

Nuevas tendencias en la automatización con autómatas programables basados en la Norma IEC-61131-3

Apellidos, nombre	Pineda Sánchez, Manuel (mpineda@die.upv.es)
Departamento	Departamento de ingeniería Eléctrica
Centro	Escuela Técnica Superior de Ingenieros Industriales Universidad Politécnica de Valencia

1 Resumen

Dada las necesidades actuales en las automatizaciones industriales mediante los autómatas programables, que necesitan además del propio control, de comunicarse con multitud de equipos, tanto de control como de información, así como con programas informáticos de ayuda a los controles de producción, de mantenimiento y de la calidad de los productos, las formas clásicas de programación se han quedado obsoletas.

A principio de los años 1990, se redactó una norma que ha propiciado unas nuevas tendencias de diseño de maniobras de automatización. Atendiendo a estas tendencias, y dada la experiencia del autor (que ha participado en dos proyectos europeos en este sentido) y las inquietudes de los alumnos de este tipo de tecnologías, se redacta un artículo que detalle escuetamente estas tendencias. Se ponen varios ejemplos prácticos para que el lector pueda adquirir unos conocimientos claros sobre la evolución de estas tecnologías.

2 Introducción

En la actualidad, uno de los equipos de control más utilizados a nivel industrial es el "Autómata Programable", también conocido como PLC (Programmable Logic Controller). Este tipo de equipos se empezó a implantar en la industria a finales de la década de los años 1960 y principios de los 1970, y su evolución ha sido constante y muy acelerada, comparable a la que han tenido los ordenadores personales (PCs), tanto desde el punto de vista del hardware como del software.

De hecho, su presencia en prácticamente cualquier tipo de automatización industrial es tal que ha propiciado el desarrollo de una profesión nueva, la de técnico especialista en este campo. Como contrapartida, su desarrollo ha tenido que adecuarse a la gran experiencia y modo de trabajar de los técnicos y electricistas expertos en automatización encargados de su programación y puesta en marcha. Un requisito fundamental para el despegue industrial de los PLCs ha sido la necesidad de que su programación fuera familiar a estos especialistas encargados de su implantación.

Conjugar estas características contrapuestas, la **introducción de un nuevo sistema de automatización que sea compatible con los métodos tradicionales** empleados en este campo, no ha resultado fácil, y ha supuesto un **coste adicional** al que ya representa la investigación y desarrollo de estos equipos. Por ello, el mecanismo empleado por los **grandes fabricantes** de PLC para rentabilizar las inversiones en I+D realizadas para su desarrollo, ha sido el **crear un producto propio y diferenciado, con escasa compatibilidad con productos de otros fabricantes**, de modo que en torno al PLC de una marca se crea un "ecosistema" de productos de la misma marca capaces de crear un sistema completo de automatización. Tal dependencia **conlleva un elevado coste de adquisición y mantenimiento**, pues la falta de competitividad en el mercado se traduce en una estructura de precios monolítica controlada por el fabricante, y no por el consumidor. Por tanto hasta **hoy en día**, aunque la filosofía de los PLCs, es la misma para cualquier marca o modelo, **desde el punto de vista de programación y/o aplicación, suelen ser muy diferentes y escasamente compatibles**.

No obstante, existe una enorme **presión** por parte de la industria hacia la **adopción de estándares industriales, que faciliten el intercambio de componentes y software**. Se trata de una **situación análoga al mundo del PC**, donde la existencia de sistemas operativos, protocolos de comunicaciones, etc., hace que un mismo software se pueda ejecutar sin ningún problema en PCs de diferentes marcas. **Con la finalidad de estandarizar** el campo de la automatización industrial basada en PLCs, los **organismos reguladores**, e incluso las **asociaciones de fabricantes, han desarrollado** una serie de **normas para unificar** en lo posible tanto **los** diferentes tipos de **software** como el **hardware** de los equipos. **Los contenidos a tratar en este artículo** son los **referidos a** los estándares que afectan al **software** de programación de los PLCs. Piedra angular en este campo ha sido la adopción de **la norma "IEC-61131-3"**, que inició su camino en el año de 1992, pero que ha ido modificándose desde entonces (se espera la revisión 3 para finales de 2013). Esta norma **trata de regular tanto los lenguajes de programación como la forma o estructura de programación**. Es bastante ambiciosa, pero que de momento **ninguna marca ni modelo** de autómatas programables la **cumple al 100%**, aunque sí que hay que reconocer que, **poco a poco, los modelos más recientes se acercan a este grado de cumplimiento**.

