

**ESTUDIO SOBRE EL ESTANDAR IEC 1131, APORTES Y
APLICABILIDAD DE ÉSTE EN LA AUTOMATIZACIÓN INDUSTRIAL**

**OTTO EDUARDO PATIÑO LUKERNA
HUMBERTO VARGAS FORERO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECANICAS
ESCUELA DE INGENIERIA MECANICA
ESPECIALIZACION EN INGENIERIA MECATRONICA
BUCARAMANGA
2008**

**ESTUDIO SOBRE EL ESTANDAR IEC 1131, APORTES Y
APLICABILIDAD DE ÉSTE EN LA AUTOMATIZACIÓN INDUSTRIAL.**

**OTTO EDUARDO PATIÑO LUKERNA
HUMBERTO VARGAS FORERO**

**Trabajo de investigación presentado como requisito para optar al título
de Especialista en Ingeniería Mecatrónica**

**Director
JORGE ENRIQUE MENESES**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECAICAS
ESCUELA DE INGENIERIA MECANICA
ESPECIALIZACION EN INGENIERIA MECATRONICA
BUCARAMANGA
2008**

DEDICATORIA

A Dios por ser la fuente inagotable de energía que me fortalece a cada
instante,

A mí adorado hijo, Carlos Eduardo, mi mayor inspiración,

A mis padres por su colaboración y apoyo.

Otto Eduardo

A Dios por fortalecerme y guiarme en la construcción de mi proyecto de vida.

A mi amada esposa, Lizeth, por su comprensión y dedicación.

Humberto

AGRADECIMIENTOS

A la Universidad Industrial de Santander, por participar activamente en nuestra formación y actualización profesional.

A todas aquellas personas que de una u otra manera colaboraron en la consecución de este logro.

Los autores

CONTENIDO

	Pág.
INTRODUCCIÓN	1
1. PERESNTACIÓN DEL PROYECTO	2
1.1 OBJETIVOS	2
1.1.1 Objetivo General	2
1.1.2 Objetivos Específicos	2
1.1.3 Planteamiento del problema	2
2. ESTÁNDAR IEC 1131	4
2.1 INTRODUCCIÓN	4
2.2 PROPUESTA DEL ESTÁNDAR	4
2.2.1 Ventajas del estándar	4
2.2.2 Claves de desarrollo del estándar.	5
2.3 COMPOSICIÓN DEL ESTÁNDAR.	5
2.3.1 Parte 1. Información general	5
2.3.2 Parte 2. Especificaciones y ensayos de equipos	8
2.3.3 Parte 3. Lenguajes de programación	10
2.3.4 Parte 4. Guía de usuario	29
2.3.5 Parte 5. Especificación de comunicaciones	29
3. APORTES DEL IEC 1131	32
3.1 INTRODUCCIÓN	32
3.2 CONCEPTOS E INTERPRETACIONES	32
3.3 MANEJO DEL DISEÑO	32
3.3.1 Esquema Secuencial De Funciones (SFC)	33
3.4 COMUNICACIÓN ENTRE DISPOSITIVOS	37
4. APLICABILIDAD DE LA NORMA	47
CONCLUSIONES	50
TRABAJO FUTURO	51
BIBLIOGRAFÍA	52

LISTA DE TABLAS

	Pág.
Tabla 1. Operadores IL	18
Tabla 2. Estructuras ST	20
Tabla 3. Operadores ST	21
Tabla 4. Elementos FBD	22
Tabla 5. Elementos LD	24
Tabla 6. Presentación de estado	31
Tabla 7. Estructura comunicación BF	31

LISTA DE FIGURAS

	Pág.
Figura 1. Estructura funcional de un sistema de autómata programable	6
Figura 2. Función de los sensores y actuadores	6
Figura 3. Función de interfaz hombre-maquina MMI	7
Figura 4. Programación, puesta a punto, ensayo y documentación.	7
Figura 5. Interfaz y alimentación de corriente	8
Figura 6. Modelo de software	11
Figura 7. Modelo de comunicación uno	11
Figura 8. Modelo de comunicación dos	12
Figura 9. Modelo de comunicación tres	12
Figura 10. Unidad de organización de programa	13
Figura 11. Flujo de datos dentro de un programa	14
Figura 12. Comunicación entre programas	15
Figura 13. Ejemplo de elementos de configuración	15
Figura 14. Declaración de Bloque funcionales y parámetros	16
Figura 15. Configuración, recursos y variables	17
Figura 16. Representación de tareas	17
Figura 17. Ejecución LD	26
Figura 18. Etapa	27
Figura 19. Transición	27
Figura 20. Acción	27
Figura 21. Divergencia en Y	27
Figura 22. Convergencia en Y	28
Figura 23. Divergencia en O	28
Figura 24. Convergencia en O	29
Figura 25. Modelo de redes	30
Figura 26. Niveles de diseño SFC	34
Figura 27. Macroetapa	35
Figura 28. Función REMOTE_VAR	38
Figura 29. Estados de Bloque Funcional (ST AUS, USTATUS)	39
Figura 30. Adquisición de datos (READ)	40
Figura 31. Programado de datos adquiridos (USEND, URCV)	41
Figura 32. Parámetros de control (WRITE)	42
Figura 33. Control de Interconexión (SEND, RCV)	42
Figura 34. Programación Reporte de Alarmas (NOTIFY, ALARM)	44
Figura 35. Conexión maestra (CONNECT)	46
Figura 36. Grafico de aplicabilidad	49

RESUMEN

TITULO: **ESTUDIO SOBRE EL ESTANDAR IEC 1131, APORTES Y APLICABILIDAD DE ESTE EN LA AUTOMATIZACIÓN INDUSTRIAL***

AUTORES: **PATIÑO LUKERNA, Otto Eduardo**
VARGAS FORERO, Humberto**

PALABRAS CLAVES: **Estándar, aportes, aplicabilidad, automatización, industrial.**

DESCRIPCIÓN

Debido a las diferencias marcadas por los fabricantes de PLC'S relacionadas con los elementos de control y la utilización de sus propios lenguajes de programación, se ha vuelto un gran inconveniente para los diseñadores, crear autómatas programables sin incurrir en grandes costos y sin tener la posibilidad de reutilizar los software y elementos de control en otras aplicaciones. Por tal razón, fue necesario recurrir a una Norma que estandarice los procedimientos en el desarrollo de proyectos de automatización industrial.

Esta norma fue creada por la Comisión Internacional Electrotécnica IEC como IEC 1131, que básicamente contiene en sus dos primeras partes las definiciones técnicas utilizadas en proyectos de la automatización, y las características Técnicas que deben cumplir los PLC'S y demás elementos de control para que faciliten la intercambiabilidad con otras aplicaciones y el tipo de ensayos que el fabricante debe suministrar para la aceptabilidad del componente.

Como tercera parte de la norma, se definen cuatro lenguajes de programación, los más utilizados en proyectos de automatización, con el fin de permitir al diseñador, utilizar en PLC'S de cualquier fabricante el lenguaje de programación que más se adapte a la aplicación.

Adicionalmente, la Norma determina que se debe suministrar una guía de operación y mantenimiento para el usuario final y estandarizar los sistemas de comunicación entre PLC'S y otros dispositivos mejorando la compatibilidad.

Como interpretación de la Norma IEC 1131 queda claro que se debe considerar el esquema secuencial de funciones SFC como un lenguaje de Diseño y no como lenguaje de programación, ya que posee un lenguaje único y bien estructurado que permite unificar la descripción del proceso para técnicos de los diferentes campos que interactúan en el proyecto.

*Proyecto de grado

**Universidad Industrial de Santander, Especialización en Mecatrónica. Director Jorge Enrique Meneses

SUMMARY

TITLE: STUDY ON THE STANDARD IEC 1131, CONTRIBUTIONS AND APPLICABILITY OF THIS IN THE INDUSTRIAL AUTOMATION*

**AUTHORS: PATIÑO LUKERNA, Otto Eduardo
VARGAS FORERO, Humberto ****

KEY WORDS: Standard, contributions, applicability, automation, industrial.

DESCRIPTION

Due to the differences marked by the makers of PLC'S related with the control elements and the use of their own programming languages, it has become a great inconvenience for the designers, to create programmable robots without incurring in big costs and without having the reuse possibility the software and control elements in other applications. For such a reason, it went necessary to appeal a Norma that standardizes the procedures in the development of projects of industrial automation.

This norm was created by the Electrotechnical International Commission IEC like IEC 1131 that basically it contains in its first two parts the technical definitions used in projects of the automation, and the Technical characteristics that should to complete the PLC'S and other control elements so that they facilitate the intercambiabilidad with other applications and the type of rehearsals that the maker should it gives for the acceptability of the component.

As third part of the Norm, they are defined four programming languages, those more used in automation projects, with the purpose of allowing to the designer, to use in any maker's PLC'S the programming language that more it adapts to the application.

Additionally, the Norm determines that it should be given an operation guide and maintenance for the final user and it standardizes the communication systems between PLC'S and other devices improving the compatibility.

As interpretation of the Norma IEC 1131 are clear that it should be considered the sequential outline of functions SFC like a language of Design and not as language of programming, since it possesses an unique and well structured language that it allows to unify the description of the process for technicians of the different fields that participate in the project.

* Grade project

** Industrial University of Santander, Specialization in Mecatrónica. Managing Jorge Enrique Meneses.

INTRODUCCIÓN

Es el campo de la automatización industrial, un escenario de crecimiento acelerado, y en el cual se han realizado avances importantes, en la consecución de un mejor rendimiento del conocimiento y de la tecnología, evidenciado esto; en los elementos de control cada vez mas robustos y con mayor funcionabilidad, de igual forma; la construcción de metodologías de trabajo que aportan a la solución de problemáticas, como la dificultad de comunicación, mantenimiento y reutilización de los proyectos de automatización, que por la dependencia a los fabricantes se presentan.

En este escenario, surge un estándar emitido por la Comisión Internacional Electrotécnica (IEC) denominado IEC 1131, donde se muestra una metodología de trabajo para automatización industrial, que: Primero, reconoce el avance de las casas fabricantes, en relación con los lenguajes utilizados por los PLC's, y articula estos desarrollos, incluyendo los lenguajes mas utilizados en el mundo y dictando para ellos, la sintaxis, los elementos y su forma de trabajo. Segundo, posibilita, al independizar los desarrollos de los proyectos de automatización, la implementación en cualquier dispositivo PLC o en combinación de PLC's de diferentes fabricantes. Con esto; surge solución a una problemática, que permite seguir en expansión el campo de la automatización.

1. PRESENTACIÓN DEL PROYECTO

El presente proyecto, esta enmarcado en un contexto de investigación, que en el ámbito de la automatización industrial, propone; indagar sobre los aportes que en al área de diseño y comunicación realiza el IEC 1131, al igual que rastrear la aplicabilidad de la norma y sus factores, es decir; la presente investigación expone un trabajo que, basándose en el estándar IEC 1131, ofrece un análisis sobre los aportes de éste a la labor de diseño y comunicación, en automatización industrial, y en sentido conductista, la aplicabilidad de esta norma en el campo.

Para el desarrollo del presente trabajo se contemplan cuatro capítulos donde se expone en primer lugar los objetivos y el problema planteado, seguido de una exposición del estándar IEC 1131, para continuar con la exposición del análisis acerca de los aportes en la labor el diseño y comunicación por parte del IEC 1131, finalmente se tendrá el capítulo de la aplicabilidad de la norma en el campo de la automatización.

1.1 OBJETIVOS

1.1.1 Objetivo General

Indagar sobre los aportes y la aplicabilidad del estándar IEC 1131, dentro del campo de la automatización industrial.

1.1.2 Objetivos Específicos

Presentar como resultado de esta investigación, desde el estudio del estándar y su entorno, análisis sobre los siguientes aspectos:

- Aportes para el diseño de proyectos de automatización.
- Contribución en comunicación de dispositivos.
- Estado de aceptación y aplicación del IEC 1131.

1.1.3 Planteamiento del problema. Entre los múltiples esfuerzos encontrados en el desarrollo dentro del campo de la automatización, todos en algún sentido, unos con más agudeza que otros, exponen o formulan formas de trabajo para el diseño e implementación de proyectos de

automatización, es decir; todos atienden la necesidad y problemática de la labor de diseño, esto como consecuencia de la dificultad existente para un mejor aprovechamiento y rendimiento de los proyectos, pues la no existencia de un puente comunicacional en esta fase ha disgregado los adelantos en la automatización, es decir; el fortalecimiento del diseño ha sido y sigue siendo un problema ávido de solución y esto en el sentido de la generación de una comunicación conceptual y de dispositivos.

2. ESTÁNDAR IEC 1131

2.1 INTRODUCCIÓN

El estándar IEC 1131, es un documento que expone una forma de trabajar en automatización industrial, recogiendo para esto, cinco lenguajes de programación, escogidos desde el conocimiento de la Comisión Internacional Electrotécnica (IEC) en el campo de la automatización y la utilización de estos en la mayoría de los PLC's; así pues, tenemos que el estándar es un documento que conteniendo los lenguajes, enuncia para cada uno su sintaxis, sus fortalezas y su mejor aprovechamiento.

Aunado a lo anteriormente expuesto, se encuentran otras partes en el estándar, que recoge desde las definiciones hasta cómo desarrollar la documentación de un proyecto de automatización, en este capítulo se expondrá de forma sintetizada el contenido del estándar IEC 1131.

2.2 PROPUESTA DEL ESTÁNDAR

Como se exponía en el primer capítulo de este trabajo, la automatización industrial ha sido desarrollada con dependencia a los lenguajes definidos por los fabricantes de los PLC's, dejando esto como consecuencia: Primero, una atadura al PLC en el que se realizó el desarrollo. Segundo: la reutilización de ese proyecto no será posible en un dispositivo diferente al programado y finalmente la interpretación estará sujeta al lenguaje definido por la casa fabricante, es entonces; el estándar la respuesta a la necesidad de un mecanismo de comunicación efectivo en la automatización industrial, tratando de que la existencia de múltiples PLC's no siga siendo un obstáculo para el mantenimiento, interpretación y reutilización de los proyectos, debido a la multiplicidad de lenguajes, por el contrario se exhorta desde la IEC a que todos los fabricantes de PLC's acojan el estándar IEC 1131 y de esta forma saldar la dificultad.

2.2.1 Ventajas del estándar

- Reduce el esfuerzo humano en entrenamiento, depuración, mantenimiento y consultaría.
 - Una vez aprendido se puede utilizar en todos los sistemas
- Permite crear software reutilizable, minimiza:

- El tiempo de desarrollo
- El esfuerzo de codificación
- Los errores de compilación y ejecución
- Coordina eficazmente diferentes componentes desde distintas localizaciones, compañías o proyectos.
 - Amplio campo de aplicación.
- Aumenta la conectividad, facilita la distribución del control.

2.2.2 Claves de desarrollo del estándar.

- Software estructurado a través de diseño, proyectos, tareas, programas y bloques.
- Descripción del comportamiento secuencial complejo de un proceso a través del SFC.
- Encapsulación del software a través de unidades de organización de programas, estructuras y tipos de datos.

