

**Universidad Central “Marta Abreu” de Las Villas**

**Facultad de Ingeniería Eléctrica**

**Departamento de Automática y Sistemas Computacionales**



## **TRABAJO DE DIPLOMA**

**Identificación y Representación de las Principales  
Variables de la Etiquetadora KOSME de la Ronera  
Central “Agustín Rodríguez Mena”**

**Autor: Liodan Valdivia Coca**

**Tutor: Msc. José E. García Arteaga**

**Santa Clara**

**2007**

**"Año 49 de la Revolución"**

**Universidad Central “Marta Abreu” de Las Villas**

**Facultad de Ingeniería Eléctrica**

**Departamento de Automática y Sistemas Computacionales**



## **TRABAJO DE DIPLOMA**

### **Identificación y Representación de las Principales Variables de la Etiquetadora KOSME de la Ronera Central “Agustín Rodríguez Mena”.**

**Autor: Liodan Valdivia Coca.**

E-mail: liodan@uclv.edu.cu

**Tutor: M Sc. José E. García Arteaga**

Profesor Asistente. Dpto. de Automática.

Facultad de Ingeniería Eléctrica. UCLV.

E-mail: arteaga@uclv.edu.cu

**Consultante: Leonardo Montejo Rodríguez.**

**Vladimir Nodal Gómez.**

Santa Clara

"Año 49 de la Revolución"



Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería en Automática, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

---

Firma del Autor

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

---

Firma del Autor

---

Firma del Jefe de Departamento  
donde se defiende el trabajo

---

Firma del Responsable de  
Información Científico-Técnica

*“De frailes que le niegan a Colón  
la posibilidad de descubrir el paso Nuevo  
está lleno el mundo,  
lo que importa no es sentarse con los frailes,  
sino embarcarse en las carabelas con Colón.”*

*José Martí.*

*A los que se han propuesto y logrado sus propios triunfos.*

*A mis padres.*

## AGRADECIMIENTOS

*A mis padres, por la confianza, el amor y la energía*

*A Jeisil por estar siempre ahí*

*A los amigos que me acompañaron hasta el final*

*A mi tutor, Cheo, por el apoyo y el conocimiento*

*A Vladimir y a Montejo, por complementar mi formación*

*A todos los que de una forma u otra han tenido que ver con mis triunfos*

*A todos, Gracias.*

## TAREA TÉCNICA

Con el propósito de desarrollar el presente trabajo y finalmente confeccionar el informe final, fue necesario realizar las siguientes tareas:

- Familiarización con el proceso de producción e identificación de las variables principales en la máquina etiquetadora.
- Estudio del autómata SYSMAC CPM2A de OMRON.
- Realización de la ingeniería inversa del programa del autómata y obtención de información de las variables principales.
- Implementación de los códigos que garanticen conocer las variables que no estén disponibles en el programa y den solución a los objetivos del trabajo.
- Estudio del MOVICON X y desarrollo de una pequeña aplicación SCADA para la supervisión y el control del proceso de etiquetado.

---

Firma del Autor

---

Firma del Tutor

## **RESUMEN**

El presente trabajo se propone como objetivo identificar y representar en un SCADA las principales variables de la etiquetadora KOSME de la Ronera Central “Agustín Rodríguez Mena”. Para lograr este objetivo primeramente se realizó un estudio general de los Autómatas Programables, y específicamente del que controla dicha máquina. Una vez identificadas las variables que intervienen en el proceso, se pasó a la reprogramación del PLC, implementando el código que garantizó conocer las variables que se necesitaban mostrar en el SCADA y no estaban presentes en el código actual. Por último se realizó una pequeña aplicación SCADA con la representación de como podría quedar el mímico del proceso que se pretende supervisar y controlar. La realización de este trabajo fue necesaria porque no se contaba con una buena interconexión entre las diferentes máquinas del proceso en cuestión, no existía ningún control sobre la materia prima y el producto final por parte de la dirección de la empresa y además no estaba disponible la documentación técnica del proceso para facilitar la labor a realizar y es por ello es que se parte de la ingeniería inversa. Para desarrollar el presente trabajo se realizaron varias visitas a la industria, donde se contactó con los diferentes operadores de las máquinas, se realizó una amplia búsqueda en internet, y un estudio de los software CX-PROGRAMMER de programación de PLC y MOVICON-X de programación de SCADA.

## TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	iii
TAREA TÉCNICA .....	iv
RESUMEN .....	v
INTRODUCCIÓN .....	1
Objetivos del trabajo.....	4
Organización del informe. ....	4
CAPÍTULO 1.  ACERCAMIENTO TEORICO AL AUTÓMATA PROGRAMABLE ...	6
1.1  Necesidad del Trabajo. ....	6
1.2  Generalidades del PLC.....	8
1.2.1  Historia de lo autómatas programables. ....	8
1.2.2  Definición. ....	10
1.2.3  Ventajas e inconvenientes de su uso. ....	11
1.2.4  Estructura.....	13
1.2.5  Interfaces. ....	18
1.2.6  Equipos o unidades de programación.....	18
1.2.7  Dispositivos periféricos.....	20
1.2.8  Funcionamiento del AP. ....	20
1.2.9  Introducción a la programación de PLC.....	22

1.3	Campos de aplicación.....	26
1.3.1	Ejemplos de sus aplicaciones en la industria nacional y extranjera.....	28
CAPÍTULO 2. IDENTIFICACION DE VARIABLES Y SISTEMA SUPERVISOR ...		30
2.1	Características y funciones básicas del CPM2A.....	30
2.1.1	CPU.....	30
2.1.2	Unidades de E/S de expansión.....	31
2.1.3	Unidades de E/S analógicas.....	31
2.1.4	Contadores de alta velocidad e interrupciones.....	32
2.1.5	Resumen de sus principales características. ....	32
2.2	La ingeniería inversa. ....	33
2.2.1	Funciones empleadas en el programa del autómeta.....	33
2.2.2	Ingeniería inversa al código del programa. ....	37
2.3	Sistema supervisor. ....	39
2.3.1	LabVIEW.....	39
2.3.2	WinCC.....	40
2.3.3	MOVICON. ....	41
CAPÍTULO 3. IMPLEMENTACION DEL SISTEMA SUPERVISOR Y ANALISIS DE LOS RESULTADOS .....		45
3.1	Mapa de memoria.....	46
3.2	Propuesta de programación.....	49
3.3	Desarrollo del sistema supervisor. ....	50
3.3.1	Implementación del sistema supervisor. ....	51
3.4	Análisis económico. ....	53
CONCLUSIONES Y RECOMENDACIONES .....		59
	Conclusiones .....	59

Recomendaciones .....	60
REFERENCIAS BIBLIOGRÁFICAS .....	61
BIBLIOGRAFIA.....	63
ANEXOS .....	64
Anexo I Características del A.P SYSMAC- CPM2A de OMRON.....	64
Anexo II Comparación del A.P CPM2A con el CPM2C de OMRON. ....	65
Anexo II. Arquitectura de un PLC. ....	66
Anexo III Programa del autómeta SYSMAC-CPM2A. ....	67
Anexo IV Software de programación de PLC CX-Programmer V3.0. ....	71

## INTRODUCCIÓN

Desde el comienzo de la industrialización, el hombre ha buscado las formas y procedimientos para que los trabajos se realicen de forma más ágil y resulten menos tediosos para el propio operador. La Automatización<sup>1</sup> Industrial ha contribuido positivamente en este sentido. Ésta consiste en el uso de sistemas o elementos computarizados para controlar maquinarias y/o procesos industriales substituyendo a operadores humanos. El alcance va más allá que la simple mecanización de los procesos ya que ésta provee, a operadores humanos, mecanismos para asistirlos en los esfuerzos físicos del trabajo; la automatización reduce ampliamente la necesidad sensorial y mental del hombre. Ésta, como una disciplina de la ingeniería, es más amplia que un mero sistema de control; abarca la instrumentación industrial, que incluye los sensores y transmisores de campo, los sistemas de control y supervisión, los sistema de transmisión y recolección de datos y las aplicaciones de software en tiempo real para la supervisión y control de las operaciones de plantas o procesos industriales.

La automatización surge para integrar varios aspectos de las operaciones de manufactura, para mejorar la calidad y uniformidad del producto, minimizar el esfuerzo y los tiempos de producción, mejorar la productividad reduciendo los costos de manufactura mediante un mejor control de la producción, reducir la intervención y posibilidad de error del hombre, reducir el daño en las piezas que resultaría del manejo manual, aumentar la seguridad para el personal, y ahorrar área en la planta, haciendo más eficiente la disposición espacial de las máquinas así como el flujo de materiales.

---

<sup>1</sup> Automatización; del griego antiguo: guiado por uno mismo.

