Introducción

IEC 61131-3 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, algunas de ellas son:

- IEC 61131-3 es el resultado del gran esfuerzo realizado por 7 multinacionales con 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 define las especificaciones de la sintaxis y semántica de los lenguajes de programación de PLCs, incluyendo el modelo de software y la estructura del lenguaje.**

La parte 3 del estándar presenta dos grandes bloques temáticos (ver figura 1):

- Elementos comunes.
- Lenguajes de programación.

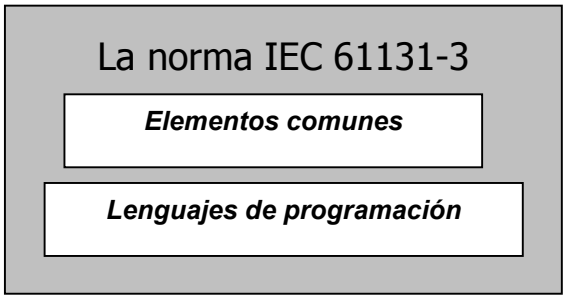

Figura 1. Partes de IEC 61131-3

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 tipo de dato denominado "canal de entrada analógica".

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.

El ámbito de "vida" de una variable está normalmente limitado a la unidad de organización en la cual ha sido declarada (variable local). Esto significa que sus nombres pueden ser reutilizados en otras partes sin conflictos, eliminando una frecuente fuente de errores. Si las variables deben tener un ámbito de "validez" superior, 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

La norma IEC 61131-3 define un modelo software representado en la figura 2.

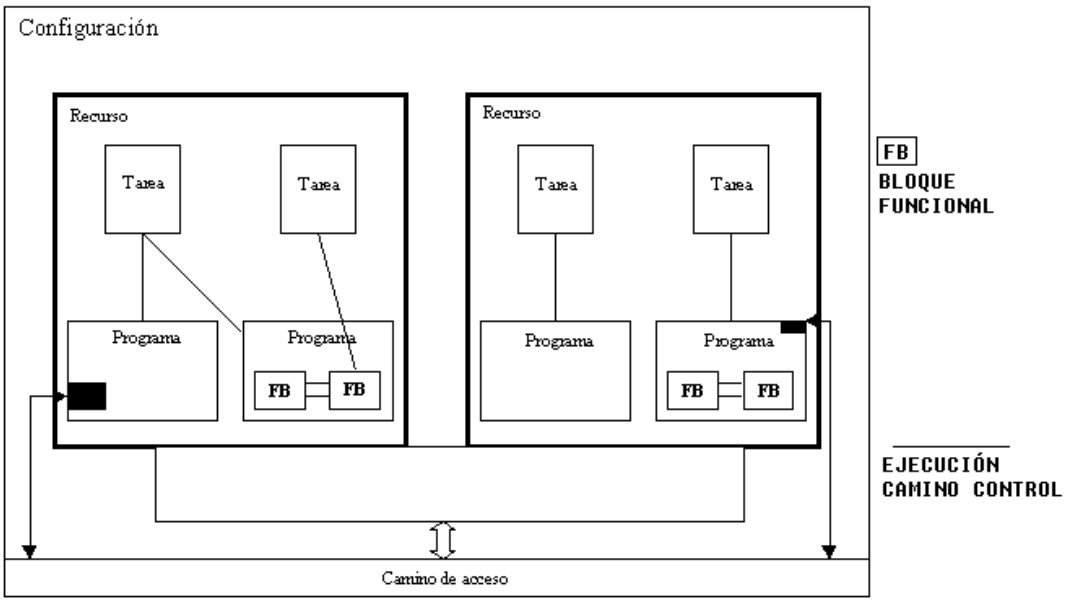

Figura 2. Modelo de software

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 e incluye las características del hardware: procesadores, direccionamiento de la memoria para los canales de I/O y otras capacidades del sistema. El concepto abstracto "configuración" puede ser asimilado a un tipo de PLC dado.