2.3 COMPOSICIÓN DEL ESTÁNDAR.

El estándar esta compuesto por cinco partes que expresan su contenido e intención para lo cual fue creado y están expuestas de forma sintetizada así:

2.3.1 Parte 1. Información general. Contiene las definiciones generales usadas en la norma, de igual forma se identifican las características típicas de los sistemas de autómatas programables.

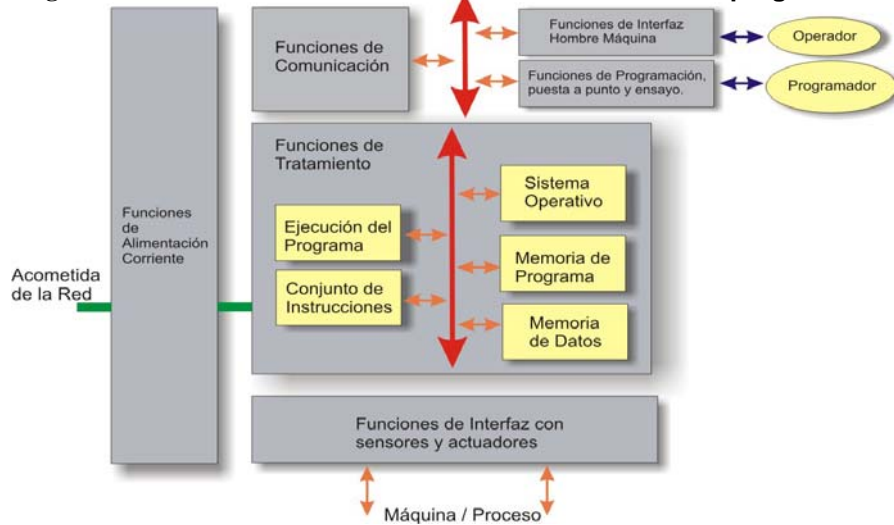
Algunas definiciones contenidas son:

Programación de aplicación, lista de asignaciones, sistema automatizado, disponibilidad, datos boléanos, bus, componente, conexión/desconexión, lista de referencia cruzada, equipo de comunicación de datos (DCE), equipo terminal de datos (DTE), ejecución, parada de seguridad, imagen de entrada/salida, entrada, instrucción, fallo interno, diagrama de escalera , lenguaje (FBD, IL, LD, ST, SFC), sistema de control lógico, procesador principal (MPU), interfaces hombre-máquina (MMI), tiempo medio entre fallos (MTBF), red, en línea, salida, programa, periférico, barras de corriente, autómata programable (AP ó PLC), equipo de programación y puesta a punto (PADT), estación de entrada/salida remota (RIOS), reinicio, sistema de control secuencial, transmisión de datos en serie, etc.

Dentro de la estructura funcional de un sistema de autómata programable tenemos:

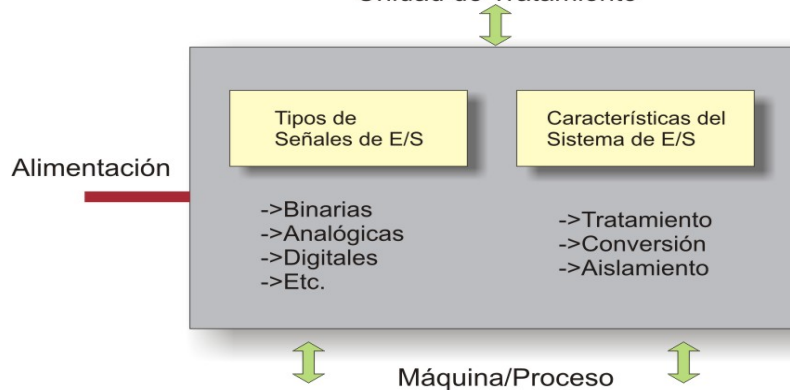
- Función de tratamiento de la señal.
- Función de interfaz con los sensores y actuadores.
- Función de comunicación.
- Función de interfaz hombre-máquina.
- Funciones de programación, puesta a punto, ensayo y documentación.
- Funciones de alimentación de corriente

Figura 1. Estructura funcional de un sistema de autómatas programables



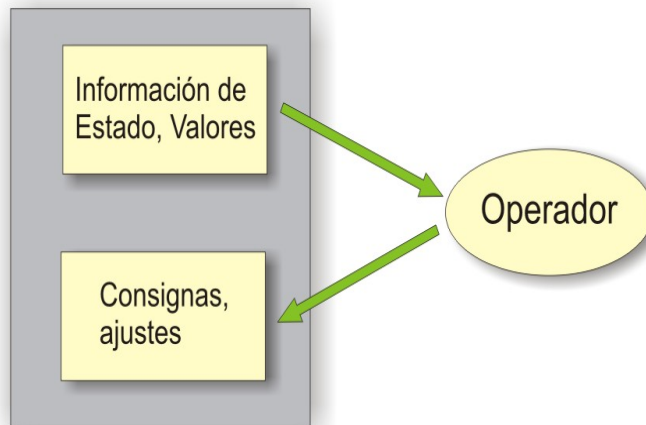
Fuente: IEC 61131-1, Estructura funcional de un sistema de autómatas programables

Figura 2. Función de los sensores y actuadores
Unidad de Tratamiento



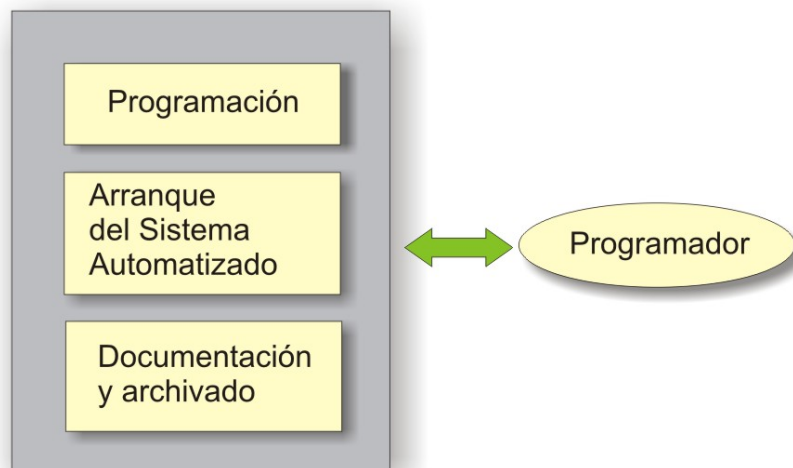
Fuente: IEC 61131-1, Función de los sensores y actuadores

Figura 3. Función de interfaz hombre-maquina MMI



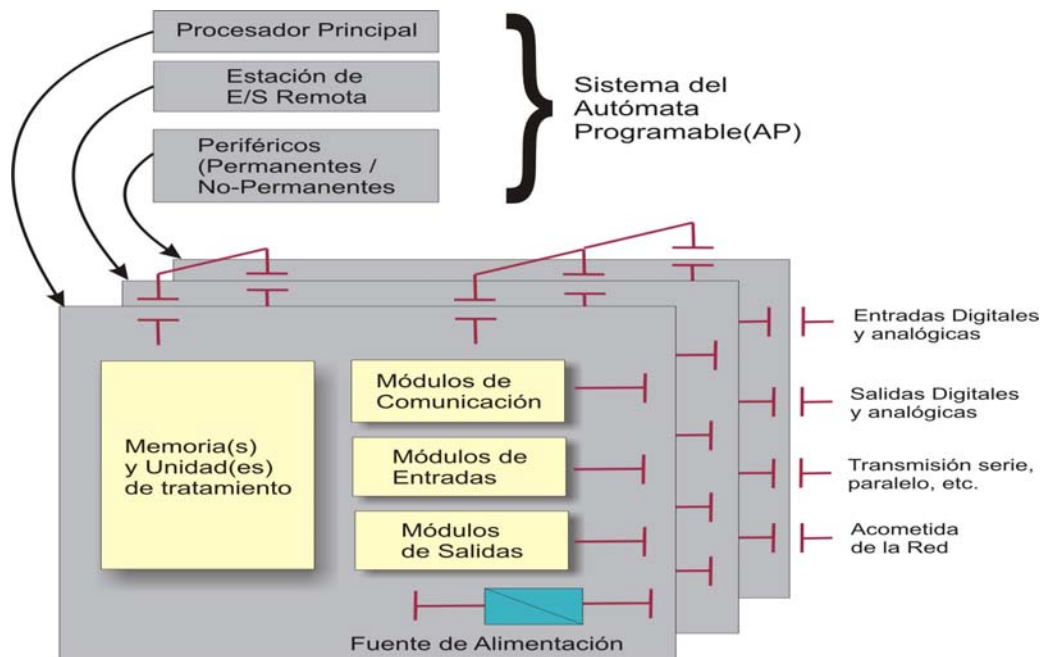
Fuente: IEC 61131-1, Función de interfaz hombre-maquina MMI

Figura 4. Programación, puesta a punto, ensayo y documentación.



Fuente: IEC 61131-1, Programación, puesta a punto, ensayo y documentación

Figura 5. Interfaz y alimentación de corriente



Fuente: IEC 61131-1, Interfaz y alimentación de corriente

2.3.2 Parte 2. Especificaciones y ensayos de equipos. Contiene los requisitos mecánicos, eléctricos y funcionales para los autómatas programables y los periféricos correspondientes, así como las condiciones de servicio, almacenamiento y transporte aplicables. Contiene además la información que ha de suministrarse por el fabricante y los métodos y procedimientos de ensayo que han de utilizarse para la comprobación del cumplimiento de los requisitos por parte de los autómatas programables y sus periféricos.

Se tiene entonces que las condiciones de servicio y requisitos del entorno físico son:

- Condiciones del entorno físico: temperatura, humedad, contaminación, inmunidad a la corrosión y altitud.
- Condiciones de servicio y requisitos eléctricos: alimentaciones, ruido eléctrico, sobretensiones, etc.
- Condiciones de servicio y requisitos mecánicos: Vibraciones, choque y caída libre.

- Condiciones de servicio especiales. Polvo, humo, partículas radiactivas, vapores, sales, insectos, pequeños animales, etc.
- Requisitos para el transporte y almacenaje: temperatura, presión atmosférica y humedad relativa.

Los Requisitos eléctricos son:

- Alimentación de corriente alterna (c.a.) y continua (c.c.).
- E/S digitales.
- E/S analógicas.
- Interfaces de comunicación.
- Procesador(es) principal(es) y memoria(s) del sistema AP.
- Estaciones de entrada/salida remota (RIOS).
- Periféricos: PADT, TE, MMI.
- Inmunidad al ruido y ruido emitido.
- Propiedades dieléctricas.
- Autodiagnósticos y diagnósticos.

Requisitos mecánicos

- Protecciones contra el riesgo de choques eléctricos.
- Requisitos de distancias en el aire y líneas de fuga.
- Requisitos de inflamabilidad para materiales aislantes.
- Envoltente.
- Requisitos mecánicos de los materiales de conexión.
- Disposiciones para la tierra de protección.
- Tierra funcional.
- Cables y conectores de interconexión.
- Conexión/desconexión de unidades desmontables.
- Requisitos de la batería.
- Marcado e identificación.

La Información que debe facilitar el fabricante es:

El fabricante deberá facilitar a los usuarios la información necesaria para la aplicación, proyecto, instalación, puesta en marcha, funcionamiento y mantenimiento del sistema de autómatas programables. Adicionalmente el fabricante puede ocuparse de la formación del usuario.

- Tabla resumen con la información que se debe facilitar.
- Tipo y contenido de la información escrita: Catálogos y hojas de características, manuales de usuario y documentación técnica.
- Información relativa al cumplimiento de esta norma.
- Información relativa a la fiabilidad.

- Información relativa a la seguridad.

Los ensayos y verificaciones son:

- Se define como ha de verificarse la conformidad del autómata programable y sus periféricos correspondientes con los requisitos fijados en las partes 1 y 2 de la norma.
- Estos ensayos no se refieren a los métodos de aplicación de los autómatas programables, para cumplir con los requisitos del sistema automatizado.
- Se dividen en ensayos de tipo y ensayos de rutina.

Los ensayos de tipo son:

- Equipos a ensayar.
- Procedimientos de verificación.
- Condiciones generales para los ensayos.
- Ensayos climáticos, mecánicos y eléctricos.
- Verificación de las características de la alimentación de c.a. y c.c.
- Verificación de las características de entrada/salida.
- Verificación de las características del procesador principal.
- Verificación de las estaciones de E/S remotas.
- Verificación de las características de los periféricos.
- Verificación del autodiagnóstico y diagnóstico.

Los ensayos de rutina son:

- Ensayo estándar de rigidez dieléctrica
- Ensayo de continuidad de la tierra de protección.

2.3.3 Parte 3. Lenguajes de programación. Contiene la definición de los lenguajes de programación de uso más común, las reglas sintácticas y semánticas. Y se divide para su comprensión en elementos comunes y los lenguajes, interpretación esta de lectura denominada Top Down.

A. Elementos Comunes.

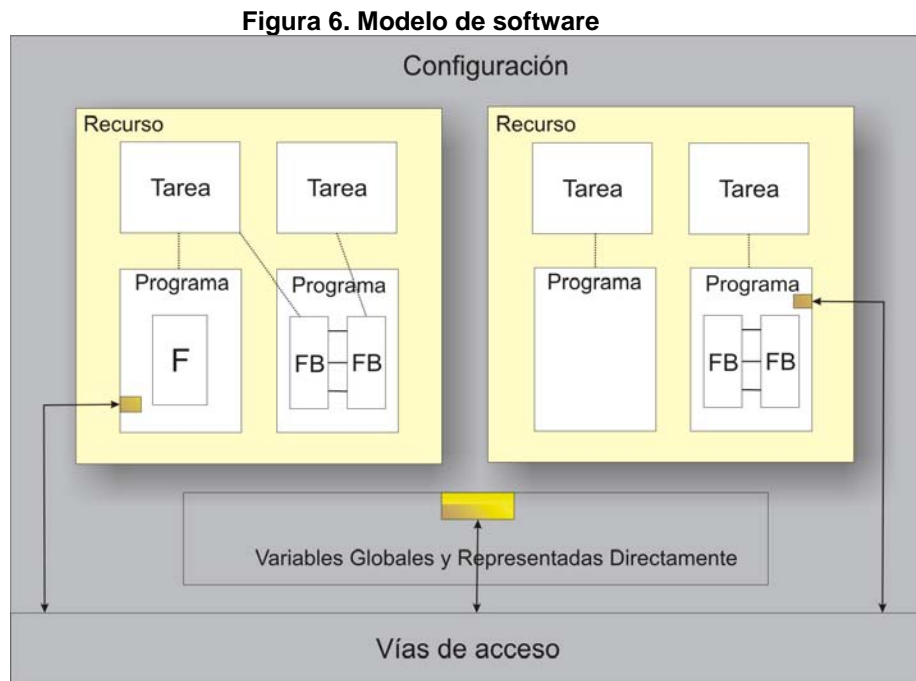
- Tipos de datos y variables.

Tipos de datos: boléanos, enteros, reales, byte, palabra, cadenas de caracteres, fecha, hora del día, canal analógico de entrada, tipos de datos derivados (creados por el usuario).

BOOL, INT, REAL, BYTE, WORD, STRING, DATE, TIME_OF_DAY.