Un mecanismo que ha sido clave en dicho proceso de automatización es el Autómata Programable o PLC (*Programmable Logic Controller* en sus siglas en inglés); este dispositivo consigue entre otras muchas cosas, que ciertas tareas se hagan de forma más rápida y evita que el hombre aparezca involucrado en trabajos peligrosos para él y su entorno más próximo. Hoy en día estamos rodeados por estos mecanismos, tanto que, han rebasado la frontera de lo industrial para hacerse más cercanos: en semáforos, gestión de la iluminación de fuentes, parques, jardines, control de puertas automáticas, etc. y hasta en la propia vivienda: control de puertas y ventanas, iluminación, climatización, etc.

El desarrollo de los Controladores Lógicos Programables estaba dirigido originalmente por los requerimientos de los fabricantes de automóviles quienes cambiaban constantemente los sistemas de control en sus líneas de producción para acomodarlos a sus nuevos modelos de carros. En el pasado, esto requería un extenso re-alambrado de bancos de relevadores y contactores, un procedimiento muy costoso. Estos sistemas basados en relés tenían, además, un tiempo de vida muy limitado y se necesitaba un sistema de mantenimiento muy estricto. El alambrado de muchos relés en un sistema muy grande era muy complicado; si había una falla, la detección del error era muy tediosa y lenta. A finales de los años 60 los sistemas de control industriales basados en relés y programadores secuenciales electromecánicos empezaron a ser sustituidos por pequeños controladores lógicos programables, como resultado de las demandas de la industria automotriz, que necesitaba reducir los enormes costos implicados en el reemplazo de los sistemas de relés. Este nuevo controlador tenía que ser fácilmente programable, con una vida útil larga y ser resistente a ambientes difíciles, además de emplear técnicas de programación conocidas.

Actualmente, el campo de acción de los PLC se ha extendido enormemente y han alcanzado un grado tal de desarrollo que tienen una fuerte competencia y demanda en el mercado internacional. Los sistemas actuales de PLC pueden incluir varias CPU en configuración redundante, con cientos de señales digitales y analógicas y enormes capacidades de cálculo y control.(Diéguez, 2005)

En la actualidad existen numerosos fabricantes de Autómatas entre los que se destacan TELEMECANIQUE, OMRON, SIEMENS, TOSHIBA, entre otros. En Cuba existen empresas comercializadoras de los productos de estas firmas como es el caso de OMRON.

Dada las facilidades que le brinda este fabricante a nuestro país y las prestaciones del propio producto de esta marca, muchas de nuestras industrias automatizadas emplean Automatas Programables de esta firma. Un ejemplo de ello se muestra en la Ronera Central “Agustín Rodríguez Mena”, perteneciente a la corporación Cuba Ron S.A, ubicada en Central Agro-Industrial (CAI) George Washington, del municipio Santo Domingo, provincia Villa Clara.

Esta empresa productora de rones, se encarga de la fabricación de productos de este tipo con una buena calidad, debido al esfuerzo de todos sus trabajadores y al grado de automatización con que cuenta la misma. A finales del año 2006 la dirección de la fábrica, decidió aumentar la calidad de sus productos, tener un mayor control sobre toda la materia prima y sobre el propio producto terminado, hacer más cómodo y menos riesgoso el trabajo para los operarios, reducir los costos de producción, adaptarse rápidamente a las nuevas orientaciones que se produzcan en el mercado, e incrementar la productividad. Para darle solución a este problema se decidió implementar un sistema SCADA con el empleo del software MOVICON X, para supervisar y controlar la línea de embotellado (Figura 1). Para el montaje de este sistema es necesario recolectar las variables del proceso, muchas de las cuales están siendo adquiridas y posteriormente almacenadas, en tiempo real, en las memorias de los distintos PLC's que forman parte de los lazos de control de los diferentes automatismos implementados en esta línea de producción.

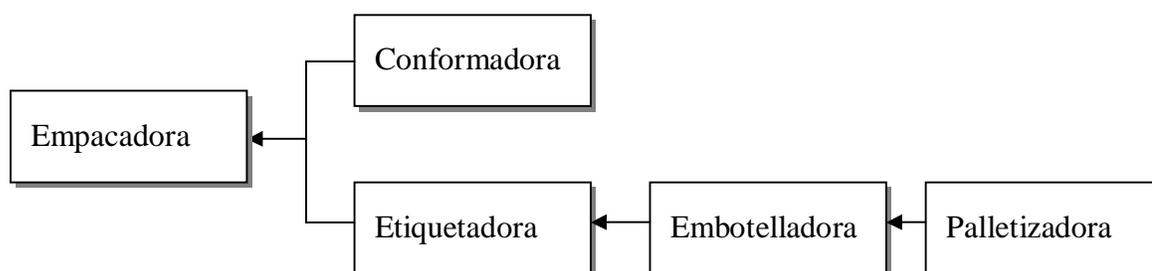


Figura 1. Línea de embotellado.