Dentro de una configuración, se pueden definir uno o más *recursos*. Se puede entender el recurso como un procesador capaz de ejecutar programas de control escritos en los lenguajes que define la norma. Dado que una configuración puede tener tantos recursos como se desee, se puede decir que la norma permite definir PLCs con tantas CPUs como se quiera.

En el seno de un recurso pueden ser definidas una o más *tareas*. Las tareas controlan la ejecución de un conjunto de programas y/o bloques funcionales. Cada una de ellos puede ser ejecutada periódicamente o por una señal de disparo especificada, como el cambio de estado de una variable.

Comparado este modelo con un PLC convencional, éste último contiene un solo recurso, ejecutando una única tarea que controla un único programa de manera cíclica. IEC 61131-3 brinda la posibilidad de disponer de estructuras más complejas. Así sería posible soportar multi-procesamiento, gestión de programas por eventos sistemas de control distribuido o en tiempo real con este estándar.

Unidades de Organización de Programa

La norma define tres formas distintas de "presentar" o crear programas de control para PLCs, a saber:

- Programas.
- Funciones.
- Bloques funcionales.

Estas presentaciones reciben el nombre de "POUs (Unidades de Organización de Programa). Los *POUs* serán diseñados a partir de un diferente número de elementos de software, escritos en alguno de los distintos lenguajes definidos en la norma. 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.

Programas

La norma define un *programa* como el "conjunto lógico de todos los elementos y construcciones que son necesarios para el tratamiento de señales que se requiere para el control de una máquina o proceso mediante un PLC".

Es decir, que un programa puede contener la declaración de tipos de datos, variables e instancias de bloques funcionales junto con el conjunto de instrucciones (código o programa propiamente dicho) necesario para llevar a cabo el control deseado del proceso o máquina.

Funciones

IEC 61131-3 especifica funciones estándar y funciones definidas por el usuario. Las funciones estándar son por ejemplo ADD (suma), ABS (valor absoluto), SQRT (raíz cuadrada), SIN (seno), y

COS (coseno). Las funciones definidas por el 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 varias veces con los mismos argumentos (parámetros de entrada) debe suministrar siempre el mismo resultado (salida).

Bloques Funcionales, FB´s

Los bloques funcionales son los equivalentes de los circuitos integrados usados en electrónica, IC´s, que representan funciones de control especializadas. Los FB´s contienen tanto datos como instrucciones, pudiendo guardar los valores de dichas variables entre sucesivas ejecuciones (que es una de las diferencias con las funciones). Se dice por tanto que los FBs tienen "memoria", característica que les confiere un gran potencial de uso.

Presentan una 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 definidos por el usuario empleando alguno de los lenguajes de la norma, pero también existen FB´s estándar (biestables, detección de flancos, contadores, temporizadores, etc.).

Otra de las diferencias fundamentales con respecto a las funciones y que les confiere gran potencia de uso, es la posibilidad de crear tantas copias como se desee de un mismo FB. A cada copia se le llama *instancia*. Cada instancia llevará asociado un identificador y una estructura de datos que contenga sus variables de entrada, de salida e internas separada del resto de instancias.

Gráfico Funcional Secuencial (Secuential Function Chart, SFC)

