

## Introducción al estándar IEC 61131-3

*Este documento es una traducción libre, comentada y resumida por el equipo técnico de AISA del material presentado en el website de la Organización **PLCopen** <http://www.plcopen.org/> con objeto de difundir el contenido y alcance de dichos estándares.*

El estándar internacional IEC 61131 es una colección completa de estándares referentes a controladores programables y sus periféricos asociados. Consiste de las siguientes partes:

### **Parte 1: Información General**

Establece las definiciones e identifica las principales características significativas a la selección y aplicación de los controladores programables y sus periféricos asociados.

### **Parte 2: Equipo requerimientos y pruebas**

Especifica los requisitos del equipo y pruebas relacionadas para los controladores programables (PLC) y sus periféricos asociados.

### **Parte 3: Lenguajes de Programación**

Define como un conjunto mínimo, los elementos básicos de programación. Reglas sintácticas y semánticas para los lenguajes de programación usados mas comúnmente, incluyendo los lenguajes gráficos de Diagrama de Escalera y Diagrama de Bloques de Funciones y los lenguajes textuales de Lista de Instrucciones y Texto estructurado. Así como sus principales campos de aplicación, pruebas aplicables y los medios por los cuales los fabricantes pueden expandir o adaptar esos conjuntos básicos a sus propias implementaciones de controlador programable.

### **Parte 4: Guías de Usuario.**

Un reporte técnico que proporciona una vista general y guías de aplicación del estándar para los usuarios finales de los controladores programables.

### **Parte 5: Especificación del servicio de Mensajería.**

Define la comunicación de datos entre controladores programables y otros sistemas electrónicos usando el “Manufacturing Message Specification” (MMS, acorde al ISO/IEC 9506).

### **Parte 7: Programación en lógica difusa.**

Define los elementos básicos de programación de “lógica difusa” para su uso en Controladores programables.

### **Parte 8: Guías para aplicación e implementación de lenguajes de programación.**

Proporciona una guía para los desarrolladores de software para los lenguajes de programación definidos en la parte 3.

## **El IEC 61131-3 un recuso de programación estándar.**

IEC 61131-3 es el primer esfuerzo real para estandarizar los lenguajes de programación usados en para la automatización industrial.. Con su soporte mundial, es independiente de una sola compañía.

Esta parte de programación, es la tercera del estándar 61131.

Hay muchas formas de ver esta parte del estándar. Algunas son:

- Es el resultado de la fuerza de trabajo<sup>3</sup> del IEC TC65 SC65B.
- Es resultado del trabajo de 7 compañías internacionales sumando muchos años de experiencia en el campo de la automatización industrial.
- Son aproximadamente 200 páginas de texto, con unas 60 tablas.
- Es la especificación de la sintaxis y semántica de un conjunto unificado de lenguajes de programación incluyendo el modelo general del software y su estructura como lenguaje.

Una forma conveniente de verlo, es dividiendo el estándar en 2 partes:

1. Elementos Comunes.
2. Lenguajes de Programación.

Veamos con mayor detalle estas 2 partes.

### **Elementos Comunes.**

#### **Tipos de Datos.**

Dentro de los elementos comunes se definen los tipos de datos. La tipificación de los datos previene errores en una etapa temprana. Se usa para definir el tipo de cualquier parámetro usado. Esto evita que por ejemplo se divida una fecha entre un entero.

Los tipos de datos comunes son: Boolean, Integer, Real, Byte y Word. También Date, Time\_of\_Day y String. Basado en ellos, uno puede definir sus propios tipos de datos, llamados “tipos de datos derivados”.

#### **Variables.**

Las variables son únicamente asignadas a direcciones de hardware explícitas (por ejemplo entradas y salidas) en la configuración, recursos o programas. De esta manera se le da a los programas una independencia de alto nivel del hardware, soportando el re-uso del software.

El enfoque (visibilidad) de las variables es normalmente limitado a la unidad de organización en la cual son declaradas (por ejemplo: local). Esto significa que sus nombres pueden ser usados nuevamente sin ningún conflicto en otras partes, eliminando otra fuente de errores. Si las variables requieren un alcance global, deben ser declaradas como tales. Los parámetros pueden recibir un valor inicial al arranque y al reinicio “en frío”, con objeto de asegurar su valor correcto al inicio de la ejecución de los programas.

## **Configuración, recursos y tareas.**

Para entender mejor esto, es conveniente ver el modelo de software, tal como se define en el estándar:

Al nivel mas alto, el software completo que se requiere para solucionar un problema de control particular puede ser formulado como una configuración. Una configuración es específica a un sistema de control particular, incluyendo el arreglo del hardware, recursos de procesamiento, direcciones de memoria para los canales de entrada/salida y otras capacidades del sistema.

Dentro de una configuración, se pueden definir una o más tareas. Estas tareas controlan la ejecución de un conjunto de programas y/o bloques de función. Las tareas pueden ser ejecutadas periódicamente o a la ocurrencia de algún evento disparador, por ejemplo el cambio en una variable.