La corporación “Cuba Ron” no consta con la documentación técnica de esta línea de producción, donde se registran los datos específicos del proyecto, tales como los planos eléctricos, programas de los autómatas que controlan las máquinas y demás especificaciones necesarias para conocer a fondo el proceso de embotellado, factores estos que son indispensables para la puesta en marcha del sistema SCADA que se desea

implementar. Esta línea de embotellado fue implementada por una compañía extranjera, que al retirarse de nuestro país no dejó disponible dicha documentación.

### **Objetivos del trabajo.**

Por todo lo antes mencionado, este trabajo tendrá como objetivos los siguientes:

#### **ØObjetivo general:**

- Identificación y representación de las principales variables de la etiquetadora KOSME de la Ronera Central “Agustín Rodríguez Mena”.

#### **ØObjetivo específico:**

- Mostrar los aspectos teóricos fundamentales para el trabajo con autómatas programables.
- Hacer una búsqueda sobre la tendencia del empleo del PLC en la industria.
- Realizar la ingeniería inversa del autómata programable SYSMAC-CPM2A, de OMRON.
- Reprogramar el autómata añadiendo el código que garantice conocer las variables deseadas y que no estén presentes en el programa actual.
- Estudiar el software de programación de SCADA’s MOVICON-X.
- Desarrollar una pequeña aplicación SCADA empleando MOVICON-X.
- Analizar los resultados obtenidos.
- Redactar el informe final.

### **Organización del informe.**

El presente proyecto constará de tres capítulos; el primero se dedicará a exponer los fundamentos teóricos necesarios para la comprensión de los temas que serán tratados en el desarrollo del trabajo, tales como los componentes principales de un PLC, su funcionamiento, evolución y tendencias, estructura, así como una breve introducción a la programación de los mismos. Además se realizará un estudio del software de programación SCADA’s MOVICON-X.

En el capítulo dos se dará cumplimiento al principal objetivo del proyecto; se realizará la ingeniería inversa al programa del autómeta y para ello previamente se hará un estudio de de las funciones empleadas en el mismo, así como de las principales características del PLC empleado en el proceso de producción en cuestión, para luego comprender el código del programa y poder conformar el “mapa de memoria”<sup>2</sup> con el cual se logrará una mejor implementación del sistema supervisor que se montará en la línea de embotellado.

En el capítulo tres se mostrarán los resultados obtenidos del trabajo desarrollado en el capítulo dos; se dará a conocer la distribución de las direcciones internas y de E/S de las variables empleadas, se mostrará la propuesta de reprogramación del código del programa para garantizar conocer las variables que no fue posibles encontrar en el programa actual y se hará un análisis económico del montaje del sistema supervisor donde se empleará el mapa de memoria obtenido.

---

<sup>2</sup> Organización de las distintas direcciones internas y de E/S del autómeta con su correspondiente ubicación y funcionamiento en la memoria del mismo.

## **CAPÍTULO 1. ACERCAMIENTO TEORICO AL AUTÓMATA PROGRAMABLE**

En el presente capítulo se hará un recorrido por todo el universo del PLC, incluyendo una breve historia desde su surgimiento hasta la actualidad, definiciones básicas, campos de aplicación, ventajas e inconvenientes de su uso, estructura interna y externa, funcionamiento, introducción a su programación, así como las características propias del SYSMAC - CPM2A, de la etiquetadora Kosme en la Ronera Central. Además se mostrará la necesidad del desarrollo de este Trabajo de Diploma, así como otros aspectos teóricos relacionados con el empleo y aplicación de los Autómatas Programables en los procesos de automatización, tanto en la industria nacional como extranjera.

### **1.1 Necesidad del Trabajo.**

Como se había mencionado con anterioridad, en la Ronera Central, donde se producen distintos tipos de rones para la distribución a la población y la exportación, no se consta con una buena interconectividad entre las máquinas que intervienen en el proceso de embotellado, además de que no se tiene un control riguroso sobre la producción como se necesita en una empresa de este tipo. Por parte de la dirección de la empresa, se decidió aumentar la calidad de sus productos, tener un mayor control sobre toda la materia prima que se emplea en la producción del ron como producto final, como es el caso de las botellas, el juego de etiquetas (etiqueta, contraetiqueta, collarín y sello de calidad), tapas y de forma general sobre el propio producto terminado, como un esfuerzo más en la lucha contra las ilegalidades; hacer más cómodo y menos riesgoso el trabajo para los operarios, reducir los costos de producción, adaptarse rápidamente a las nuevas orientaciones del mercado, y de forma general incrementar la productividad. Con el empleo del software