En este trabajo no se va a hacer una exposición detallada de la norma. Por el contrario lo que se pretende es que el lector entienda hacia donde se dirige este tipo de regulaciones normativas, y cómo afecta al desarrollo de los programas en nuevas automatizaciones. Para facilitar el desarrollo del trabajo, se realizará una comparativa de la evolución de la programación de PLCs con la experimentada, décadas atrás, por los ordenadores personales.

3 Objetivos

El objeto del presente trabajo es el de familiarizar a los lectores con las nuevas tendencias de automatización mediante PLCs. Para ello se plantean ejemplos de programación sencillos, con los diferentes lenguajes propuestos por la norma. Los objetivos a alcanzar empleando esta metodología son:

- Asimilar los lenguajes de programación recogidos en la norma.
- Entender las herramientas para el desarrollo de programas mediante el empleo de componentes y bloques funcionales.
- Aplicar estas herramientas al desarrollo de maniobras sencillas con dos autómatas y entornos de programación diferentes.
- Comprobar las diferencias existentes entre programas desarrollados con marcas de PLCs distintas.

Los lectores a los que van dirigido este trabajo deben de tener unos conocimientos básicos de programación de autómatas programables, ya que dada la extensión de este trabajo es impropio pensar que sirva como un método de aprendizaje. Para ello se recomienda acudir a los textos especializados de esa materia, algunos de los cuales se citan en la bibliografía asociada [1],[2].

4 Desarrollo

Los lectores de este artículo deben de conocer que los **PLCs** y los **PCs** tienen **tecnologías** de funcionamiento prácticamente **iguales**: son equipos basados en microprocesadores o microcontroladores, que se **programan** con lenguajes de **alto nivel** [3]. Sin embargo, existe una salvedad crucial: las aplicaciones que desarrollan **los PLCs no admiten fallos** que puedan poner en peligro la seguridad de las personas o de las instalaciones bajo control. Un sistema operativo nuevo puesto en un PC, si se bloquea o se "cuelga", no necesariamente debe de tener consecuencias dramáticas, sin embargo un "bloqueo" de los sistemas operativos de un PLC **puede generar paradas muy costosas en la industria o incluso la puesta en peligro de vidas humanas**. Es por ello que el ámbito de **los PCs es más propicio a la introducción de innovaciones** (con el riesgo potencial de errores en sus etapas iniciales), que en de los PLCs.

Por tanto las evoluciones en los autómatas programables siempre se producen con un retraso temporal significativo con respecto a los ordenadores personales, ya que se necesita que las posibles evoluciones se encuentren exhaustivamente probadas y optimizadas. En la exposición de este trabajo, en primer lugar, se analizan las evoluciones ya implantadas, aunque recientes y que serán, casi con toda seguridad novedosas para los lectores; en segundo lugar se referencian las nuevas tendencias que el autor espera que se incorporen en un futuro más o menos cercano.

4.1.- Lenguajes de programación según la norma IEC-61131-3.

Son cinco los lenguajes de programación recogidos en la norma, todos ellos deben de ser familiares a los alumnos, estos son:

- **"IL"**, Listado de instrucciones (Instruction List). Es el más antiguo y hoy en día poco utilizado, basado en el lenguaje ensamblador de los microcontroladores. Es el más cercano al lenguaje nativo la unidad de control, y por tanto el que proporciona el mayor grado de optimización de su funcionamiento, aunque lleva aparejado un elevado grado de dificultad, por lo que siempre ha sido bastante desconocido por los técnicos electricistas.