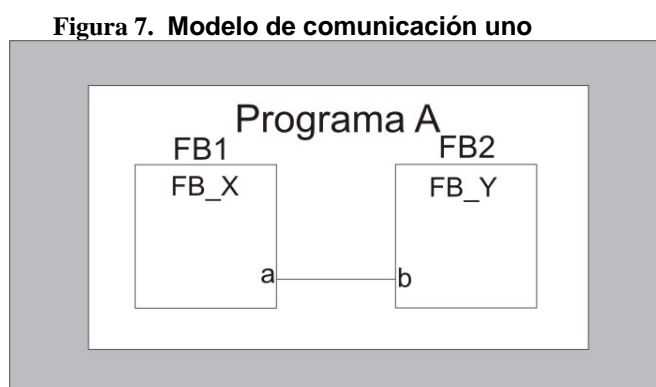
Variables. Asignan direcciones del hardware: E/S, memoria y datos. Locales o globales. Hacen la programación independiente del hardware.

- Modelos de software.



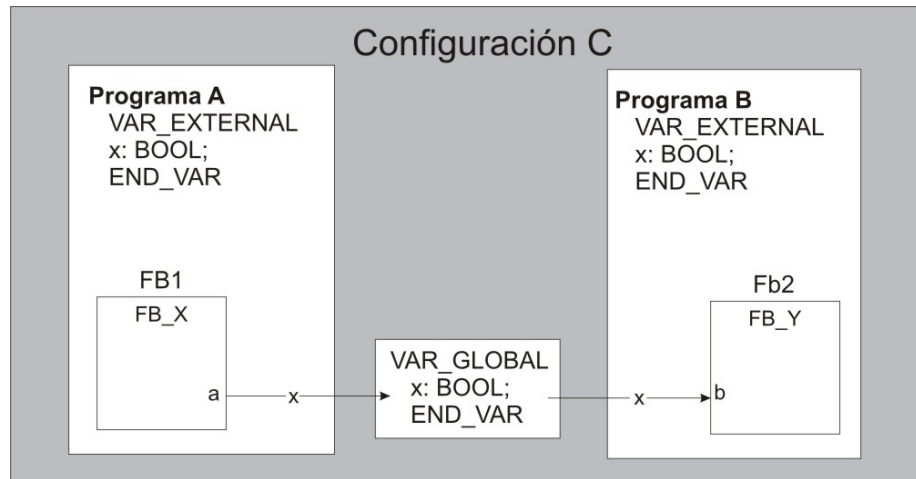
Fuente: IEC 61131-3, Modelo de software

- Modelos de comunicación de datos



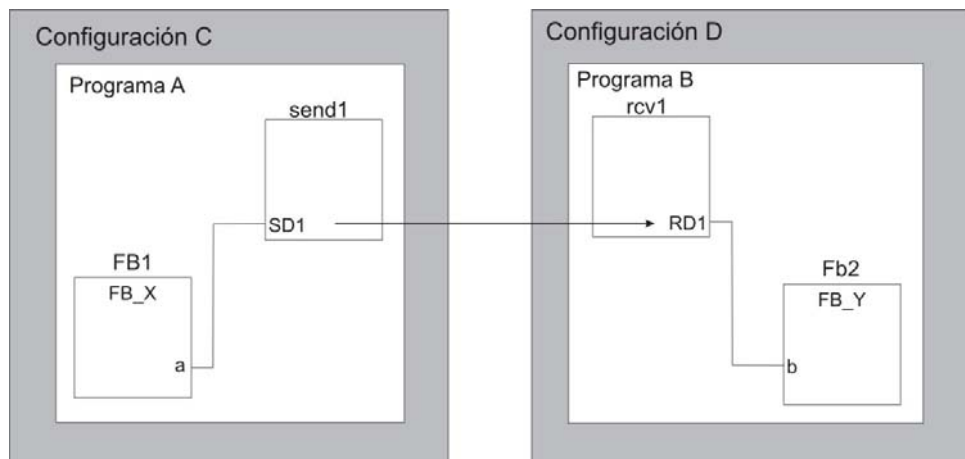
Fuente: IEC 61131-3, Modelo de comunicación uno

Figura 8. Modelo de comunicación dos



Fuente: IEC 61131-3, Modelo de comunicación dos.

Figura 9. Modelo de comunicación tres



Fuente: IEC 61131-3, Modelo de comunicación tres.

- Modelo de programación

Tipos de datos derivados

Unidades de organización de programa:

- Funciones
- Bloques funcionales
- Programas

Elementos del diagrama secuencial (SFC)

Elementos de configuración:

Variables globales

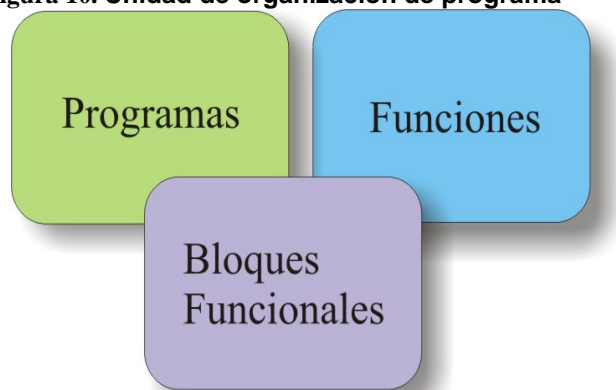
Recursos

Tareas

Vías de acceso

- o Unidades de organización del programa

Figura 10. Unidad de organización de programa



Fuente: IEC 61131-3, Unidad de organización de programa.

Funciones

Una función se define como una unidad de organización del programa que al ser ejecutada suministra exactamente un elemento de datos y cuya invocación se puede utilizar en lenguajes literales como operando en una expresión.

Las funciones no deben 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). Existen funciones estándar y Funciones definidas por el usuario

Bloques funcionales:

Un bloque funcional es una unidad de organización del programa que al ser ejecutada suministra uno o más valores. Existe la posibilidad de crear múltiples (copias) de un bloque funcional, denominadas instancias.

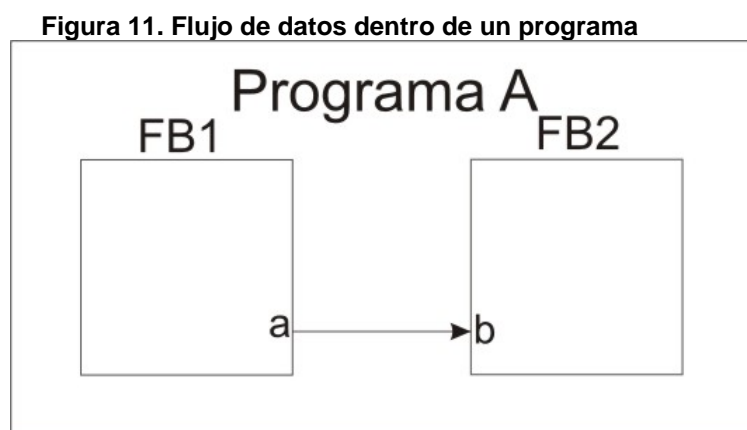
Cada instancia llevará asociado un identificador (el nombre de la instancia) y una estructura de datos que contenga sus variables de salida e internas.

Todos los valores de las variables de salida e internas deberán persistir de una ejecución del bloque funcional al siguiente, por lo que la llamada de un mismo bloque funcional con los mismos argumentos (parámetros de entrada) no tiene por qué dar siempre los mismos valores de salida.

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 autómeta programable”.

Flujo de datos dentro de un programa:

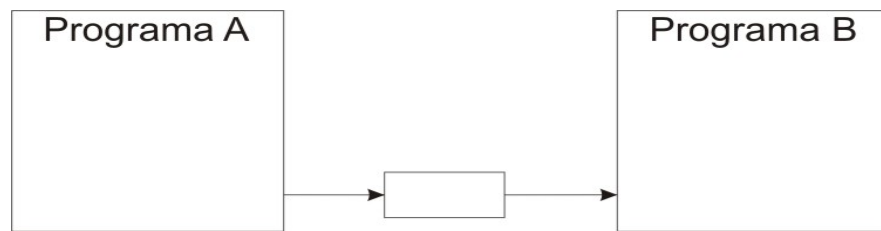


Fuente: IEC 61131-3, Flujo de datos dentro de un programa

Comunicación entre programas.

En la misma configuración (variables globales), o en configuraciones distintas (vías de acceso, comunicaciones)

Figura 12. Comunicación entre programas



Fuente: IEC 61131-3, Comunicación entre programas

- o Elementos de configuración

Una configuración se compone de:

RECURSOS
TAREAS
VARIABLES GLOBALES
VIAS DE ACCESO

Esta es la exposición de los elementos comunes, que estructuran las generalidades de los lenguajes de programación. Se presenta a continuación un ejemplo que muestra grafica y textualmente la definición de bloques funcionales y programas.

Figura 13. Ejemplo de elementos de configuración

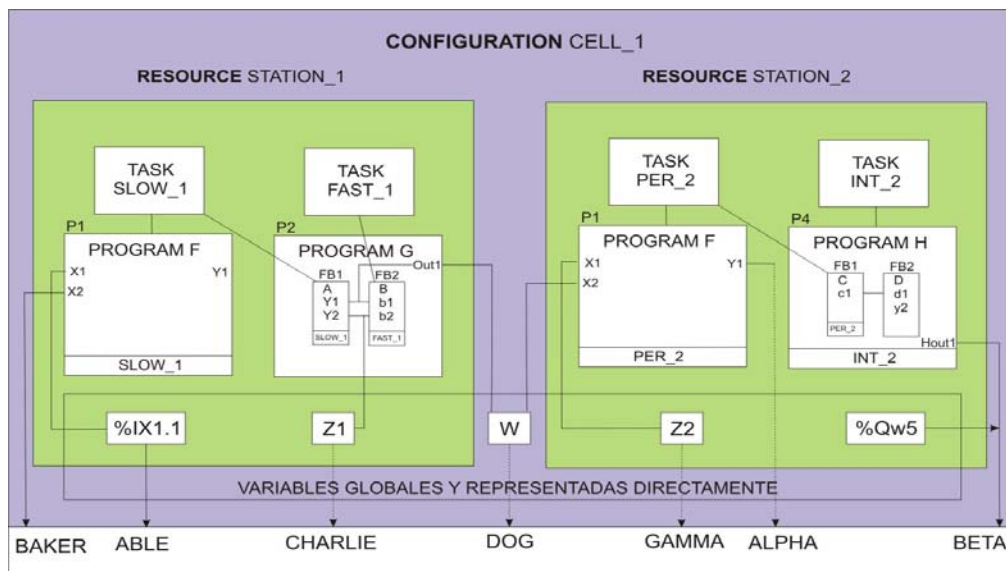


Figura 14. Declaración de Bloque funcionales y parámetros

<pre> FUNCTION_BLOCK A VAR_OUTPUT y1 : UINT; y2 : BYTE; END_VAR END_FUNCTION_BLOCK </pre>	<pre> FUNCTION_BLOCK B VAR_INPUT b1 : UINT; b2 : BYTE; END_VAR END_FUNCTION_BLOCK </pre>	<pre> FUNCTION_BLOCK C VAR_OUTPUT c1 : BOOL; END_VAR END_FUNCTION_BLOCK </pre>
<pre> PROGRAM F VAR_INPUT x1 : BOOL; x2 : UINT; END_VAR VAR_OUTPUT y1 : BYTE; END_VAR END_PROGRAM </pre>		
<pre> PROGRAM G VAR_OUTPUT OUT1 : UINT; END_VAR VAR_EXTERNAL Z1 : BYTE; END_VAR VAR FB1 : A; FB2 : B; END_VAR FB1(...); OUT1 := FB1.Y1; Z1 := FB1.Y2; FB2(B1 := FB1.Y1); B2 := FB1.Y2; END_PROGRAM </pre>		
<pre> PROGRAM H VAR_OUTPUT HOUT1 : INT; END_VAR VAR FB1 : C; FB2 : D; END_VAR FB1(...); FB2(D1 := FB1.C1); HOUT1 := FB2.Y2; END_PROGRAM </pre>		

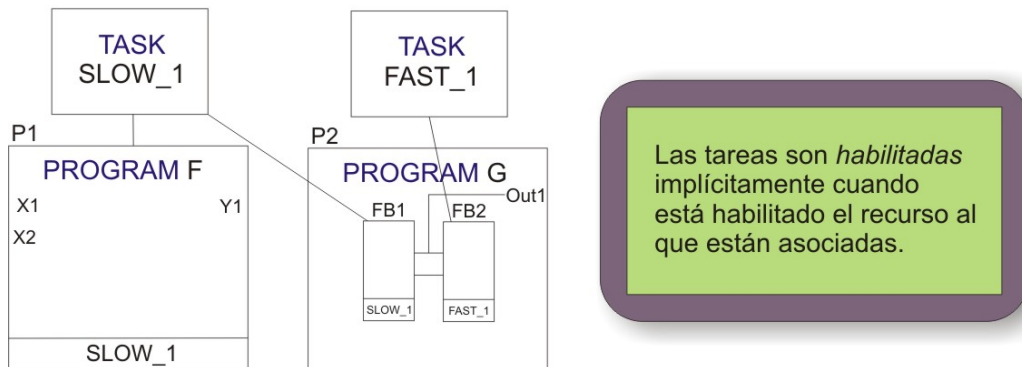
Figura 15. Configuración, recursos y variables

```

CONFIGURATION CELL_1
VAR_GLOBAL w : UINT; END_VAR
RESOURCE STATION_1 ON PROCESSOR TYPE_1
VAR_GLOBAL z1 : BYTE; END_VAR
TASK SLOW_1(INTEGRAL := t#20ms, PRIORITY := 2);
TASK FAST_1(INTERVAL := t#10ms, PRIORITY := 1);
PROGRAM P1 WITH SLOW_1;
    F(x1 := %IX1.1);
PROGRAM P2 : G(out1 => w,
    FB1 WITH SLOW_1,
    FB2 WITH FAST_1);
END_RESOURCE
RESOURCE STATION_2 ON PROCESSOR TYPE_2
VAR_GLOBAL z2 : BOOL;
    AT %QW5 : INT;
END_VAR
TASK PER_2(INTERVAL := t#50ms, PRIORITY := 2);
TASK INT_2(SINGLE := z2, PRIORITY := 1);
PROGRAM P1 WITH PER_2;
    F(x1 := z2, x2 := w);
PROGRAM P4 WITH INT_2;
    H(HOUT1 => %QW5,
    FB1 WITH PER_2);
END_RESOURCE
VAR_ACCESS
ABLE : STATION_1.%IX1.1 : BOO READ_ONLY;
BAKER : STATION_1.P1.X2 : BOO READ_ONLY;
CHARLIE : STATION_1.z1 : BOO READ_ONLY;
DOG : w : BOO READ_ONLY;
ALPHA : STATION_2.P1.y1 : BOO READ_ONLY;
BETA : STATION_2.P4.HOUT1 : BOO READ_ONLY;
GAMMA : STATION_2.z2 : BOO READ_ONLY;
END_VAR
END_CONFIGURATION
    
```

Por ultimo se tiene el elemento de configuración tareas, Tarea es un elemento de control de ejecución que es capaz de iniciar la ejecución de un conjunto de POU's: programas y bloques funcionales, cuyas instancias están en la declaración de los programas.