Los programas están constituidos por diferentes elementos de software escritos en cualquiera de los lenguajes definidos por IEC. Típicamente un programa consiste de una red (network) o funciones y bloques de función que son capaces de intercambiar datos. Las funciones y los bloques de función son los bloques de construcción básicos y contienen una estructura de datos y un algoritmo.

Comparemos lo anterior con un PLC convencional: Este contiene recursos corriendo una tarea, corriendo un programa. IEC 61131-3 le agrega a esto mucho mas, haciéndolo abierto a mayores capacidades tales como multiprocesamiento y conducción por sucesos.

## **Unidades de organización del programa.**

En IEC 61131-3 los Programas, Bloques de Función y Funciones son llamados Unidades de Organización de Programa (program organization units o POUs).

### Funciones

IEC define Funciones Estándar y Funciones Definidas por el Usuario. Las funciones estándar son por ejemplo: ADD (suma), ABS (absoluto), SQRT (cuadrado) SIN (seno), etc. Las funciones definidas por el usuario (basadas en las funciones estándar), una vez definidas pueden ser re-usadas una y otra vez.

### Bloques de Función (Function Blocks FBs)

Los Bloques de Función son los equivalentes a los circuitos integrados y representan una función de control especializada. Contienen datos así como el algoritmo así que pueden conservar información de su estado. Esto las diferencia de las funciones.

Nota: Una función siempre da la misma salida para las mismas entradas. Un bloque de Función no, ya que puede tomar en cuenta como llego a su estado actual.

Los Bloques de Función cuentan con una Interface bien definida y su parte interna oculta. Esto es actúan como una caja negra. Esto permite una clara separación entre diferentes niveles de programadores o personal de mantenimiento.

Un lazo de control de temperatura, o un PID es un excelente ejemplo de un Bloque de Función. Una vez definido puede ser usado una y otra vez en el mismo programa, diferentes programas o diferentes proyectos, es decir son re-usables.

Los Bloques de Función pueden ser escritos en cualquiera de los lenguajes definidos por el estándar IEC y en muchos casos incluso en "C". También pueden definirse por el usuario, basados en los existentes, obteniéndose así los Bloques de Función derivados.

## **Programas**

Con los bloques constructivos mencionados anteriormente se puede decir que un Programa es una Red de Funciones y Bloques de Función. Un programa puede ser escrito en cualquiera de los lenguajes de programación definidos en el estándar.

## **Grafica de Secuencia de Funciones (Sequential Function Chart - SFC)**

SFC describe gráficamente el comportamiento secuencial de un programa de control. Se derivan de sus antecesores "Petri Nets" y del IEC848 Grafcet.

El SFC estructura la organización interna de un programa y ayuda a descomponerlo en partes más fácilmente manejables, mientras mantiene la visión general.

El SFC consiste de "Pasos" enlazados con "Bloques de Acción" y "Transiciones". Cada Paso representa un estado particular del sistema que se esta controlando. Una transición se asocia con una condición (condiciones) que cuando es cierta causa que el paso previo se desactive y el paso próximo se active. Los pasos están ligados a bloques de Acción, que ejecutan algunas acciones de control pertinentes a dicho Paso.

Cada elemento del diagrama puede ser programado en cualquiera de los lenguajes definidos por el estándar IEC, incluido el propio SFC.

Se pueden usar secuencias alternativas o incluso paralelas según se requiere con frecuencia en los procesos batch.

Por su estructura general, SFC proporciona un medio de comunicación o entendimiento entre personas con diferentes especialidades.

## Lenguajes de Programación

Dentro del estándar se definen 4 lenguajes de programación. Esto es, se definen su sintaxis y semántica. Una vez que usted los aprendió, puede aplicarlos a una gran cantidad de sistemas que están basados en estos estándares.

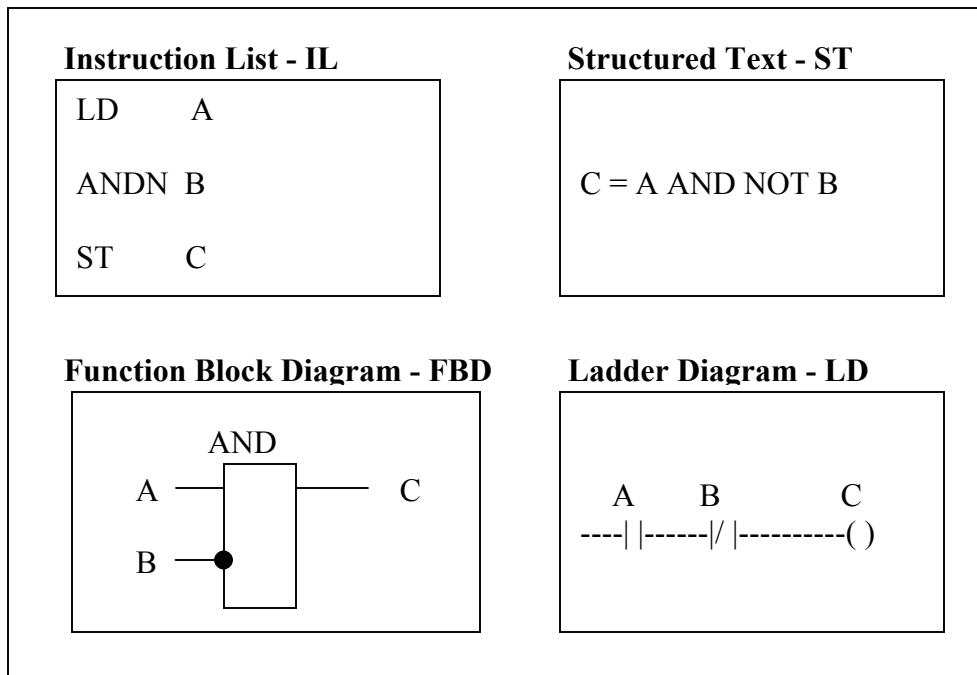
Los lenguajes son 2 de tipo textual y 2 de tipo gráfico:

Textuales:

- Lista de Instrucciones (Instruction List – IL)
- Texto estructurado (Structured Text – ST)

Gráficos

- Diagrama de Escalera (Ladder Diagram – LD)
- Diagrama de Bloques de Funciones (Function Block Diagram – FBD)



En la figura, los cuatro lenguajes describen la misma parte simple de un programa.

La elección del lenguaje de programación depende de:

- La formación y experiencia del programador.
- El problema que se atiende (la aplicación particular).
- El nivel de descripción del problema.
- La estructura del Sistema de Control.
- La Interface con otras personas o departamentos.

Todos los 4 lenguajes están interrelacionados. Forman un conjunto común relacionado con la experiencia existente. De esta manera, también proporcionan una forma de comunicación a personas con diferente especialidad.

El Diagrama de escalera (LD) tiene sus raíces en USA. Esta basado en la representación grafica de la lógica de relevadores.

La Lista de Instrucciones (IL) Es tiene su origen en Europa y se asemeja a los programas en ensamblador.

El Diagrama de Bloques de Función (FBD) es común en la industria de proceso y representa el comportamiento del programa mediante un conjunto de bloques de funciones a la manera de los diagramas de circuitos de electrónica. Esto es: miran al sistema en términos de flujo de señales entre elementos de procesamiento.

El Texto estructurado (ST), es un lenguaje poderoso de alto nivel, con sus raíces en Ada, Pascal y C. Contiene todos los elementos esenciales de un lenguaje de programación moderno, incluyendo selección del flujo de ejecución (IF-THEN-ELSE y CASE OF) y lazos de iteración (FOR, WHILE y REPEAT), que pueden ser anidados. Este lenguaje resulta excelente para la definición de bloques de función complejos que pueden ser usados en cualquiera de los otros lenguajes.

## **Desarrollo Hacia Abajo o Hacia Arriba.**

El estándar permite dos maneras de desarrollar su programa, ya sea partiendo de una visión general para luego resolver los detalles (hacia abajo) o iniciando por la parte de detalles específicos (iniciando desde abajo hacia arriba) por ejemplo mediante bloques de función y bloques de función derivados.

El ambiente de desarrollo le ayudara durante el proceso completo.

## **Implementaciones**

Los requerimientos completos del IEC 61131-3 no son fácilmente llenados. Por esa razón el estándar permite implementaciones parciales en varios aspectos. Tales como el numero de lenguajes soportados, funciones y bloques de función. Esto proporciona libertad del lado del suministrador, pero el usuario (cliente) debe estar prevenido al respecto durante su proceso de selección.

Muchos ambientes de desarrollo de programación IEC ofrecen lo que usted puede esperar de un ambiente moderno: Operación mediante "Mouse", menús descolgables, pantallas graficas, soporte de múltiples ventanas, funciones de hipertexto, verificación durante el diseño. Tenga previsto que esto no es

especificado dentro del estándar en si mismo y es una de las cosas en que diferentes suministradores pueden diferenciarse.

## **Conclusión**

Las implicaciones técnicas del estándar IEC 61131-3 son altas, pero dejan espacio suficiente para el crecimiento y la diferenciación entre proveedores, haciéndolo aplicable tanto al corto como al largo plazo.

El IEC 61131-3 tendrá un gran impacto en toda la industria de control. Ciertamente no se restringe al mercado tradicional de los PLC's. También se ha adoptado en el mercado de control de movimiento, Sistemas distribuidos y Softlogic basados en PC's. Incluyendo paquetes SCADA y continua en aumento.

Algunos de los beneficios de adoptar este estándar son:

- Disminuir el desperdicio de recursos humanos en entrenamiento, solución de errores y mantenimiento.
- Favorecer el enfoque a la solución del problema mediante un alto nivel de rehuso del software.
- Reduciendo los malentendidos y errores.
- Técnicas de programación de enfoque amplio para la generalidad del control industrial.
- Permitiendo la combinación de diferentes componentes de diferentes programas, proyectos, localidades compañías y países.