- **"LD"**, lógica de contactos, o lenguaje de escalera (LadDer). Es el más familiar para los técnicos electricistas, ya que está basado en los esquemas clásicos de automatización con relés y contactores. Sin embargo, por ello mismo, es poco adecuado para el desarrollo de automatizaciones complejas con un alto grado de contenido matemático, de proceso y almacenamiento de datos, o necesidades avanzadas de comunicaciones y control distribuido.

- **"FBs"**, lenguaje de bloques funcionales (Function Blocks). Su gran ventaja es que permite encapsular complejos algoritmos de control en módulos reutilizables que presentan al usuario un interfaz bien definido. Se utiliza para poder dotar a los programadores de una serie de componentes de automatización que resultarían muy complejos de generar al programador. Un ejemplo claro son los temporizadores y contadores.

.- **“ST”**, lenguaje estructurado (Structured Text), es muy parecido a los lenguajes de programación de los PCs de alto nivel, tales como el lenguaje “C” o “C++”. Es un lenguaje muy potente pero bastante complejo para el técnico electricista.

.- **“SFC”**, lenguaje gráfico de secuencia de comandos (Sequential Function Chart), basado en el lenguaje GRAFCET, que permite una formalización de procesos secuenciales mediante estructuras gráficas de control.

Los autómatas nuevos y su correspondiente software de programación, permiten usar de forma indistinta varios de los lenguajes anteriores [4]. Con ello se pretende facilitar el trabajo de desarrollo de los electricistas, aportando al mismo una gran versatilidad y potencia de automatización. Casi todos los PLCs se pueden programar en el lenguaje “LD” y el “FB”, combinación de lenguajes que ya va siendo familiar para los técnicos de desarrollo y mantenimiento.

4.2.- Estructura de programa de PLC bajo la norma IEC-61131-3 mediante el empleo de bloques funcionales.

A la hora de estructurar los programas, en la norma se propone el empleo de unidades de programación denominadas como **“POUs”** (Programming Organization Units - Unidades Organizativas de Programa). Bajo este epígrafe la norma facilita al usuario la estructuración de **grandes programas como** un conjunto de **pequeños programas** (más o menos extensos) que desarrollan una función específica, **de forma similar a los bloques funcionales** ofrecidos por el fabricante del PLC en su software de programación. Cada una de **estas unidades de programación** se puede **desarrollar** empleando cualquiera de los **lenguajes especificados en la norma**, lo que proporciona una gran flexibilidad al programador y al usuario final.

Dentro de estas POUs, se puede instanciar y utilizar cualquier “FB” del tipo temporizador o similar, de las creadas por el fabricante del autómata programable. Pero además, dentro de una POU se puede (o se debe de poder) incluir otra POU, que se haya diseñado previamente. Si el lector analiza con detenimiento esta nueva característica, entenderá que con ella se puede disponer de una **“potentísima herramienta de trabajo”**.

Esta estructura es común a los cuatro primeros lenguajes de programación descritos. No así con el SFC, que tiene una estructura propia mucho más rígida y particular, optimizada para la resolución de problemas en los que intervienen maniobras de carácter secuencial.

En cuanto a las nuevas tendencias en el desarrollo de la norma, se ahonda en el enfoque orientado a objetos, dotando a los componentes de programación de características avanzadas como la herencia, propiedades, polimorfismo, etc., heredados de los entornos de programación avanzados ya disponibles en el mundo del PC.

4.3.- Ejemplo diseñado, mediante los lenguajes tradicionales y los lenguajes recogidos en la norma IEC-61131-3.

Una vez expuestas las nuevas tendencias en la programación de los autómatas programables, se va a exponer varios ejemplos prácticos sencillos con los que se pongan de manifiesto de forma clara la importancia y potenciabilidad de las nuevas características citadas, en particular el desarrollo de programas mediante componentes funcionales [5].