Figura 16. Representación de tareas
RESOURCE STATION_1



Fuente: IEC 61131-3, Representación de tareas

Operador	Grupo de Operadores	Descripción
R	Salida	Asigna un Falso al operando que lo sigue si el resultado del CR es 1
)	Salida	Paréntesis de Cierre, evalúa la operación que encierra
ADD	Proceso	Suma el operando con el valor del CR
ADD(SUB	SUB(Proceso	Sustrae al CR el operando
MUL	Proceso	Multiplica el operando con el CR
MUL(DIV	DIV(Proceso	Divide el CR por el operando
GT	GT(Proceso	CR > Operando (Resultado Booleano)
GE	Proceso	CR >= Operando (Resultado Booleano)
GE(EQ	Proceso	CR = Operando (Resultado Booleano)
EQ(NE	Proceso	CR <> Operando (Resultado Booleano)
NE(LE	Proceso	CR <= Operando (Resultado Booleano)
LE(LT	Proceso	CR < Operando (Resultado Booleano)
LT(

Fuente: IEC 61131-3, Operadores.

El operando hace referencia a las variables sobre las que se opera y estas son de algún tipo de dato soportado por el lenguaje. Por último tenemos el comentario debe empezar con (*) y terminar con *).

Funciones y bloques funcionales. Se coloca el nombre en el campo del operando y los parámetros, si los lleva, entre paréntesis. Se emplea el operador CAL.

- **Texto estructurado (ST)**

Es un lenguaje textual denominado de alto nivel, por su completa funcionalidad y amplitud de operadores, posee grandes similitudes con lenguajes de programación de PC, como Pascal o C; sus características son:

- Estructuración en bloques.
- Es una construcción sintáctica que al ser evaluada proporciona un valor.
- Esta compuesta por operadores y operandos.
- Facilita la programación de procesos que requieren instrucciones complejas y cálculos grandes.

Un programa en ST consiste en un número de declaraciones separadas por punto y coma (;), lo cual significa que las declaraciones pueden ser extendidas en una línea o en múltiples líneas. De igual forma que en IL los comentarios empiezan con (*) y terminan con *) y estos pueden estar en cualquier parte del programa ST.

Tabla 2. Estructuras ST

Etiqueta	Descripción	Ejemplo	Explicación
:=	Asignación	d:=10;	Asignación de un valor calculado en la derecha y es asignado en el de la izquierda.
	Llamado de un FB	FBNombre(P1:= 10; P2:=20);	Llamado de otra POU de tipo FB incluyendo sus parámetros.
RETURN	Retorno	RETURN;	Salida de un POU y retorna el valor de llamada del POU.
IF	Condición	IF d<e THEN F:=1; ELSIF d=e THEN F:=2; ELSE f:=3; END_IF	Selección de alternativas por alguna expresión booleana.
CASE	Selección Múltiple	CASE f OF 1: g:=11; 2: g:=FunA(); ELSE g:=12; END_CASE	Selección de un bloque de instrucciones, dependiendo del valor de la expresión “f”.
FOR	Bucle “Para”	FOR h:=1 TO 10 BY 2 DO F[h/2]:=h; END_FOR;	Múltiples ciclos de un bloque de instrucciones, con condición tanto de inicio como de fin y un valor de incremento
WHILE	Bucle “Mientras”	WHILE m>1 DO n := n / 2; END_WHILE;	Múltiples ciclos de un bloque de instrucciones, con condición de finalización.
REPEAT	Bucle “Repetir hasta”	REPEAT l:=i*j; UNTIL i<10000 END_REPEAT;	Múltiples ciclos de un bloque de instrucciones con condición de finalización.
EXIT	Finaliza el Ciclo	EXIT;	Terminación prematura de una iteración en un

Etiqueta	Descripción	Ejemplo	Explicación
;	Final de Instrucción	;;	bucle

Fuente: IEC 61131-3, Estructuras ST

Dentro del lenguaje ST se encuentran los operandos y operadores, los primeros se pueden clasificar como sigue:

- o Literales (numéricos, alfanuméricos caracteres, de tiempo)
- o Variables (simples, multi-elementos)
- o Llamado de funciones (Nombrefuncion(argumentos))
- o Expresiones extendidas (10+20)

Tabla 3. Operadores ST

Operador	Descripción	Ejemplo y Valor de la Expresión	Prioridad
()	Paréntesis	(2 * 3) + (4 * 5) Valor: 26	Alta
	Llamado a una Función	CONCAT('AB','CD') Valor: 'ABCD'	
**	Exponenciación	3**4 Valor: 81	
-	Negación	-10 Valor: -10	
NOT	Complemento	NOT TRUE Valor: FALSE	
*	Multiplicación	10*20 Valor:200	
/	División	20/10 Valor: 2	
MOD	Módulo	17 MOD 10 Valor: 7	
+	Suma	1.4 + 2.5 Valor: 3.9	
-	Resta	3 - 2 - 1 Valor: 0	
<, >, <=, >=	Comparación	10 > 20 Valor: FALSE	
=	Igualdad	T#26h = T#1d2h Valor: TRUE	
<>	Desigualdad	8#15 <> 13 Valor: FALSE	

&, AND	AND Lógico	TRUE AND FALSE Valor: FALSE	
XOR	OR exclusivo lógico	TRUE XOR FALSE Valor: TRUE	
OR	OR Lógico	TRUE OR FALSE Valor: TRUE	Baja

Fuente: IEC 61131-3, Operadores ST.

- **Diagrama de Bloques Funcionales (FBD)**

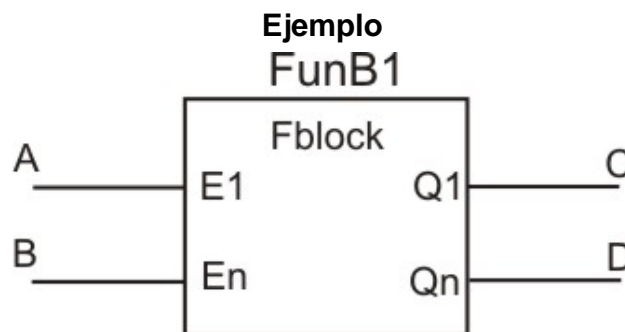
Es un lenguaje gráfico, y sus representaciones se dan mediante caracteres ISO/CEI 646 o elementos gráficos o semigráficos. Sus características básicas son:

- Los programas son bloques cableados entre sí de forma análoga al esquema de un circuito.
- Posee una interfase de E/S bien definida y además posee un código interno oculto.
- Posee conjunto de funciones y bloques bien estructurados.
- Su programación es estructurada.

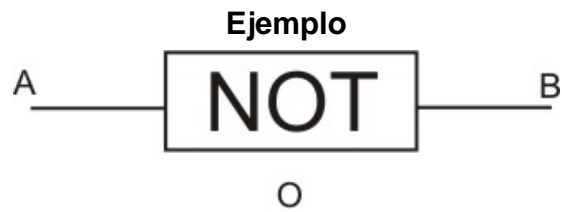
Dentro del FBD, para la estructuración de un programa, la parte de declaración puede ser definida de forma textual o gráfica, dependiendo del soporte que se tenga para el lenguaje; siendo la más utilizada la declaración textual, la parte de codificación está dividida en redes y las redes consisten en; etiqueta de la red, comentario de la red y el gráfico de la red.

Tabla 4. Elementos FBD

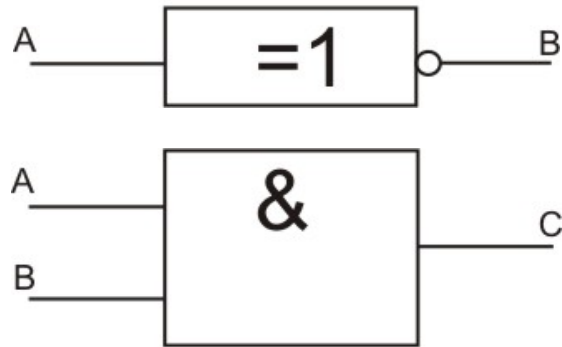
Descripción
Llamado a un bloque funcional (Entre ellos: FlipFlop, Temporizador, Contador, detectores de flanco, etc)



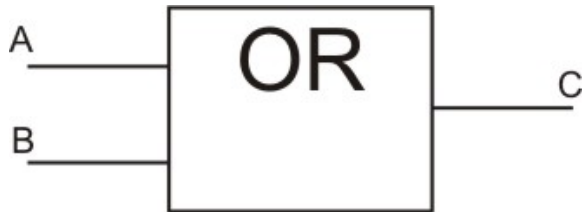
Descripción
Complemento



AND Lógico



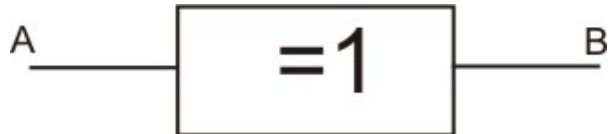
OR Lógico



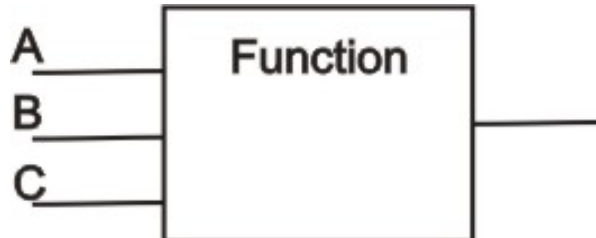
XOR Lógico



Asignación



Llamado de funciones (Entre ellas: ABS, MUL, DIV, MOD, CONCAT, etc.)



Finalmente se tiene las reglas de ejecución del FBD

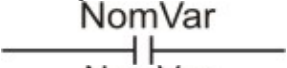


- El bloque se ejecuta cuando todas sus entradas han sido evaluadas
- El bloque se evalúa por completo cuando se han calculado todas sus salidas.
- La evaluación de un conjunto de bloques termina cuando se calculan todas y cada una de sus salidas.


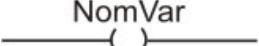
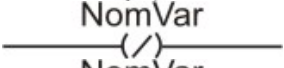
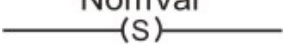
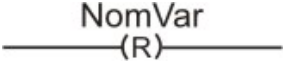
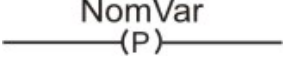
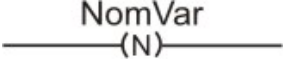
• **Esquema de Contactos (LD)**

Es un lenguaje grafico, es el lenguaje de más antigua utilización, de esto se deriva que presente similitud con los diagramas eléctricos de escalera utilizados por los técnicos, anteriormente a la programación del autómata. Sus características básicas son:

- Las decisiones de control son programas en ejecución.
- Se ejecuta por líneas, que contienen las entradas y salidas; las entradas permiten una comparación o test de las condiciones y se obtiene el resultado de la evaluación. La salida examina el resultado de la evaluación y si es verdadera ejecuta alguna operación o función.
- Las instrucciones de entrada pueden ejecutarse mediante relaciones lógicas AND y OR de forma sencilla; si las instrucciones están en serie se evalúa AND y en caso que estén en paralelo se evalúa OR, de igual forma la salida en paralelo permite activar varias operaciones o funciones con el mismo resultado de la evaluación.

Tabla 5. Elementos LD

Objeto Gráfico	Nombre	Explicación
	Contacto Abierto	$CD^a := CI^b \text{ AND nomVar}$
	Contacto Cerrado	$CD := CI \text{ AND NOT nomVar}$
	Transición Positivo	a $CD := \text{TRUE}$, Cuando: <ul style="list-style-type: none"> • La variable tenía el valor de FALSE en el ciclo de ejecución anterior y • La variable ahora tiene el valor de TRUE (en el ciclo de ejecución actual) y • CI tiene como valor TRUE.

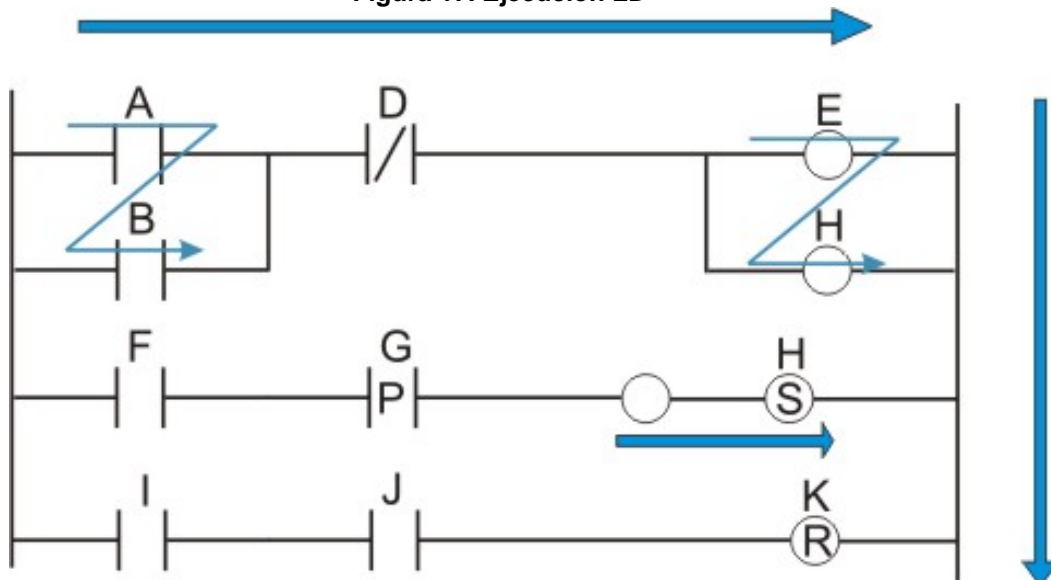
Objeto Gráfico	Nombre	Explicación
	Transición a Negativo	<p>:= FALSE, En cualquier otro caso</p> <p>CD := TRUE, Cuando:</p> <ul style="list-style-type: none"> • La variable tenía el valor de TRUE en el ciclo de ejecución anterior y • La variable ahora tiene el valor de FALSE (en el ciclo de ejecución actual) y • CI tiene como valor TRUE
	Bobina	nomVar := CI
	Bobina Negada	nomVar := NOT CI
	Activación (Set)	nomVar := TRUE, si CI tiene el estado de TRUE, si esto no ocurre: nomVar no cambia de valor.
	Desactivación (Reset)	nomVar := FALSE, si CI tiene el valor de TRUE, si esto no ocurre: nomVar no cambia de valor.
	Bobina de transición a positivo	nomVar := TRUE, cuando
	Bobina de transición a negativo	<ul style="list-style-type: none"> • CI tenía el valor de FALSE en el ciclo de ejecución anterior y • CI tiene ahora el valor de TRUE (en el presente ciclo de ejecución) <p>Si esto no ocurre: la variable no cambia de valor.</p> <p>nomVar := TRUE, Cuando:</p> <ul style="list-style-type: none"> • CI tenía el valor de TRUE en el ciclo de ejecución anterior y • CI ahora tiene el valor de FLASE (en el presente ciclo de ejecución).