MOVICON X se implementará un sistema SCADA para supervisar y controlar dicha línea, tener registros históricos de las variables; conteo de etiquetas, contraetiquetas, sellos y botellas que entran y salen de línea, detección de elementos defectuosos, y generación alarmas, etc., dándole solución a la problemática antes mencionada. Para el montaje de este sistema es necesario recolectar las variables deseadas del proceso, muchas de las cuales están siendo usadas por los PLC's que controlan la línea y se encuentran almacenadas en sus respectivas memorias.

Este sistema automatizado fue montado por una corporación italiana en el año 2004, la cuál al retirarse de nuestro país no dejó disponible la documentación técnica donde se registran los datos específicos del proyecto, tales como planos eléctricos, programas de los autómatas y las tablas de memorias de los mismos, factores estos que son indispensables para la puesta en marcha del sistema supervisor que se desea implementar. Por todo lo antes mencionado, este trabajo tendrá como objetivo específico principal la realización de la ingeniería inversa al autómata programable SYSMAC-CPM2A, de OMRON, el cual se encarga de controlar el automatismo de la máquina etiquetadora ubicada en la línea de embotellado, de la empresa antes mencionada; además mostrará un pequeño acercamiento al software de programación de SCADA's MOVICON-X, como una herramienta de última tecnología en programación de este tipo de sistemas. Con la realización de este proyecto se podrá conocer la localización de memoria de las distintas variables que intervienen en el proceso de etiquetado de botellas, para su posterior implementación en el SCADA, que se desarrollará con el software MOVICON-X, al cual se le dedicará un epígrafe en este trabajo.

De forma más objetiva se desea obtener del código del programa del autómata las siguientes especificaciones:

- § Número de etiquetas, contraetiquetas y collarines que se ponen correctamente.
- § Número de botellas que entran en la línea.
- § Número de botellas que salen sin etiqueta, contraetiquetas y/o collarín.
- § Estado de la máquina (Parada / En marcha).

## 1.2 Generalidades del PLC.

### 1.2.1 Historia de lo autómatas programables.

Los PLC's se introdujeron por primera vez en la industria en la década del 60 y hoy en día continúa su desenfrenado desarrollo. La razón principal del nacimiento de los autómatas fue la necesidad de eliminar el alto costo que se producía al reemplazar el complejo sistema de control basado en relés y contactores que caracterizaban los procesos industriales hasta ese momento.

El problema de los relés era que cuando los requerimientos de producción cambiaban también lo hacía el sistema de control. Esto comenzó a resultar bastante caro cuando los cambios fueron frecuentes. Dado que los relés son dispositivos mecánicos y poseen una vida limitada, se requería un mantenimiento planificado. Por otra parte, a veces se debían realizar conexiones entre cientos o miles de relés, lo que implicaba un enorme esfuerzo de diseño y mantenimiento. Los nuevos controladores debían ser fácilmente programables por ingenieros de planta o personal de mantenimiento. El tiempo de vida debía ser largo y los cambios en el programa tenían que realizarse de forma sencilla. Finalmente se imponía que trabajaran sin problemas en entornos industriales adversos. La solución fue el empleo de una técnica de programación familiar y reemplazar los relés mecánicos por relés de estado sólido.

#### ØLa década de 1960.

A finales de los años 60, la industria demandaba cada vez más un sistema de control económico, robusto, flexible y fácilmente modificable. En 1968 nacieron los primeros autómatas programables (APIs o PLC's). La compañía *Bedford Associates* propuso algo denominado Controlador Digital Modular (MODICON, *MOdular Digital CONtroller*) a una empresa automotriz. Otras compañías propusieron esquemas basados en ordenador, uno de los cuales estaba basado en el PDP-8<sup>3</sup>. El MODICON 084 resultó ser el primer PLC del mundo en ser producido comercialmente.(Canto, 2006)

---

<sup>3</sup> PDP-8. Primera minicomputadora, presentada el 22 de Marzo de 1965 por *Digital Equipment Corporation*.

### ØLa década de 1970.

A mediados de los 70 las tecnologías dominantes de los PLC eran máquinas de estado secuencial y CPU basadas en desplazamiento de bit. Los AMD 2901 y 2903 fueron muy populares. Los microprocesadores convencionales suministraron la potencia necesaria para resolver de forma rápida y completa la lógica de los pequeños PLC's. Por cada modelo de microprocesador había un modelo de PLC basado en el mismo. No obstante, el 2903 fue de los más utilizados.