Aunque como se ha dicho, este trabajo va dirigido a lectores con conocimiento de programación de PLCs, los ejemplos a tratar serán de automatizaciones cercanas a cualquier lector, de manera que al menos los controles desarrollados se puedan ser familiares desde el punto de vista de la aplicación. Unos ejemplos se realizarán con un tipo de lenguaje de programación y otros con otro lenguaje diferente, y con software de autómatas diferentes [6].

4.3.1.- Automatización del control de una luz de escalera.

El **control** de la luz de una escalera se puede plantear bajo **diversos requerimientos** [7]. Por ejemplo, si el **tiempo de encendido** se empieza a contar al pulsar o al soltar el pulsador, si la posible avería de un pulsador puede o debe de dejar la luz encendida durante todo el tiempo que dure la avería o solo durante el tiempo previsto que dure un encendido, y que no se vuelva a encender la luz hasta que se repara la avería, etc. Esta automatización es **similar al control de la luz de un parking o luz de un garaje**. En la figura 1, se recoge, como ejemplo, una representación gráfica (**cronograma**) de dos diferentes alternativas de la maniobra de control propuesta, correspondientes a la elección del inicio de la temporización. Se deja al lector analizar las diferencias entre ambas, como prueba de sus conocimientos en automatización.

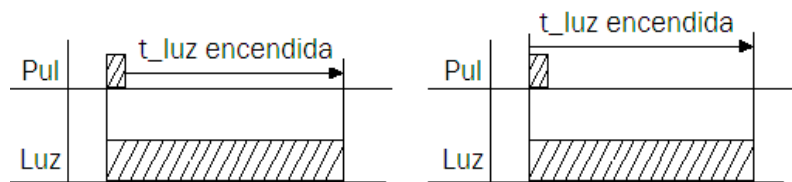


Figura 1.- Funcionamiento con dos alternativas de control de la luz de escalera.

Para resolver este ejercicio se opta por definir que el tiempo de encendido se inicie en el instante de accionar el pulsador. Con este criterio una posible maniobra o programa podría ser el de la figura 2.

El tipo de temporizador empleado en la figura 2 está recogido en la norma, y aparece en dicha norma con el nombre de temporizador tipo "TP". Observando la maniobra se puede deducir que además de llevar el control del tiempo de encendido se da como información la evolución del tiempo durante el encendido, quedando disponible esta información en una determinada memoria que se ha identificado como "ET". Desde allí se puede tomar el dato y transferirlo a un visualizador.

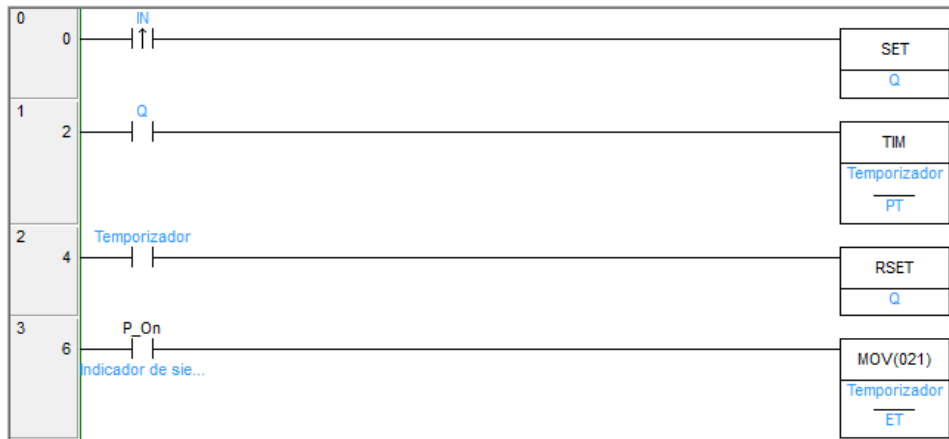


Figura 2.- Maniobra de control de una luz de escalera o garaje.

En la figura 3, se aprecia una aplicación de la función creada. Esta función se puede utilizar (asignándole diferentes nombres) cuantas veces sea necesario. Así se aprecia en la figura 4, una doble aplicación, en la cual, se añade una función más para el control de la luz del garaje del sótano 2.