Fuente: IEC 61131-3, Elementos LD

- CD es la sigla equivalente para este caso de Conexión por Derecha
- CI es la sigla equivalente para este caso de Conexión por Izquierda

La ejecución del LD se da de izquierda a derecha y de arriba abajo, las líneas con bifurcaciones se ejecutan de arriba izquierda a abajo derecha.

Figura 17. Ejecución LD



Fuente: IEC 61131-3, Ejecución LD

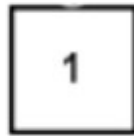
- **Esquema Secuencial de Funciones (SFC)**

Es SFC es un lenguaje, de representación gráfica primordialmente, sin embargo; posee la representación textual de sus elementos gráficos, proviene de un estándar anterior IEC 848 que fue construido basando en el Gráfico funcional de control etapa transición (Grafcet). Sus características básicas son:

- Describe el comportamiento secuencial de un proceso y programa.
- Usado para distribuir problemas de control.
- Permite un rápido diagnóstico.
- Las etapas o estados implican acciones asociadas.
- Las transiciones gobiernan los cambios de estado
- Pueden darse esquemas que poseen divergencia y convergencias, así como ciclos.
- Se pueden definir secuencias alternativas o paralelas.

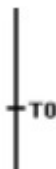
Los elementos gráficos del SFC son:

Figura 18. Etapa



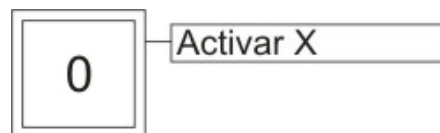
Estado específico del diagrama en el cual se da la orden de ejecutar sus acciones asociadas cuando se active y que permanece en ese estado hasta el momento en que la condición de alguna de sus transiciones salientes sea verdadera (La etapa inicial se especifica por medio de dos recuadros concéntricos).

Figura 19. Transición



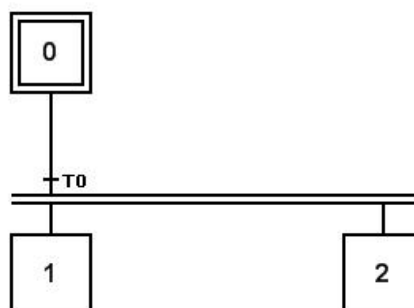
Pasaje de una etapa a otra, Tiene asociada una condición (función booleana combinatorial), su condición es validada cuando alguna de las etapas anteriores está activa.

Figura 20. Acción



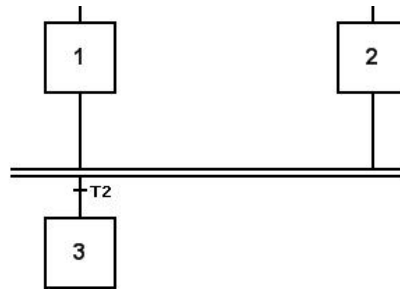
Operaciones a realizar sobre el sistema cuando la etapa a la que está asociada está activa.

Figura 21. Divergencia en Y



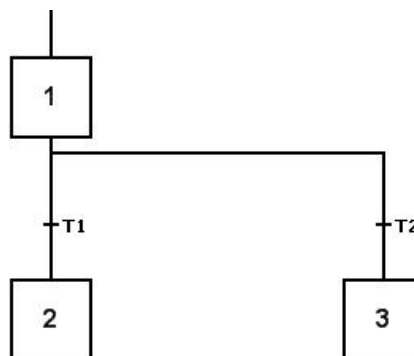
Asocia varios caminos diferentes agrupándolos con una condición común que al cumplirse desata una ejecución simultanea de las etapas a las cuales están enlazadas. Toda Divergencia en Y debe terminar en una Convergencia en Y.

Figura 22. Convergencia en Y



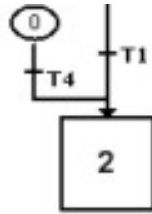
Es la manera de cerrar dos o más caminos diferentes con ejecución simultanea gracias a una divergencia en Y asociando una condición como común a las líneas que se van a cerrar, de tal forma que si se cumple estos caminos terminan y convergen en una única etapa de destino que continua con el proceso.

Figura 23. Divergencia en O



Es cuando una etapa tiene asociadas mas de una transición saliente, con condiciones combinatoriales lógicas excluyentes entre si para evitar conflictos e incongruencias. Cada una de esas condiciones tiene como destino un camino diferente lo que hace que haya una continuación de proceso de manera alternativa.

Figura 24. Convergencia en O



Es cuando una etapa tiene asociadas mas de una transición entrante, cuyas condiciones combinatoriales lógicas son excluyentes entre si. Convirtiéndose en un destino alternativo para varios caminos del proceso. Cuando una de las condiciones se cumple y la etapa inmediatamente superior está activa se produce un flanqueamiento a la etapa destino de la convergencia.

La ejecución del SFC se da en virtud de las líneas de evolución, es decir; después de la etapa de doble rotulo, llamada etapa inicial, se desprenden líneas que se conectan a otra etapa, el orden esta dado por estas líneas y las condiciones consignadas en las transiciones, que permiten la ejecución del grafico diseñado, así la ejecución de las acciones asociadas a las etapas estará sujeta al cumplimiento de la condición de la transición y al orden secuencial de las líneas de evolución.

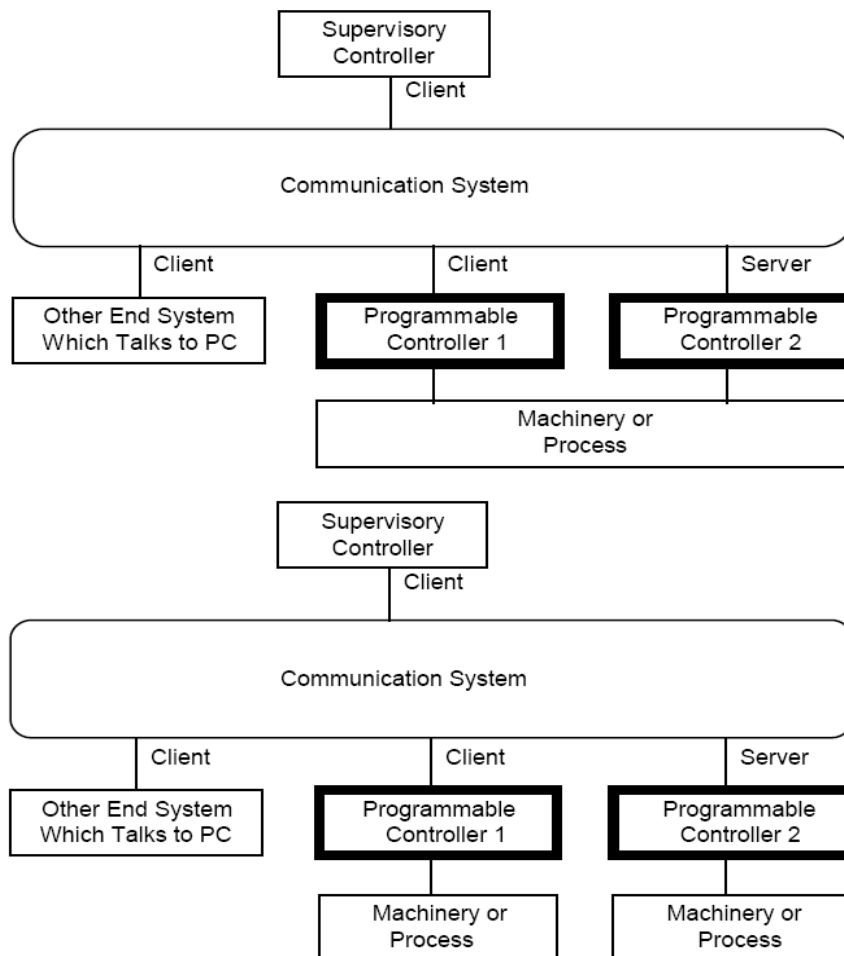
2.3.4 Parte 4. Guía de usuario. La cuarta parte se piensa como una guía para ayudar al usuario de PLC en todas las fases del proyecto de la automatización. La información Práctica-orientada se da en los asuntos que se extienden de análisis del sistema y de la opción del correcto equipo para el mantenimiento.

2.3.5 Parte 5. Especificación de comunicaciones. Esta última parte se refiere a la comunicación entre PLC's de diferente fabrica con ellos y con otros dispositivos. En cooperación con ISO 9506 (especificación del mensaje de la fabricación; las clases de la conformidad de MMS) serán definidas para permitir que PLC's se comuniquen, por ejemplo, vía redes. Éstos cubren la función de la selección del dispositivo, del intercambio de datos, del proceso de alarma, del control de acceso y de la administración de la red.

Se tiene para esta parte en cuenta lo expuesto en las anteriores con relación a:

1. Modelo de comunicación de redes

Figura 25. Modelo de redes



2. Modelo funcional del PC (Ver figura 1.)
3. Modelo hardware del PC (Ver figura 5)
4. Modelos de Software (Ver figura 6)

Al igual que los modelos esta parte presenta:

- Servicios de comunicación de PC, donde se estructura lo concerniente a los subsistemas y que se resume así:

Tabla 6. Presentación de estado

Nº	Estado
1	PC
2	I/O subsistema (incluye entrada y salida del modulo y I/O de otros dispositivos inteligentes)
3	Unidad de procesamiento
4	Sistema suplente de poder
5	Memoria del subsistema
6	Comunicación del subsistema
7	Implementación específica del subsistema

Fuente: IEC 61131-5, Presentación de estados.

- Comunicación PC de bloques funcionales, quizá la parte mas importante para la comunicación, esto debido; a que son los bloques funcionales los encargados de enviar y recibir la información con la cual el sistema actuara, es decir; es a nivel de los bloques funcionales que se interactúa entre dispositivos, por lo cual es para la comunicación la parte mas importante dentro del POU, y se puede observar su estructura así:

Tabla 7. Estructura comunicación BF

Nº	Subclase	Nombre de comunicación (Bloque funcional o función)
1	Semántica de comunicación de parámetros	REMOTE_VAR
2 3	Verificación periférico	STATUS, USTATUS
4	Adquisición de datos	READ
5 6	Programado de datos adquiridos	USEND, URCV
7	Parámetros de control	WRITE
8 9	Control de interconexión	SEND, RCV
10 11	Programación de reportes de alarma	NOTIFY, ALARM
12	Conexión maestra	CONNCTC

- Compilación e implementación específica de funciones y parámetros; se remonta a tablas descritas en la parte uno de la norma donde se indica la compilación y las implementaciones mas usuales.

De esta forma queda condensada la comunicación en la quinta parte de la norma recogiendo la complejidad y completitud en relación con la comunicación para un montaje en automatización.

3. APORTES DEL IEC 1131

3.1 INTRODUCCIÓN

En el presente capítulo se encuentra condensada la interpretación y sustento conceptual que posibilitó el desarrollo de la investigación, por lo cual es el interés mostrar el análisis, acerca de los aportes de la norma al diseño y comunicación en el campo de la automatización.

La investigación que se realizó, desde una perspectiva de una investigación proyectiva, mostrara que frente al problema planteado, una vía de abordaje con relación al diseño, es analizar la fortaleza que evidencia el SFC y para este lenguaje construir los recursos necesarios de forma tal que sea el diseño mediante él, el que logre establecer un puente comunicacional para el campo de la automatización industrial.

3.2 CONCEPTOS E INTERPRETACIONES

Como se anticipaba en la introducción de este capítulo, la primera interpretación será alrededor de la separación del SFC, como lenguaje de programación y situarlo como lenguaje de diseño con capacidad por parte de los dispositivos, de interpretación y ejecución directa sobre la implementación, esto sustentado en lo expuesto en el capítulo anterior, donde por sus elementos constitutivos despliega una gran capacidad de descripción de un automatismo.

En segunda instancia, la denominación del SFC como lenguaje de diseño, arroja como consecuencia favorable al desarrollo de la investigación una forma más específica de comunicación a nivel descriptivo de los proyectos en el campo de la automatización, es decir; al poseer un lenguaje único, bien estructurado y con suficientes recursos para el diseño, se tendrá un puente capaz de interconectar diversos proyectos, con esto se avanzara en la interpretación, mantenimiento y reutilización de los proyectos.

3.3 MANEJO DEL DISEÑO

En el campo de la automatización es frecuente encontrar procesos que implican la realización de una serie de actividades. La ejecución de estas actividades convoca, para el caso de los automatismos, señales. El desarrollo del proceso es una sucesión encadenada de operaciones, cuya

evolución se controla mediante unas condiciones de tipo lógico, que indican si el proceso continua y como debe hacerlo.

Ahora, los automatismos que controlan estos procesos, poseen bloques combinacionales y células de memoria biestable interconectados. De esta forma se tiene que para cualquier lenguaje que aspire realizar el diseño de un automatismo debe tener en cuenta estas condiciones, por tanto debe ser observable que el SFC recoge estas condiciones.

El manejo que se da al diseño, es en consecuencia con el cumplimiento de las condiciones necesarias para los procesos en los que confluyen el bloque combinacional y las células de memoria y que el SFC los maneja desde sus elementos.

Con lo anteriormente expuesto, queda claro que es el SFC un lenguaje que posee las características para ser denominado un lenguaje de diseño y es en virtud de esta interpretación que la investigación avanza hacia la solución de: Primero, el aporte de la norma al fortalecimiento de la labor de diseño desde la focalización del SFC como el lenguaje destinado para que mediante él se diseñe y Segundo, que siendo el SFC un lenguaje de independencia tecnológica posibilita la comunicación en el campo de la automatización industrial, esto por que coloca en igualdad de condiciones a todos los que participan en este campo.

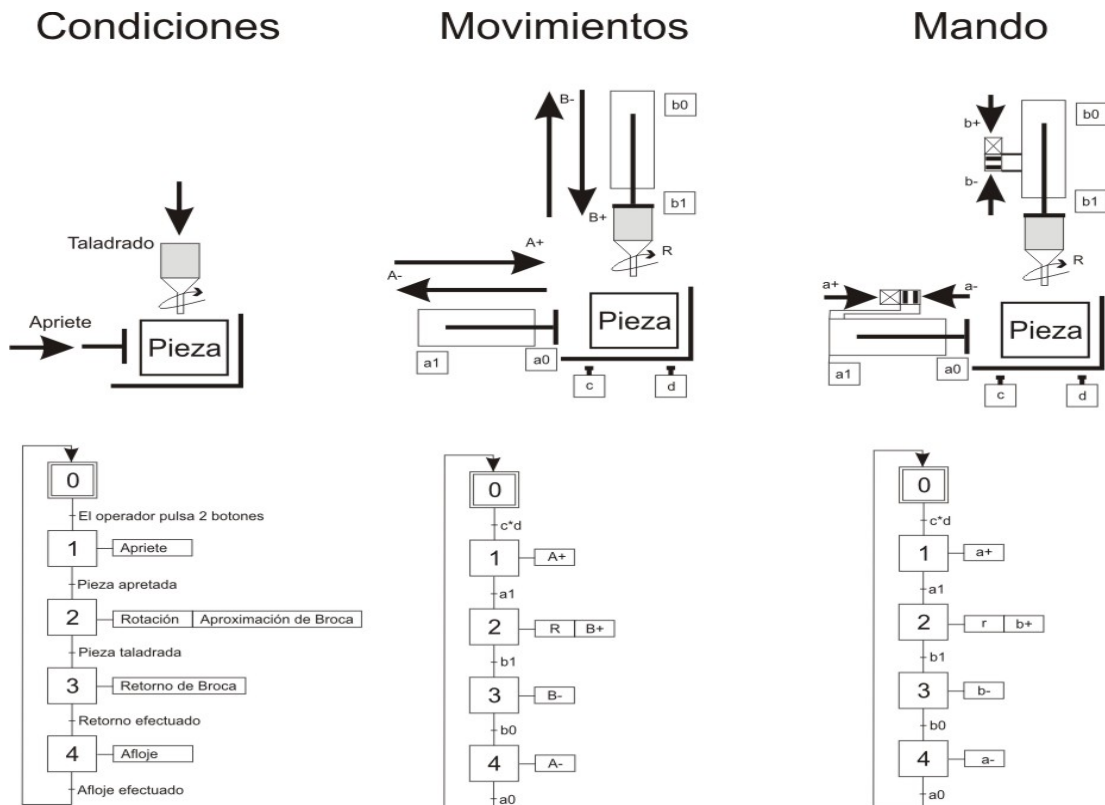
En adelante se mostrara como el SFC, cumple con las condiciones para el diseño de automatismos y su importancia.