La capacidad de comunicación comenzó a aparecer en 1973 aproximadamente. El primer sistema fue el bus Modicon (Modbus). El PLC podía ahora dialogar con otros PLC y ubicarse alejado de las máquinas que controlaba. También se introdujeron en el mundo analógico al poder enviar y recibir señales de tensión. Desafortunadamente, la falta de un estándar, acompañado de un continuo cambio tecnológico, ha hecho que la comunicación de PLC sea un cúmulo inconexo de sistemas físicos y protocolos incompatibles entre sí.

### ØLa década de 1980.

En los años 80 se produjo un intento de estandarización de las comunicaciones con el protocolo MAP (*Manufacturing Automation Protocol*) de General Motor's. También se redujeron las dimensiones del PLC y se pasó a programar con lenguajes simbólicos a través de ordenadores personales en vez de los clásicos terminales de programación (Ver figura 1.4).

En la década del 80 se mejoran sus prestaciones, es decir, velocidad de respuesta, reducción de las dimensiones, mayor concentración de número de entradas/salidas en los módulos respectivos, desarrollo de módulos de control continuo, PID, servocontroladores, y control inteligente, fuzzy, etc.

### ØLa década de 1990.

Los años 90 mostraron una gradual reducción en el número de nuevos protocolos de comunicación, y en la modernización de las capas físicas de los protocolos más populares que sobrevivieron a los 80. El estándar IEC 1131-3 intenta unificar el sistema de programación de todos los PLC en un único estándar internacional. Ahora disponemos de PLC's que pueden ser programados en diagramas de bloques, lista de instrucciones, C y

diagrama de contacto al mismo tiempo. Las PC están comenzando a reemplazar al PLC en algunas aplicaciones, incluso la compañía que introdujo el Modicon 084 ha cambiado al control basado en PC.

### ØEn la actualidad.

Debido al desarrollo de la electrónica, hoy en día hay distintas variedades de autómatas que van desde Microautómatas y Nanoautómatas que se utilizan en apertura y cierre de puertas, domótica, control de iluminación, control de riego de jardines, etc. hasta autómatas de gama alta con prestaciones de un pequeño ordenador.

La principal virtud de un PLC es su robustez y facilidad de interconexión con el proceso. La tendencia actual es dotarlo de funciones específicas de control y de canales de comunicación para que puedan conectarse entre sí y con ordenadores en red.

### 1.2.2 Definición.

A lo largo de la historia se han dado varios conceptos de autómata programable, ya sea por varios fabricantes al darle publicidad a sus productos o por estudiosos del tema en cuestión. A continuación se mostrará algunos de estos conceptos:

“Un autómata programable es una microcomputadora de propósitos específicos, especialmente diseñada para el control industrial y adquisición de datos donde se mezclan elementos de computo (lenguajes de programación, interfaces hombre-máquina, unidades de memoria, etc.) y elementos de control (regulación PID, procesamiento analógico, señalización, mando, temporización y conteo, etc.) integrados en un mismo equipo, para cumplir las funciones de automatización de procesos en tiempo real con elevada fiabilidad y seguridad.”(Castellanos, 2007)

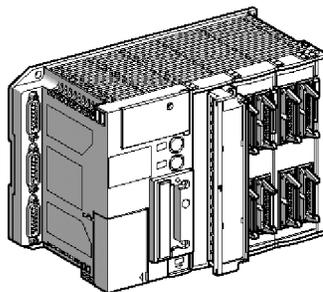


Figura 1.1. Estructura externa de un PLC.

Importantes centros de investigación como es el caso del Departamento de Electrónica de la Universidad de Oviedo en España, han dado su propia definición de lo que es un autómata programable; ellos plantean lo siguiente:

“Un autómata programable (AP) es un sistema electrónico programable diseñado para ser utilizado en un entorno industrial, que utiliza una memoria programable para el almacenamiento interno de instrucciones orientadas al usuario, para implantar unas soluciones específicas tales como funciones lógicas, secuencia, temporización, recuento y funciones aritméticas con el fin de controlar mediante entradas y salidas, digitales y analógicas diversos tipos de máquinas o procesos.”(Bayón, 2005)

De forma general podemos decir que se trata de un sistema con un hardware estándar, con capacidad de conexión directa a las señales de campo (niveles de tensión y corriente de transductores y otros periféricos electrónicos) y programable por el usuario, diseñada para controlar, en tiempo real y en medio industrial, procesos secuenciales.