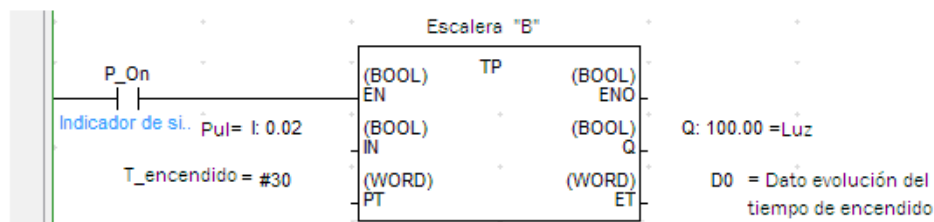


Figura 3.- Bloque funcional creado y una posible aplicación.

La realización de este ejemplo se ha efectuado mediante un autómata de la marca Omron y modelo CPL1. Para su programación se ha utilizado el software denominado "CX-Programmer", y la asignación de las variables de entrada y salida, así como la de las posiciones de memoria se ha efectuado de acuerdo con el mapa de memoria del equipo empleado.

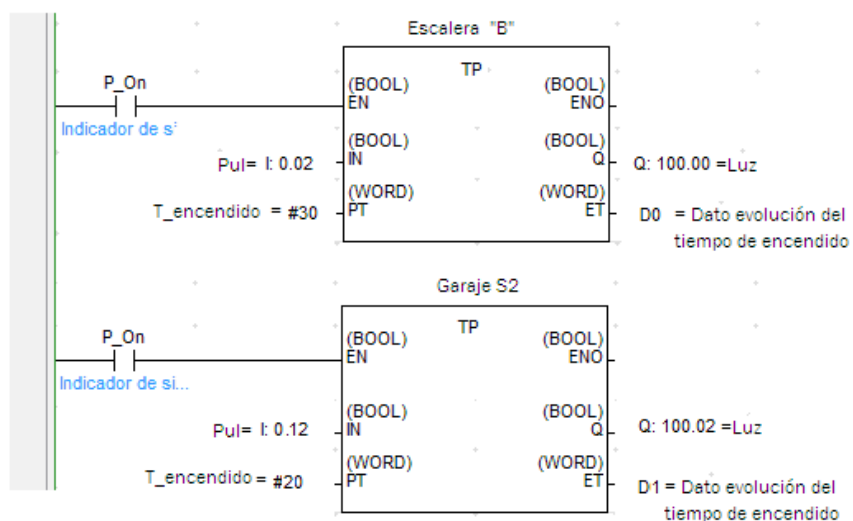


Figura 4.- Doble aplicación del bloque funcional creado.

Para obtener una información completa de la forma en la que se declaran las variables se aconseja al lector que recabe la correspondiente información en la documentación técnica de esta marca o **bien puede seguir este y otros ejemplos en la publicación de la UPV (Universitat Politècnica de Valencia), con número "444"**. No obstante la identificación de las variables y tipología es las que se recogen en la tabla1.

Tipo variable	Nombre	Tipo de dato	Retenido	AT	Valor inicial
Entradas	EN	BOOL	No		FALSE
Entradas	IN	BOOL	No		FALSE
Entradas	PT	WORD	No		0
Salidas	ENO	BOOL	No		FALSE
Salidas	Q	BOOL	No		FALSE
Salidas	ET	WORD	No		0
internos	Temporizador	TIMER	No		

Tabla 1.- Asignación de variables.

4.5.- Automatización del control de una taladradora industria.

Una aplicación industrial sencilla puede ser la automatización de una máquina de taladrar, tal como la recogida en la figura 5.