3.3.1 Esquema Secuencial De Funciones (SFC). El SFC se hace importante, debido a que se ha diagnosticado que en el campo de la automatización industrial existe una debilidad entorno al diseño, por tanto es el diseño una labor importante que apoyar, esto; por que en el contexto de ingeniería, son los diseños los que marcan la pauta del desarrollo de cualquier proyecto, de un buen diseño se desprende un buen proyecto, y además en la medida de la existencia de un diseño, se posee la capacidad de interpretación, mantenimiento y reutilización de éste.

Por otra parte, y en sintonía con lo anterior, el SFC en la forma de estructurar sus elementos permite unificar la descripción del proceso para técnicos de distintas campos, desde el ingeniero industrial o de producción, que define las necesidades del automatismo, pasando por el de diseño, que debe implementar el sistema de control y los accionamientos, hasta el técnico de mantenimiento, que debe cuidar de su funcionamiento o introducir alguna modificación.

El SFC, posee la capacidad mediante su representación de presentar varios niveles, entre los cuales se puede encontrar. El nivel de condiciones, el de movimientos y el de mando (ver Figura 26. Niveles de diseño SFC), sin embargo; para la generación de codificación que pueda ser ejecutada por un PLC, deberá diseñarse en el último de los niveles, pues es en este nivel donde se relacionan los accionamientos y condiciones.

Figura 26. Niveles de diseño SFC



Fuente: Autómatas programables, Niveles de Diseño SFC

Para la realización de un diseño mediante SFC para automatismos lógicos secuenciales es necesario tener en cuenta los siguientes aspectos:

- Caracterizar el funcionamiento del automatismo independientemente de los componentes con los que se vaya a construir.
- El elemento fundamental es la operación, entendida como una acción ha realizar por el automatismo, que pueden ser complejas o simples, las complejas deben poderse subdividir en otras mas elementales.
- Las acciones ha realizar deben ser elementales, de forma tal que solo dependan de relaciones combinatoriales entre entradas y salidas.
- Se debe establecer un grafico de evolución que indique la secuencia de operaciones y las condiciones lógicas para pasar de una a otra, así

se obtendrán las ecuaciones lógicas de las variables de estado. Y con esto queda establecida la parte secuencial del automatismo.

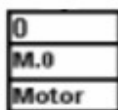
Ahora, como se mostró en el capítulo anterior, donde se expuso los elementos del lenguaje, que por la interpretación adoptada son elementos de diseño, el diseño se maneja entorno a ellos, por lo tanto; es necesario precisar que implicaciones tiene cada elemento.

A. Elementos de diseño

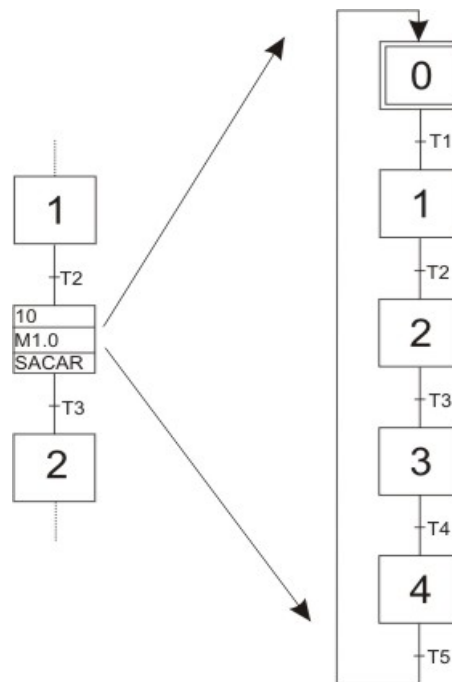
Estos elementos constituyen los símbolos a partir de los cuales se diseña mediante el gráfico funcional. Los elementos son:

- Las etapas: representan cada uno de los estados del sistema, cada etapa tiene asociada por lo menos una acción, y ésta debe darse mediante una relación combinacional. Cada etapa corresponde a un estado del sistema por lo cual cada etapa tiene asociada una variable de estado¹ que es la encargada de generar la ejecución de la acción, de igual forma; esta asociada a la etapa un número de identificación para ella, que dependerá del orden de agregación al diseño.
- Las macroetapas: Como se mostró en el capítulo anterior las etapas pueden albergar subrutinas o subprocesos que pueden tener la misma magnitud de complejidad que un diseño normal, pues se ha introducido para el cubrimiento de estas situaciones un elemento que tiene las mismas características básicas de la etapa, en relación a las variables de estado, sin embargo; se ha otorgado la posibilidad que con la introducción de este elemento de diseño se logre poseer procesos interiores con el grado de complejidad que se requiera, haciendo mucho más organizada la elaboración del diseño y del proyecto de automatización.

Figura 27. Macroetapa



¹ Variable de estado o interna de estado son llamadas a las variables que elabora el sistema de las entradas y eventualmente de otras variables internas. Para el caso las variables de estado asociadas a las etapas son variables que dependen de las condiciones combinacionales y de otras variables de estado de otras etapas, todas estas relacionadas con los elementos de memoria biestables.



- Las líneas de evolución: son líneas que enlazan o unen a las etapas o macroetapas (ver definición macroetapa) que representan actividades consecutivas; su orientación siempre se entenderá de arriba hacia abajo, a menos que con flechas se indique lo contrario.
- Las transiciones: representan la condición lógica necesaria para que finalice la actividad de una etapa o macroetapa y se inicie la de la o las etapas o macroetapas inmediatamente consecutivas.
- Los reenvíos: Son símbolos en forma de flecha que indican la procedencia o destino de las líneas de evolución. Estas flechas permiten fraccionar el gráfico de forma tal que no se dificulte su entendimiento.
- Divergencia y convergencia en “O”: Conjuntamente a la divergencia y convergencia se le llama bifurcación en “O”, consiste en varios caminos que se pueden dar de forma alternativa, cada uno con su propia condición de transición; esta estructura debe ser globalmente cerrada, implica que; los caminos abiertos por la divergencia deben tener su correspondiente camino de cierre en la convergencia.
- Divergencia y Convergencia en “Y”: Conjuntamente a la divergencia y convergencia se le llama bifurcación en “Y”, consiste en la existencia de varios caminos que se inician simultáneamente cuando se cumpla una condición de transición común, esta estructura debe ser globalmente cerrada, es decir;

el número de camino abiertos por la divergencia deben ser cerrados por la convergencia.

Algunas consideraciones de sintaxis para el SFC son:

- Cuando se recorre el gráfico de evolución, por cualquier camino posible, deben alternarse siempre una etapa y una transición.
- Entre dos etapas debe existir una y solo una condición de transición, que puede venir expresada por una función combinacional con la complejidad que se requiera siempre que de cómo resultado un bit que indique uno para condición verdadera y cero para condición falsa.
- Los números de las etapas nada indican con relación al orden de ejecución, simplemente tiene un carácter de identificación.
- El diseño SFC debe poseer para cada etapa una(s) transición(es) que la conecte con otra u otras etapas o macroetapas.

Con la exposición realizada se ha mostrado, el argumento conceptual del SFC, entorno al manejo del diseño para automatismo.

3.4 COMUNICACIÓN ENTRE DISPOSITIVOS

Seguido al aporte que, dada la interpretación del SFC, realiza la norma al campo del diseño de automatismos, se tiene; que este aporte da inicio al otro punto de interés para la investigación, la comunicación; y se visualizan entonces dos escenarios para exponer los aportes que al respecto tendría la norma, el primero; concatenado a la interpretación del SFC, y que centra su fortaleza en las ventajas que representa el poseer un lenguaje de diseño, que logre sintonizar alrededor de él a todos cuantos participan en un proyecto de automatización, esto representa un avance comunicacional, pues el lograr emprender un proyecto con la posibilidad que profesionales de diferentes áreas hablen en un mismo terreno y además sin la tensión de la particularización del desarrollo a un lenguaje específico que provee el PLC en el cual se implementaría, da por superada la dificultad de comunicación en la etapa de desarrollo del proyecto.

En segundo lugar, está el escenario de la comunicación entre dispositivos, este escenario, posee variables para el análisis que desbordan el tratamiento técnico del tema, como el control comercial por parte de los fabricantes de dispositivos de control, sin embargo; estos aspectos serán analizados en el capítulo siguiente, por lo cual será el sustento técnico que presenta el estándar lo que importe, de este sustento se puede extraer la intención de la

norma en proveer desde los lenguajes ya un acercamiento a la comunicación entre dispositivos, sin embargo; no solo mediante la exhortación a la utilización de estos lenguajes en los dispositivos de control, se estructura la comunicación, la quinta parte del estándar expone de forma completa, como en función de la comunicación es necesario adoptar un modelo de conexión y transferencia de información, un modelo software, donde se hace necesario capacidad de interpretación de lenguajes (los de la norma), y un modelo hardware, al igual que una lógica de procedimiento donde se incorporan conceptos y métodos de procedimiento para la transferencia de datos, esta ultima parte con base a la ISO/IEC 9506-5.

Como complemento a los modelos, la norma dicta de forma muy especifica el funcionamiento de variables, transferencia de señales para generación de controles, la forma de interpretar y responder a estos controles, de igual forma dicta el envío y recibo de señales para inicio o detección de procesos entre dispositivos, y de esta forma va incrementando el margen de acción y utilización de la comunicación entre dispositivos, con una gran ventaja, y es que en la lógica procedimental del estándar no importa de que casa fabricante son los dispositivos de control, ya que todos están en la capacidad de interpretar y emitir información interpretable a la luz del IEC 1131.

En términos técnicos, se visualiza la importancia de los bloques funcionales para la comunicación entre dispositivos, pues son estos los poseedores de la funcionalidad respectiva, y esto representa una ventaja pues el la unidad de organización de programa que se encuentra con mayor desarrollo por parte de la mayoría de los dispositivos de control es esta, pues ha sido la forma de clásica de estructurar la funcionalidad en los dispositivos, con lo cual la incorporación propuesto por la norma no genera dificultades; Revisando los bloques funcionales y su elaboración al respecto de la comunicación se pueden visualizar los siguientes gráficos:

Figura 28. Función REMOTE_VAR

		REMOTE_VAR	
SCOPES	---	SCOPE	--- VAR_ADDR
STRING	---	SC_ID	
STRING	---	NAME	
(Note)	---	SUB	


```

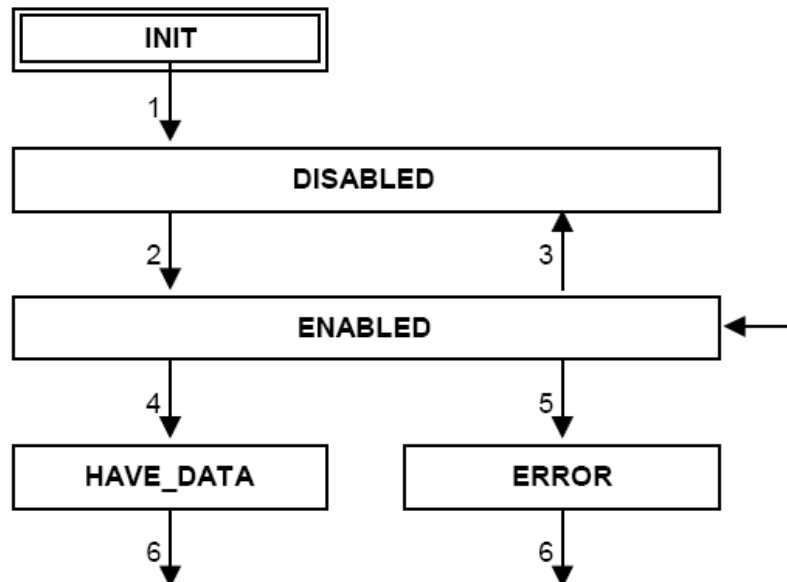
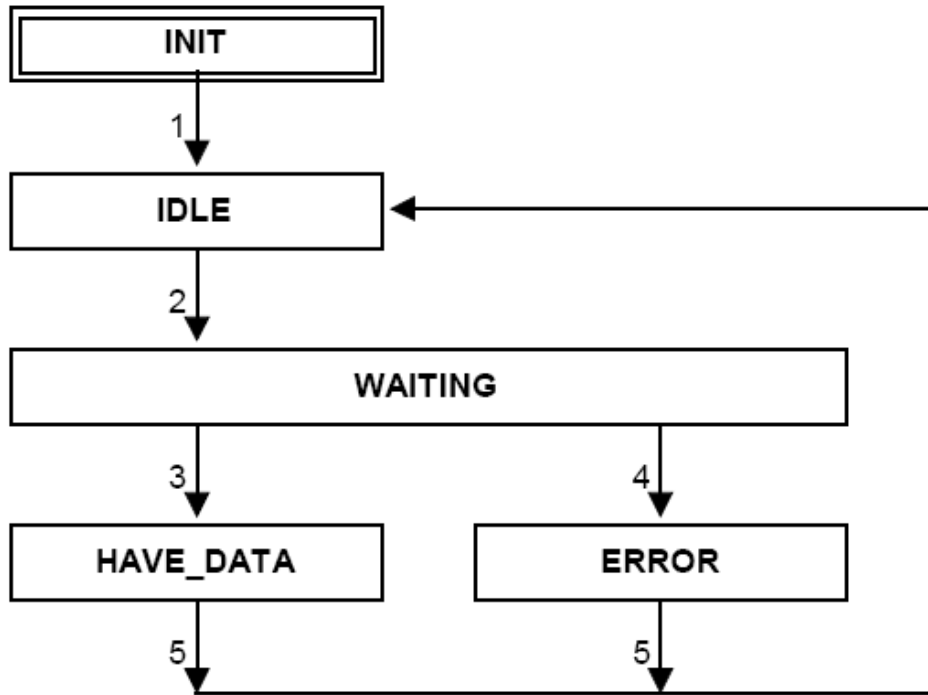
FUNCTION REMOTE_VAR (* Generate remote variable address *)
  : VAR_ADDR;      (* Data which may be used at *)
                  (* VAR_i inputs of the READ and *)
                  (* WRITE function blocks *)

VAR_INPUT
  SCOPE      : UINT;      (* Scope of the variable *)
  SC_ID      : STRING;   (* Identifier of the name scope *)
  NAME      : STRING;   (* Name of the variable *)
  SUB       : (Note);
END_VAR

