una visión global. Los elementos del SFC proporcionan un medio para subdividir una POU de un autómata programable en un conjunto de etapas y transiciones interconectadas por medio de enlaces directos. Cada etapa lleva asociados un conjunto bloques de acción y a cada transición va asociada una condición de transición que cuando se cumple, causa la desactivación de la etapa anterior a la transición y la activación de la siguiente. Los bloques de acción permiten realizar el control del proceso. Cada elemento puede ser programado en alguno de los lenguajes IEC, incluyéndose el propio SFC. Dado que los elementos del SFC requieren almacenar información, las únicas POU´s que se pueden estructurar utilizando estos elementos son los bloques funcionales y los programas.

Se pueden usar secuencias alternativas y paralelas, comúnmente utilizadas en muchas aplicaciones. Debido a su estructura general, de sencilla comprensión, SFC permite la transmisión de información entre distintas personas con distintos niveles de preparación y responsabilidad dentro de la empresa.

Lenguajes de Programación

Se definen cuatro lenguajes de programación normalizados. Esto significa que su sintaxis y semántica ha sido definida, no permitiendo particularidades distintivas (dialectos). Una vez aprendidos se podrá usar una amplia variedad de sistemas basados en esta norma.

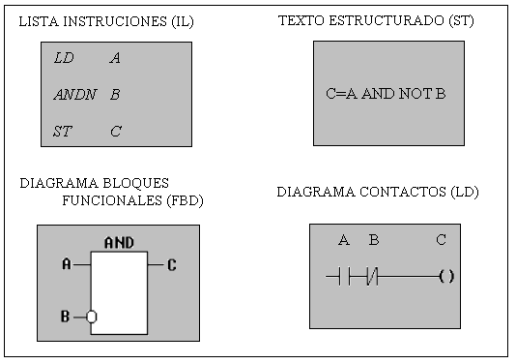
Los lenguajes consisten en dos de tipo literal y dos de tipo gráfico:

Literales:

- Lista de instrucciones (IL).
- Texto estructurado (ST).

Gráficos:

- Diagrama de contactos (LD).
- Diagrama de bloques funcionales (FBD).



En la figura superior, los cuatro programas describen la misma acción. La elección del lenguaje de programación depende de:

- los conocimientos del programador,
- el problema a tratar,
- el nivel de descripción del proceso,
- la estructura del sistema de control,
- la coordinación con otras personas o departamentos.

Los cuatros lenguajes están interrelacionados y permiten su empleo para resolver conjuntamente un problema común según la experiencia del usuario.

El Diagrama de contactos (LD) tiene sus orígenes en los Estados Unidos. Está basado en la presentación gráfica de la lógica de relés. *Lista de Instrucciones* (IL) es el modelo de lenguaje ensamblador basado un acumulador simple; procede del alemán ´Anweisungliste, AWL.

El *Diagramas de Bloques Funcionales* (FBD) es muy común en aplicaciones que implican flujo de información o datos entre componentes de control. Las funciones y bloques funcionales aparecen como circuitos integrados y es ampliamente utilizado en Europa. El lenguaje *Texto estructurado* (ST) es un lenguaje de alto nivel con orígenes en el Ada, Pascal y ´C´; puede ser utilizado para codificar expresiones complejas e instrucciones anidadas; este lenguaje dispone de estructuras para bucles (REPEAT-UNTIL; WHILE-DO), ejecución condicional (IF-THEN-ELSE; CASE), funciones (SQRT, SIN, etc.).

Top-down vs. Bottom-up-

La norma también permite dos formas de desarrollar tu programa de control (ver figura 5): de arriba a abajo (Top-down) y de abajo a arriba (bottom-up). Puedes especificar inicialmente la aplicación completa y dividirla en partes, declarar las variables y demás. También puedes comenzar la programación desde abajo, por ejemplo, por medio de funciones y bloque funcionales. Por cualquiera de los caminos que elijas, IEC 61131-3 te ayudará durante todo el proceso.

Implementaciones

Cumplir todos los requerimientos de la norma IEC 61131-3 no es fácil, por eso se permiten implementaciones parciales en varios aspectos. Esto hace referencia al número de lenguajes que soportan las herramientas de desarrollo disponibles, y al número de funciones y de bloques funcionales. Con ello se deja libertad al suministrador, pero el usuario debe tener cuidado durante el proceso de selección de la herramienta adecuada. Incluso una actualización del software puede dar lugar a un nivel muy alto de trabajo durante la implementación.

Muchos entornos de programación IEC actuales ofrecen aquello que se espera a nivel de interface de usuario: uso de ratón, menús desplegables, pantallas de programación gráfica, múltiples ventanas, ayuda en línea, verificación durante el diseño, etc. Debe hacerse notar que estos detalles no están especificados en la norma por lo que es una de las partes donde los proveedores pueden diferenciarse.

Conclusiones

Las implicaciones técnicas de la norma IEC 61131-3 son altas, dejando bastante espacio para el crecimiento y la diferenciación. Esto la hace adecuada para entrar óptimamente en el próximo siglo.

La norma IEC 61131-3 tendrá un gran impacto en el mundo del control industrial y éste no se restringe al mercado convencional de los PLC´s. Ahora mismo, se pueden ver adoptada en aplicaciones para control de movimiento, sistemas distribuidos y sistemas de control basados en PC (SoftPLC), incluyendo los paquetes SCADA. Y las áreas de su utilización siguen creciendo.

El uso de IEC 61131-3 proporciona numerosos beneficios para usuarios/programadores. Los beneficios de la adopción de este estándar son varios, dependiendo de las áreas de aplicación: control de procesos, integrador de sistemas, educación, programación,

mantenimiento, instalación, etc. Vamos a nombrar sólo algunos de estos beneficios:

1. Se reduce el gasto en recursos humanos, formación, mantenimiento y consultoría.
2. Evita las fuentes habituales de problemas por el alto nivel de flexibilidad y reusabilidad del software.
3. Las técnicas de programación son utilizables en amplios sectores (control industrial en general).
4. Combinan adecuadamente diferentes elementos que pueden provenir de diferentes fabricantes, programas, proyectos...
5. Incrementa la conectividad y comunicación entre los distintos departamentos y compañías.

El estándar IEC 61131-3 es una realidad en papel. Ahora los usuarios que aprecian los beneficios del estándar deben demandar productos que cubran sus necesidades, de modo que las empresas proveedoras puedan amortizar los gastos de desarrollo de las herramientas adecuadas: *´el problema del huevo y la gallina´*.

Para mas información, contactar directamente con PLCopen o a través de la World Wide Web. La dirección postal de PLCopen es:

PO Box 2015
NL 5300 CA Zaltbommel
The Netherlands

Tel.: +31-418-541139
Fax: +31-418-516336

www.plcopen.org

Documento traducido para PLCopen por el miembro educacional:

Área de Ingeniería de Sistemas y Automática
Universidad de Oviedo
Campus de Viesques s/n
33204 - Gijón
España