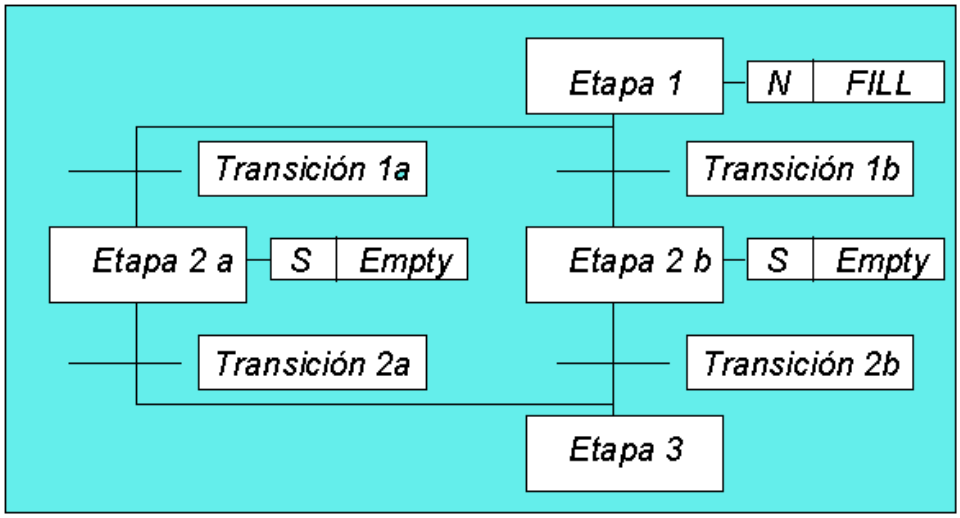

Figura 3. SFC, ejemplo

SFC describe gráficamente el comportamiento secuencial de un programa de control. Esta definición deriva de las Redes de Petri y GRAFCET (IEC 60848), con las modificaciones adecuadas para convertir las representaciones de una norma de documentación (o de un lenguaje de modelado) en un conjunto de elementos de control de ejecución para una POU de un autómata programable (osea, en un lenguaje de programación).

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 de SFC proporcionan un medio para subdividir una POU de un autómata programable en un conjunto de etapas y transiciones interconectadas. Cada etapa lleva asociados un conjunto bloques de acción y cada transición va asociada con una condición 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 de la norma, incluyéndose el propio SFC. Dado que el modo de funcionamiento de SFC requiere que la información almacenada acerca de cuál es la etapa activa se mantenga entre llamadas sucesivas, los únicos POUs que se pueden programar utilizando SFC 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 de manera sencilla e intuitiva.

Lenguajes de Programación

La norma define cuatro lenguajes de programación normalizados. Esto significa que su sintaxis y semántica ha sido definida, no permitiendo particularidades distintivas (dialectos).

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

Literales: * Lista de instrucciones (Instruction List, IL).

* Texto estructurado (Structured Text, ST).

Gráficos: * Diagrama de escalera (Ladder Diagram, LD).

* Diagrama de bloques funcionales (Function Block Diagram, FBD).

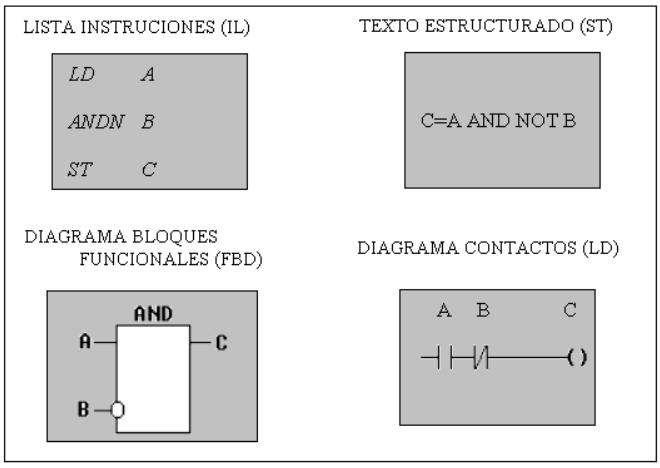

Figura 4. Lenguajes IEC 61131-3

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 escalera (LD) también conocido como "lenguaje de contactos" tiene sus orígenes en los Estados Unidos. Está basado en la representación gráfica de la lógica de relés (automatismos eléctricos). *Lista de Instrucciones* (IL) es el modelo de lenguaje ensamblador basado un acumulador o pila 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 mediante instrucciones para bucles (REPEAT-UNTIL; WHILE-DO), ejecución condicional (IF-THEN-ELSE; CASE), funciones (SQRT, SIN, etc.), etc.