### **1.2.3 Ventajas e inconvenientes de su uso.**

Un autómata programable suele emplearse en procesos industriales que tengan una o varias de las siguientes necesidades:

- Espacio reducido.
- Utilización en ambientes exigentes o agresivos.
- Procesos de producción periódicamente cambiantes.
- Procesos secuenciales.
- Maquinaria de procesos variables.
- Instalaciones de procesos complejos y amplios.
- Chequeo de programación centralizada de las partes del proceso.

Aplicaciones generales:

- Maniobra de máquinas.
- Maniobra de instalaciones.
- Señalización y control.

Esto se refiere a los autómatas programables industriales, dejando de lado los pequeños autómatas para uso más personal (que se pueden emplear, incluso, para automatizar procesos en el hogar, como la puerta de un garaje o las luces de la casa, etc.).(Bayón, 2005)

Actualmente, el campo de acción de los PLC se ha extendido enormemente y ha alcanzado un grado tal de desarrollo que compiten con los DCS (Sistemas de control distribuido) incluso en el área de procesos. Los sistemas actuales de PLC pueden incluir varias CPU en configuración redundante, con cientos de señales digitales y analógicas y enormes capacidades de cálculo y control. Esto ha conducido al desarrollo de sistemas híbridos, como Siemens PCS7 y Rockwell Process Logix, que proporcionan los beneficios del DCS y el PLC a un precio asequible y se emplean en proyectos tipo PLC que demandan capacidades de control adicionales.(Diéguez, 2005)

### ØVentajas:

Entre las principales ventajas del empleo de autómatas podemos citar:

- Menor tiempo de elaboración de proyectos.
- Posibilidad de añadir modificaciones sin costo añadido en otros componentes.
- Mínimo espacio de ocupación.
- Menor costo de mano de obra.
- Mantenimiento económico.
- Posibilidad de gobernar varias máquinas con el mismo autómata.
- Menor tiempo de puesta en funcionamiento.
- Fiabilidad tanto de hardware como de software. (los autómatas no se cuelgan).
- Homologación.

La homologación es garantía, seguridad, fiabilidad, es indicador de que el fabricante cumple con las normas.

- Flexibilidad.

§ Fáciles y rápidos de programar.

§ Aplicaciones específicas (control de temperatura por ejemplo).

§ Se le adicionan módulos para prestaciones medias como pueden ser módulos analógicos, módulos de temperatura, etc.

§ Instrucciones de programa que incluyen operaciones básicas.

§ Se configuran a partir de módulos más complejos

§ Gran número y diversidad de E/S.

§ Lenguajes muy potentes y fáciles de comprender.

- Si el autómatas queda pequeño para el proceso industrial puede seguir siendo de utilidad en otras máquinas o sistemas de producción.

**ØInconvenientes:**

Entre los principales inconvenientes del uso de autómatas programable podemos citar las siguientes:

- Adiestramiento de técnicos.
- Costo inicial.

Hoy en día los inconvenientes se han minimizado ya que la formación previa del personal suele incluir la automatización como una de sus asignaturas. En cuanto al costo tampoco hay problema, ya que hay autómatas para todas las necesidades y a precios ajustados (encontramos desde pequeños autómatas por unos 80€ hasta PLC's que alcanzan cifras exorbitantes).(Castellanos, 2007)

#### **1.2.4 Estructura.**

**ØEstructura externa:**

En cuanto a su estructura, todos los autómatas programables se clasifican en:

- Compactos: en un solo bloque están todos los elementos, es decir, la alimentación, entradas, salidas y/o la CPU. Se expanden conectándose a otros bloques con parecidas características.
- Modulares: separan por unidades las distintas partes operativas, o sea, por módulos o tarjetas adosadas en racks con funciones definidas: CPU, fuente de alimentación, módulos de E/S, etc. La conexión se realiza mediante conexión entre racks.

Además se pueden distinguir como:

- Estructura americana: separa las E/S del resto del autómatas.
- Estructura europea: cada módulo es una función (fuente de alimentación, CPU, E/S, etc.).

Exteriormente nos encontraremos con cajas que contienen una de estas estructuras, las cuales poseen indicadores y conectores en función del modelo y fabricante.

Para el caso de una estructura modular se dispone de la posibilidad de fijar los distintos módulos en raíles (o racks) normalizados, para que el conjunto sea compacto y resistente. Los Microautomatas pueden encontrarse sin caja, en formato kit, ya que su empleo no es determinado y se suele incluir dentro de un conjunto más grande de control o dentro de la misma maquinaria que se debe controlar.