```

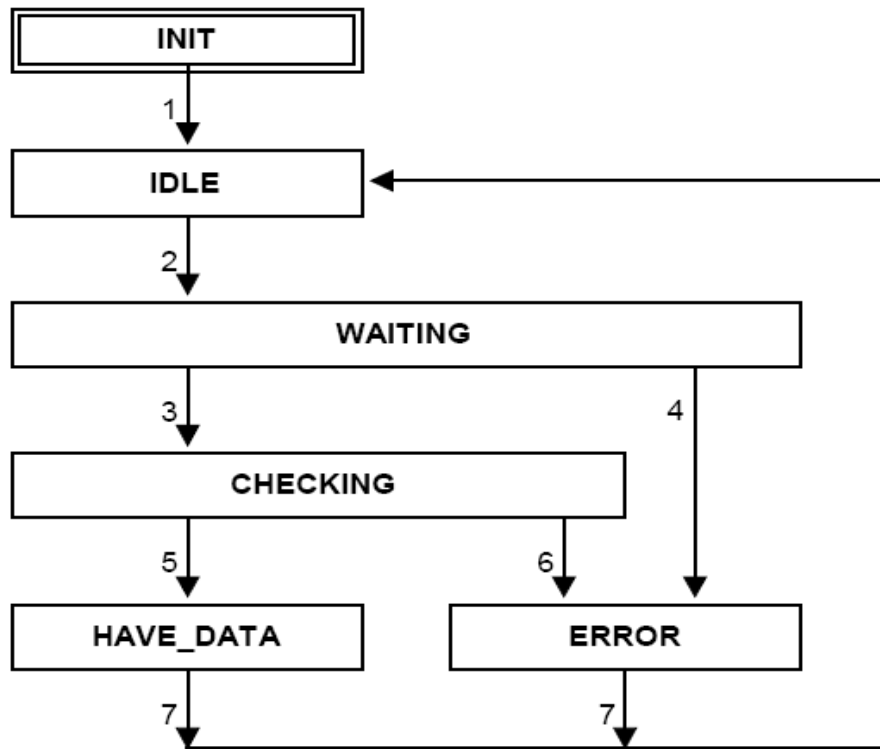
Esta es la función definida para el acceso a información por variables

Figura 29. Estados de Bloque Funcional (STAUS, USTATUS)



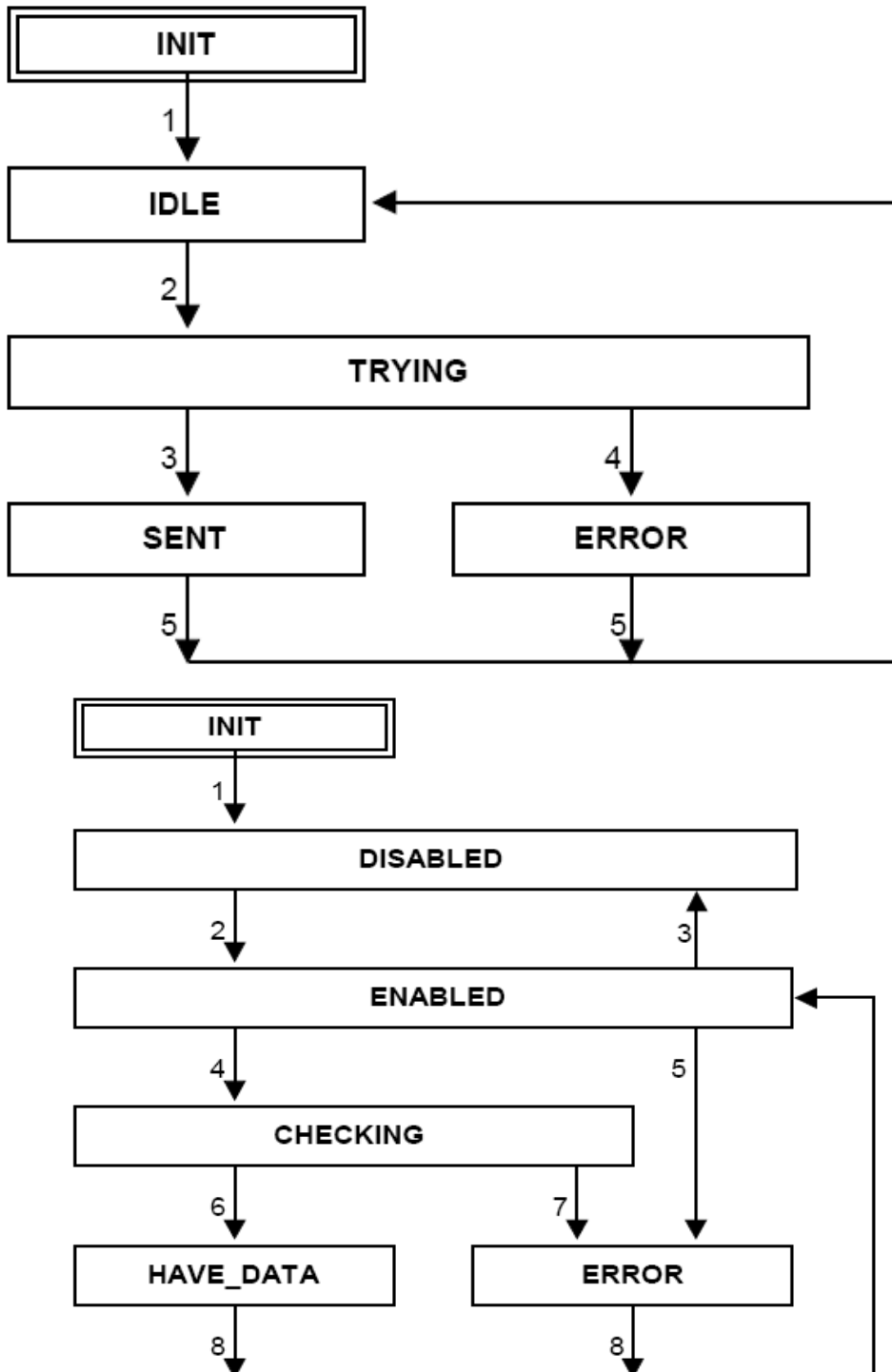
Representan el algoritmo para los estados de verificación de periféricos

Figura 30. Adquisición de datos (READ)



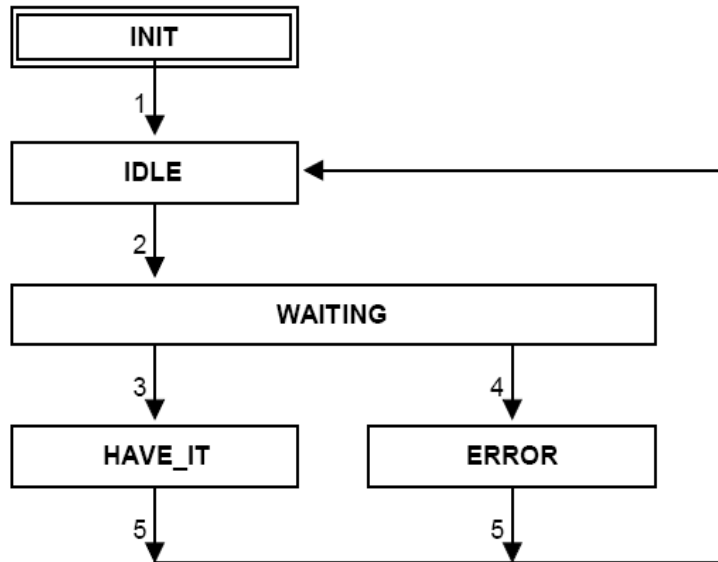
Representa el algoritmo de lectura de bloques funcionales

Figura 31. Programado de datos adquiridos (USEND, URCV)



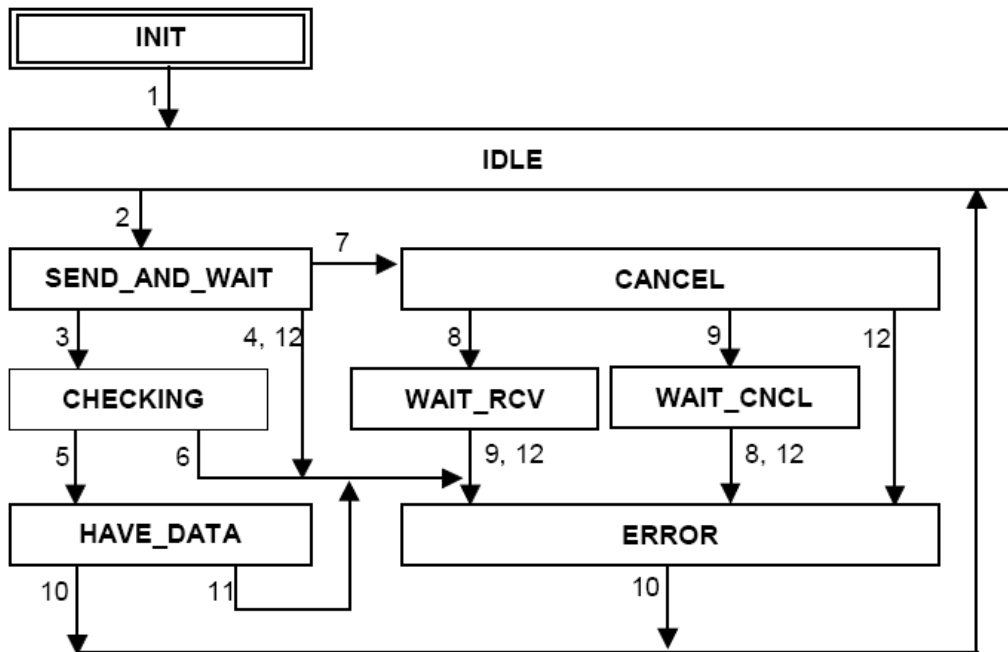
Representan los algoritmos para el tratamiento de los datos recibidos

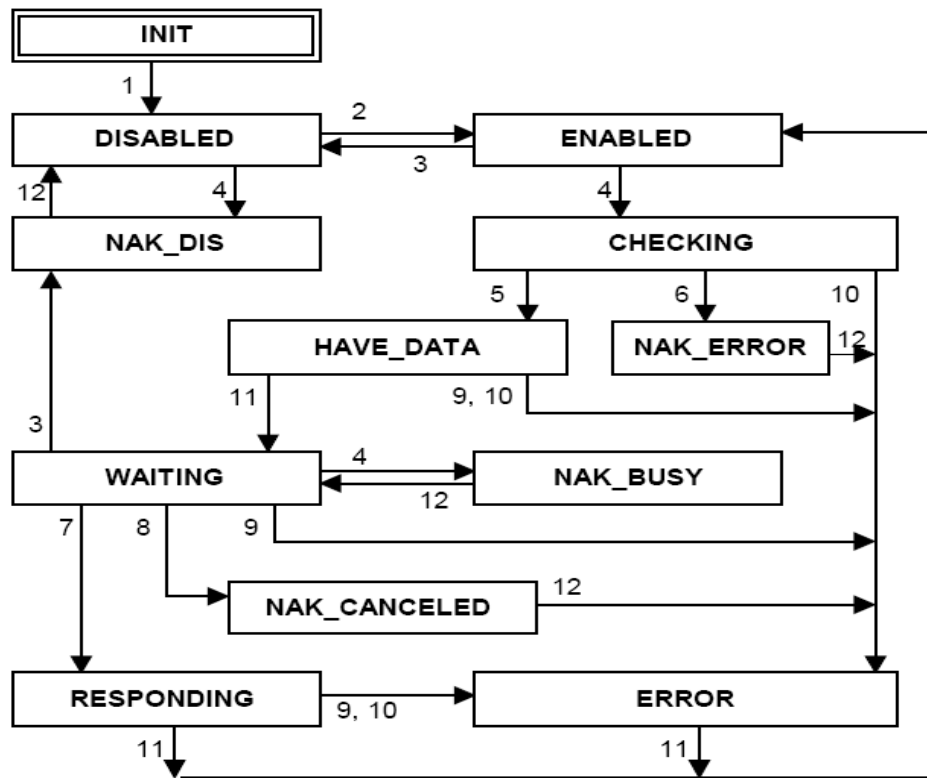
Figura 32. Parámetros de control (WRITE)



Representa el algoritmo para los parámetros de control, lo cual tiene por función la escritura en bloques funcionales.

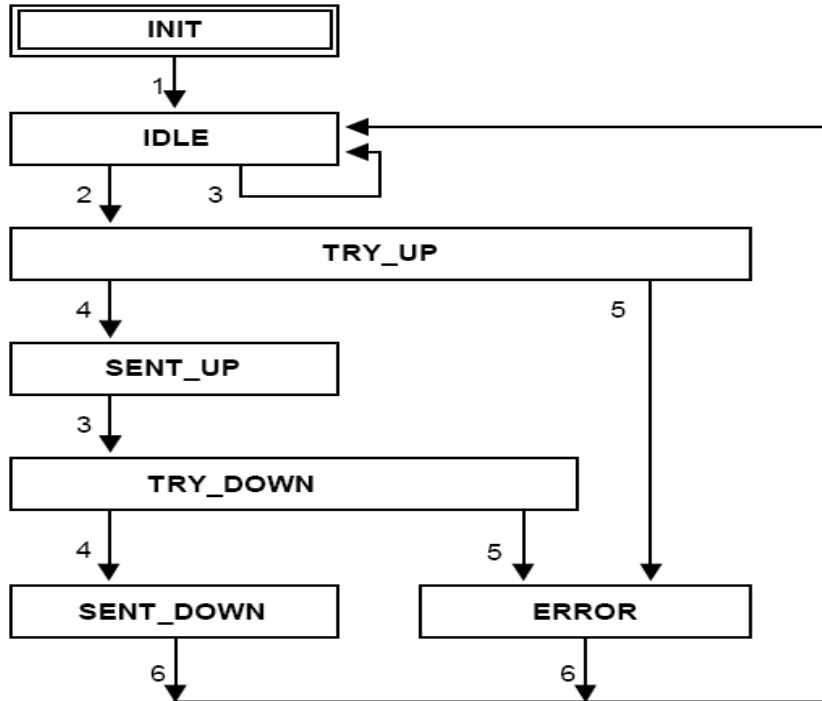
Figura 33. Control de Interconexión (SEND, RCV)

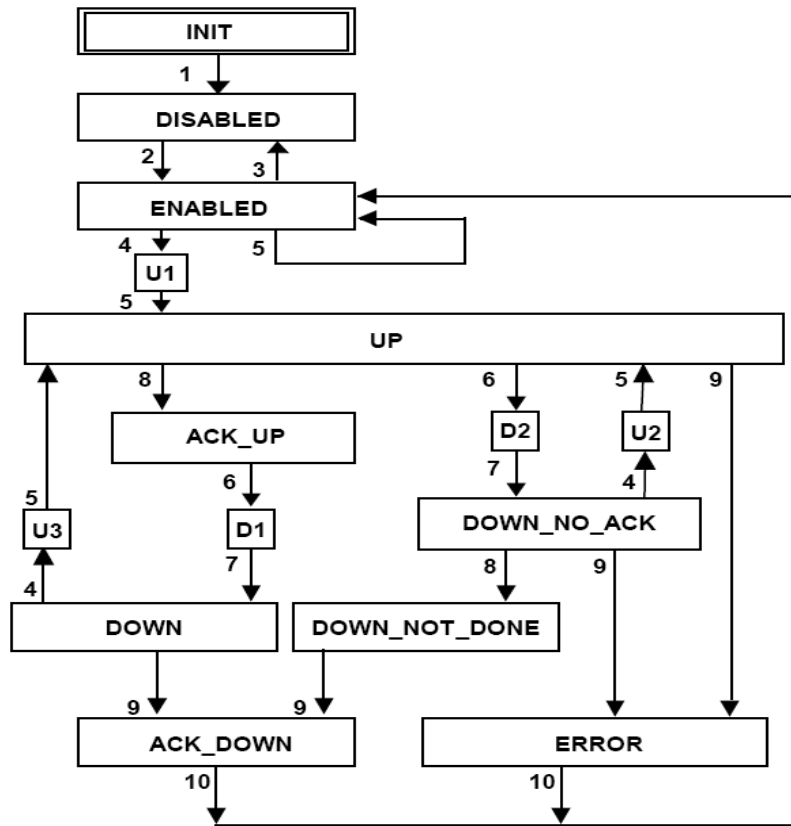




Algoritmos que representan el control de interconexión. Mediante bloques funcionales (SEND, RCV) para el control de canales, trafico, recursos y ejecución.