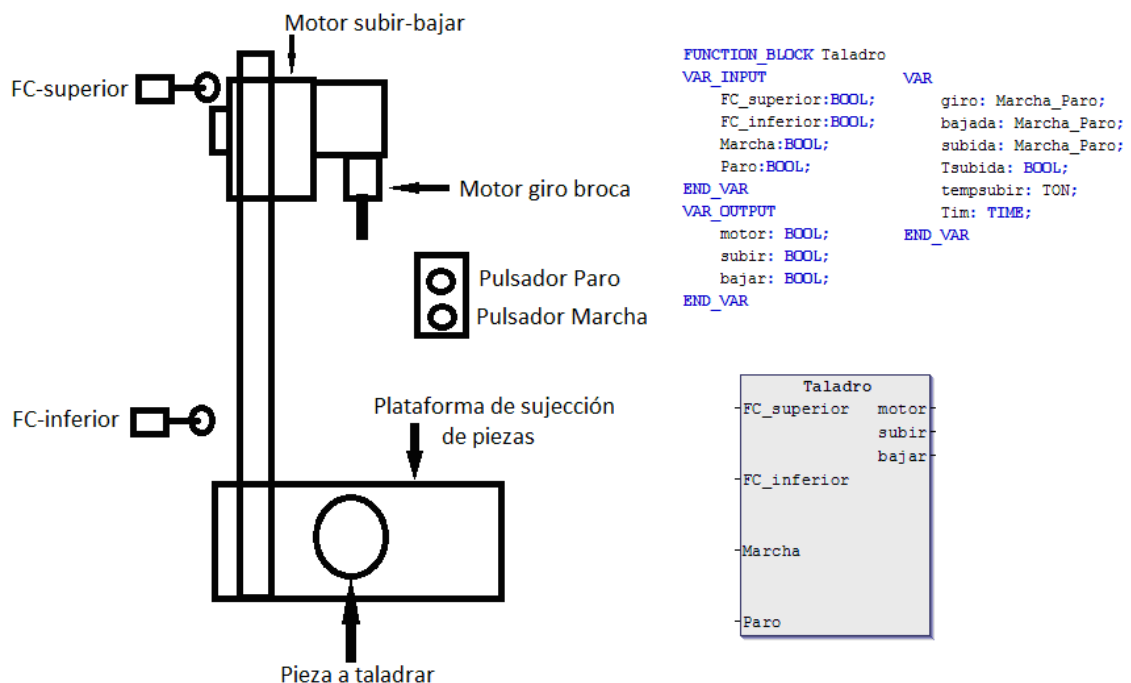


Figura 5.- Esquema de la máquina de taladrar y declaración de variables.

En este ejemplo se plantea la **automatización de una taladradora**, de un único ciclo, de forma que al accionar el pulsador de marcha la broca se pone en funcionamiento y baja, liberando el final de carrera superior. Se considera que la pieza ha sido

agujerada al llegar el portabrocas al final de carrera inferior, con lo que la máquina deja de bajar e inicia su subida, hasta que llega al final de carrera superior, donde se detiene su funcionamiento. En **la figura 5, se representa un esquema de un prototipo de esta máquina**, la declaración de las variables de entrada, de salida y las variables internas de programación (que son funciones más simples generadas con anterioridad), así como la estructura de la función creada.

Esta función se ha realizado con el software "Codesys", que es empleado por diferentes marcas de autómatas. En la figura 6, se representa la automatización completa de la maniobra.