Top-down vs. Bottom-up-

Figura 5. Desarrollo de aplicaciones IEC 61131-3

La norma define dos formas de desarrollar los programas de control (ver figura 5): de arriba a abajo (Top-down) y de abajo a arriba (bottom-up). Se puede especificar inicialmente la aplicación completa y dividirla en partes, declarar las variables y demás. O también se puede comenzar la programación desde abajo, por ejemplo, por medio de funciones y bloque funcionales que resuelvan problemas concretos. Estas funciones y bloques funcionales adecuadamente combinados podrán ser encapsulados en funciones o bloques funcionales que a su vez podrán ser empleados para resolver problemas más complejos, y así hasta resolver el problema en su totalidad.

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 una carga de trabajo mayo 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 (figura 6).

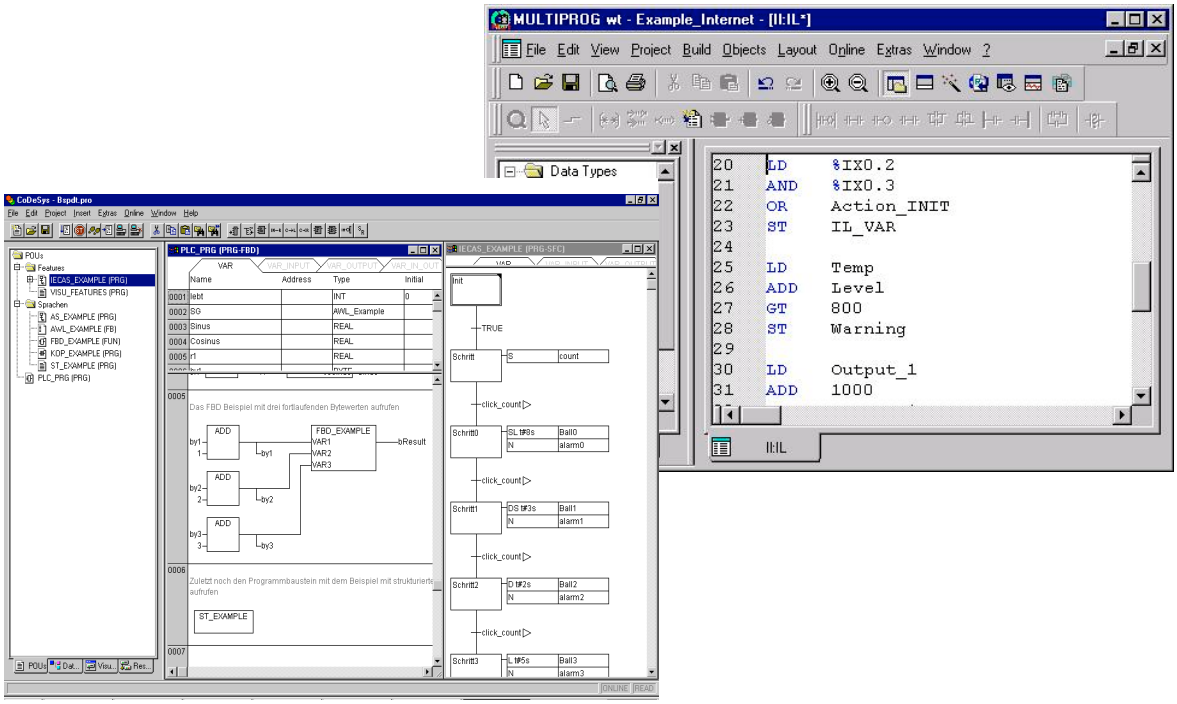

Figura 6. Ejemplos de herramientas de desarrollo IEC 61131-3

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 puede 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. Entre estos beneficios cabe destacar:

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´*.