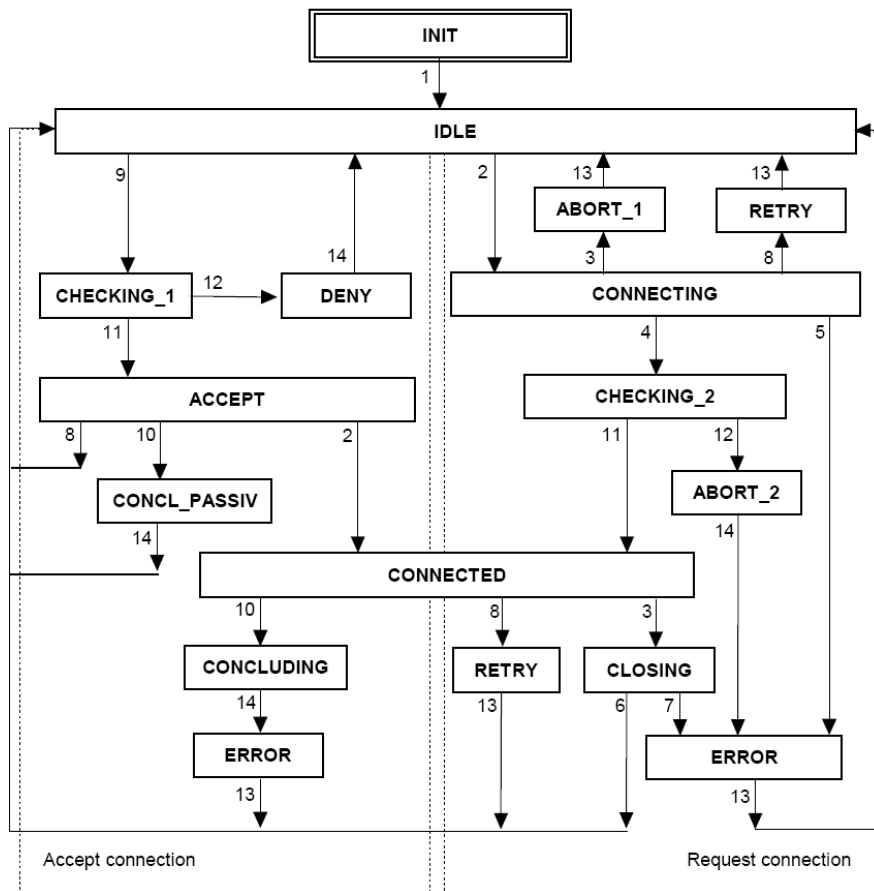
Figura 34. Programación Reporte de Alarmas (NOTIFY, ALARM)





Representan el algoritmo para la generación de alarmas mediante los bloques funcionales (NOTIFY, ALARM).

Figura 35. Conexión maestra (CONNECT)



Representa el algoritmo para la conexión, que es el encargado de establecer la comunicación, mantenerla y definir la forma de transporte, esto mediante el bloque funcional (CONNECTC).

Con lo anteriormente expuesto, se tiene en síntesis del capítulo; que estos dos aspectos puntuales que han interesado a la presente investigación arrojan un resultado favorable, esto sustentado en el avance que constituye la inclusión del SFC dentro de una norma que estructura una metodología de trabajo, y la capacidad de este lenguaje por sus elementos constitutivos de desempeñar una labor de diseño con capacidades de ejecución mediante interpretación por parte de los elementos de control; de igual forma, en términos de la comunicación entre dispositivos se encuentra que se cuenta con la estructuración de un modelo completo capaz dar respuesta a las necesidades de transmisión de señales e interpretación de las mismas al margen de dispositivos concretos de casas fabricantes específicas.

4. APLICABILIDAD DE LA NORMA

El estándar IEC 1131 se aplica a los autómatas programables y sus periféricos asociados, los cuales han sido diseñados para el control industrial.

Ahora, la funcionalidad de un PLC es implementada usualmente sobre plataformas hardware y software específicos, aunque en determinados casos estas implementaciones se pueden realizar en equipos de cómputo personales con características industriales, para cualquiera que sea la forma de implementación, el estándar es aplicable, por que la norma esta dirigida a cualquier producto que implemente la funcionalidad de un PLC.

Con la definición del campo de aplicación de la norma, lo que se tiene es una forma de proceder que engloba todo el desarrollo, y con lo expuesto en los capítulos anteriores lo que fuese de esperar en términos técnicos es la rápida aplicación de ésta, sin embargo; existen factores que han dificultado una difusión e implementación contundente del IEC 1131, algunos aspectos son:

Dispersión en la publicación de la norma

La norma, como se ha visto, tiene cinco partes, estas partes; tuvieron fechas de publicación diferentes y la publicación de cada parte implicaba nuevos interrogantes e incertidumbre frente a lo venidero, las fechas de publicación de las partes se podrían exponer así:

Parte 1: 1992 Información general

Parte 2: 1992 Especificación y ensayos de equipos

Parte 3: 1993 Lenguajes de programación

Parte 4: 1994 Guía de usuario

Parte 5: 1995 Comunicación.

Esta forma de publicación que extiende entre la primera parte y la ultima tres años, y que hoy por hoy sigue en desarrollo incorporando nuevas partes como lógica fuzzy ha conducido a un progresivo desgaste pues no logro dar claridad contundente y posibilidad de pruebas pilotos especificas en corto tiempo, lo cual condujo a constante teorización sin implementación.

De igual forma cuando se logra el documento completo de las primeras cinco partes, el impacto no logra generar reacciones de implementación, esto debido a que de cierta forma se conocía la intención e idea pero la profundidad había sido disgregada, así el retomar el estudio completo de

la norma tomo un espacio de tiempo suficiente para contemplar que en la globalidad existía una capacidad real de implementación.

Limitantes para la retroactividad de los desarrollos

Después de superado lo anteriormente dicho, cuando el conocimiento de la norma era lo suficiente para generar implementación, nació la pregunta sobre la propuesta del estándar para poder colocar los desarrollos ya terminados e implementados bajo la norma, pregunta cuya única respuesta era el cambio total de los dispositivos de control, esto debido a que la norma fue construida sin convenios con casas fabricantes, por el contrario, era necesario que los usuarios exigieran a las casa fabricantes que los dispositivos de control acogieran la norma, para que así en el futuro fuese posible el cambio o implementación de nuevos dispositivos de casas fabricantes.

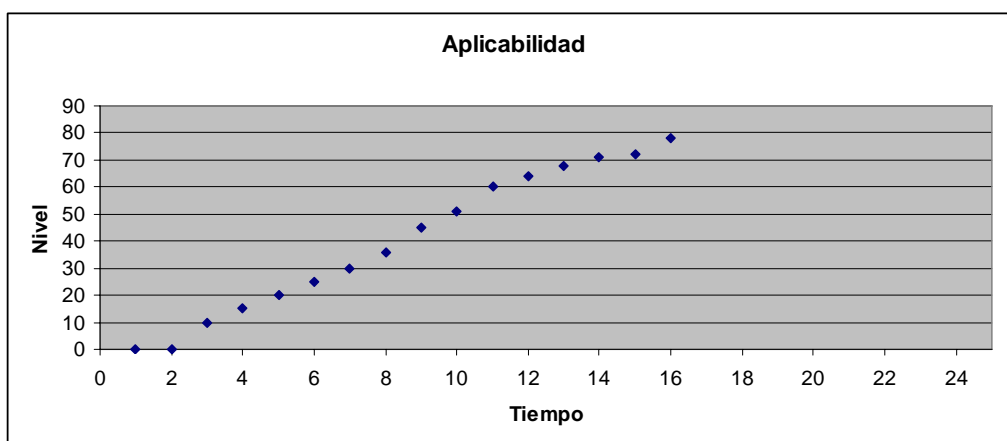
Control de mercados de los fabricantes

Esta variable para la discusión dispuesta sobre este punto, es algo que su protagonismo ha venido disminuyendo, dada la mejor comprensión por parte de los fabricantes al respecto de la norma, pues en estos tiempos ya se pueden enunciar PLC"s sobre la norma por parte de Simmens, FESTO, Telemecanique entre otros, sin embargo; en comienzo en el marco de las primeras exposiciones del estándar este factor fue muy importante para la dificultad de implementación, esto debido a que los fabricantes debían dejar libre, un dominio por tradición de control de ellos, todo por la necesidad de que los desarrollos ya no estuvieran alrededor de los dispositivos, sino alrededor del estándar y por consiguiente, con la capacidad de implementación en cualquier dispositivo que reconociera la norma y los implicantes de esta implementación a futuro, pues al ser acogida la interconexión entre diferentes dispositivos sería un paso sencillo.

Estas son algunas de las dificultades mas relevantes en la implementación de la norma, sin embargo; la aplicabilidad del estándar ha logrado superar estas dificultades y el balance se encuentra en una posición mas favorable, así se puede encontrar, como se ha expuesto en el ítem 4.3 que por parte de unas las mas representativas casas fabricantes ya existen PLC"s con la implementación de la norma, esto promovido básicamente por dos factores, Primero: el reconocimiento por parte de los usuarios de las ventajas de la norma que propiciaron y propician una presión para la implementación del estándar en los equipos de control. Segundo: Dada la presión, el estudio profundo de la norma y el reconocimiento por parte de las casas fabricantes, de las ventajas y el error que representaría quedar por fuera del circuito comunicacional que se propone en la misma.

Si se proyectara una grafica de una forma no muy rigurosa, basada en los tiempos de publicaciones al respecto de la norma, las discusiones mediante foros y revistas técnicas, la publicación de escritos y artículos al respecto, de implementación del espantar en equipos de control, y la utilización de estos equipos que recogen la norma en implementaciones reales de automatización lo que se lograría visualizar seria:

Figura 36. Grafico de aplicabilidad



Este grafico muestra como en un primer tiempo el nivel aplicabilidad era nulo, esto de forma coherente por las primeras publicaciones que apenas daban pie para la investigación y discusión, sin embargo; la evolución se nota lenta pues en los siguientes cuatro periodos el nivel de aplicabilidad solo alcanza un 20%, que para la norma es fruto de las dificultades expuestas anteriormente en este capitulo, en adelante se logra observar un progreso mas acelerado que ha permitido contar en los últimos periodos un nivel aproximado de aplicabilidad de la norma de 80%, y esto debido a que aun la norma, como se exponía anteriormente, sigue trabajando en nuevas partes lo cual con este nivel de aplicabilidad demuestra ahora una fortaleza frente a la continuidad de avances en este tema para el estándar.

Finalmente se puede establecer que el nivel de aplicabilidad es favorable, afirmación basada en la información consignada para esta investigación y que ha logrado visualizar esta proyección, por otra parte este tema generara al igual que las nuevas partes de la norma, la continuidad de análisis al respecto.

CONCLUSIONES

Con el análisis realizado en la presente investigación, se puede afirmar; que el estándar IEC 1131 posee los contenidos necesarios y suficientes, para el desarrollo sólido de una metodología de trabajo y para la solución de comunicación y evolución de los proyectos de automatización industrial, consiguiendo con esto unos impactos técnicos y económicos que redundan en mayor agilidad, menor costo, menor complejidad en el mantenimiento, entendimiento y reutilización de los desarrollos.

Dentro del campo de la automatización industrial, la norma tiene amplio desarrollo, por lo tanto el estándar para este sector es un avance en la construcción de una forma de trabajo con independencia de fabricantes y con la focalización puesta en el diseño.

La capacidad evidenciada en el lenguaje SFC para su denominación como un lenguaje de diseño, y la realización de la conexión de los lenguajes IL, ST, FBD y LD con el diseño, para la generación coherente de codificación en ellos.

TRABAJO FUTURO

Para la investigación sería un paso siguiente, avanzar en la caracterización del SFC con un lenguaje de diseño universal, incluyendo para él, el análisis de los demás bloques funcionales, funciones y programas que quedan por fuera de esta investigación por su carácter de exploración inicial, incluyendo para esto las partes 6 y 7 de la norma.

BIBLIOGRAFÍA

BALCELLS, Joseph y ROMERAL, José Luís. Autómatas programables. México. Alfaomega. 1998.

CALDERON, Jesús, ABRIL, Tilsmay y TELLO, Jesús. Iniciación en CONCEPT V2.5. Venezuela. Universidad de los Andes. 2003.

COMISIÓN INTERNACIONAL DE ELECTROTÉCNICA. IEC 61131, Suiza. IEC 1995.

DE CASTRO LOZANO, Carlos y ROMERO MORALES, Cristóbal. Introducción a SCADA. 2001.

HURTADO DE BARRERA, Jacqueline. Metodología de la Investigación Holística. Caracas, Venezuela. Fundación Sypal. 1998.

JOHN, Kart-Heinz y TIEGELKAMP, Michael. IEC 61131-3 Programming Industrial Automation Systems. New York. Springer. 2000.

MATEUS MARTIN, Felipe. Autómatas Programables: Introducción al Estándar IEC 61131. España. Universidad de Oviedo Grupo de investigación Genia. 2000.

M. MORRIS, Mano. Diseño Digital. Los Ángeles. Universidad del estado de California. Pentrice Hall. 1987.

PAPPAS, Chris H y MURRAY III, William H. Visual C++.Net, Mc Graw Hill, España. 2002.

PLCOPEN. leguajes de Programación. Departamento. Países Bajos. PLCopen de sistemas electrónicos y de control. 2004.

SIEMENS. Manual de usuario PLC S7-300. España. Siemens. 2000.

UNIVERSIDAD DE ORIENTE. Tecnologías Emergentes En Automatización. Venezuela. Facultad de Ingeniería Eléctrica. 2002.