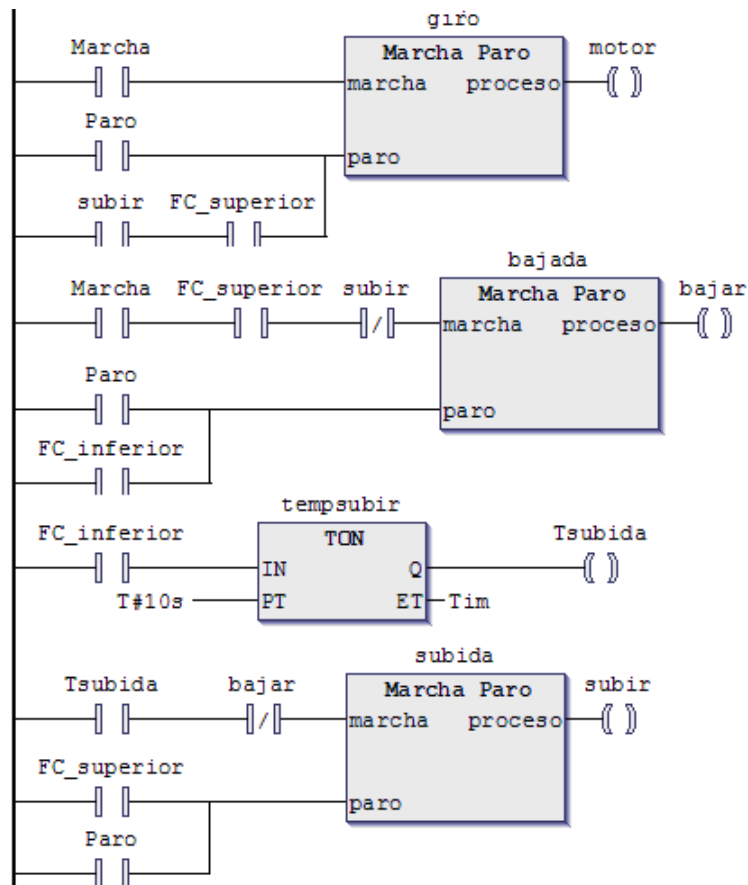


Figura 6.- Esquema de la automatización de la función taladro.

Solo añadir a lo ya comentado sobre el funcionamiento de la maniobra que se ha dispuesto un temporizador de retardo cuando llega la máquina al final de carrera inferior, para que reinicie la subida. Con ello se consigue que el cambio de sentido (inversor de giro) no origine corrientes de valor muy elevado en el motor.

5 Cierre

Este trabajo quiere dejar de manifiesto la importancia que tiene la implementación de las automatizaciones industriales construyendo base de funciones creadas por el diseñador y que se correspondan con maniobras muy sencillas y que hayan sido previamente probadas y contratada su eficacia. De esta manera maniobras

complejas o muy extensas, quedan encuadradas en pequeños diagramas de bloques (funciones con aplicaciones especiales asignadas) que facilita tanto la programación como la interpretación de la automatización.

Este tipo de tendencias, no son nuevas desde el punto de vista de la "filosofía" de diseño, pero sí desde el punto de vista de la programación, y sin duda representa el futuro inmediato en este campo, por lo que se considera de vital importancia que los alumnos/lectores interesados en estas tecnologías, se vayan familiarizando con ellas.

Se recomienda a los Ingenieros que profundicen más en este campo ya que hoy en día representa un importante banco de futuros puestos de trabajo, y además se está empezando a implementar desde este tipo de tareas la visualización directa en pantallas "SCADA" o pantallas de operador, de las máquinas y sus accionamientos.

6 Bibliografía

[1] Pérez Cruz, J., Pineda Sánchez, M. "Automatización de maniobras industriales mediante autómatas programables". Univ. Politéc. Valencia. 2006.

[2] Valentín Labarta, J.L. "Automatismos Industriales". Donostiarra S.A. 2012.

[3] Mandado Pérez, E., Marcos Acevedo, J., Fernández Silva, C. "Autómatas programables y sistemas de automatización". Marcombo S.A. 2009.

[4] Karl-Heinz, J., Tiegelkamp, M. "IEC 61131-3: Programming Industrial Automation Systems: Concepts and Programming Languages, Requirements for Programming Systems, Decision-making Aids". Springer. 2010.

[5] Romera Ramírez, J.P., Lorite, J.A., Montoro, S. "Automatización: problemas resueltos con autómatas programables". Thomson-Paraninfo, 1994.

[6] Tiegelkamp, M., Karl-Heinz, J. "IEC 61131-3: Programming Industrial Automation Systems". Springer-Verlag Berlin Heidelberg. 2010.

[7] Moro Vallina, M. "Instalaciones domóticas". Paraninfo. 2011.