### ØEstructura Interna:

Los elementos esenciales, que todo autómata programable posee como mínimo, son:

- Sección de entradas: se trata de líneas de entrada, las cuales pueden ser de tipo digital o analógico. En ambos casos tenemos unos rangos de tensión característicos, los cuales se encuentran en las hojas de características del fabricante. A estas líneas conectaremos los sensores.
- Sección de salidas: son una serie de líneas de salida, que también pueden ser de carácter digital o analógico. A estas líneas conectaremos los actuadores.
- Unidad central de proceso (CPU): se encarga de procesar el programa de usuario que le introduciremos. Para ello disponemos de diversas zonas de memoria, registros, e instrucciones de programa. Adicionalmente, en determinados modelos más avanzados, podemos disponer de funciones ya integradas en la CPU, como reguladores PID, control de posición, etc.
- Memoria: Es la encargada de contener el programa de usuario y de trabajo. En los autómatas está separada en diversas áreas según su función o datos que debe contener. Las imágenes de salidas y entradas se hacen en memoria.

Tanto las entradas como las salidas están aisladas de la CPU según el tipo de autómata que utilicemos. Normalmente se suelen emplear optoacopladores<sup>4</sup> en las entradas y relés/optoacopladores en las salidas.

---

<sup>4</sup> Optoacoplador: Dispositivo que se compone de un diodo LED y un fototransistor, de manera de que cuando el diodo LED emite luz, ilumine el fototransistor y conduzca. Su objetivo es garantizar aislamiento eléctrico.

Aparte de estos elementos podemos disponer de los siguientes:

- Unidad de alimentación (algunas CPU la llevan incluida).
- Unidad o consola de programación: que nos permitirá introducir, modificar y supervisar el programa de usuario.
- Dispositivos periféricos: como nuevas unidades de E/S, más memoria, unidades de comunicación en red, etc.
- Interfaces: facilitan la comunicación del autómata mediante enlace serie con otros dispositivos (como un PC).

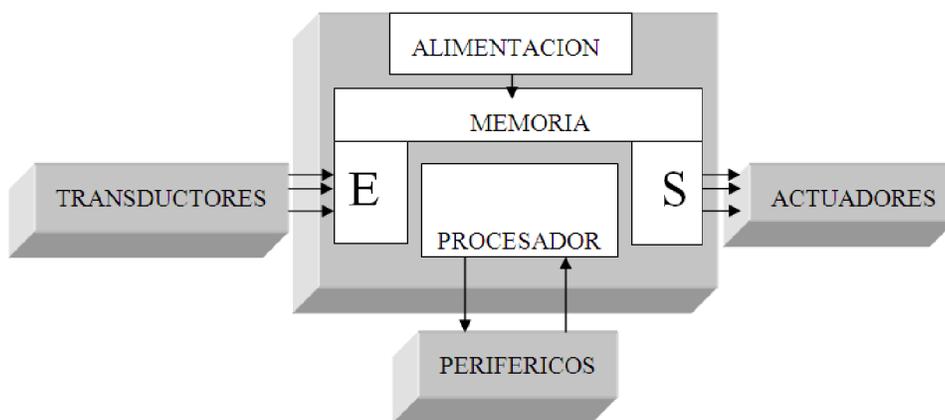


Figura 1.2. Estructura básica de un autómata.

Todos estos puntos se catalogan de forma general como periféricos, pues son dispositivos que realizan tareas complementarias al funcionamiento del autómata y están en constante comunicación con éste. Se usan tanto para programar como para visualizar el estado del mismo.

### Ø Memoria.

Vamos a disponer de un área de memoria, la cual emplearemos para diversas funciones:

---

- Memoria del programa de usuario: aquí introduciremos el programa que el autómata va a ejecutar cíclicamente.
- Memoria interna: contiene datos intermedios de los cálculos realizados así como variables internas y una imagen de las entradas y salidas.
- Memoria de datos: se suele subdividir en zonas según el tipo de datos (como temporizadores, contadores, etc.).
- Memoria del sistema: aquí se encuentra el programa en código máquina que monitoriza el sistema (programa del sistema o firmware). Este programa es ejecutado directamente por el microprocesador/microcontrolador que posea el autómata. (No accesible para el usuario)
- Memoria auxiliar: se trata de memoria externa que empleamos para almacenar el programa de usuario y en ciertos casos, parte de la memoria de la tabla de datos.

La memoria puede ser accesible bit a bit o en palabras de 8 o 16 bits. Cada autómata divide su memoria de esta forma genérica, haciendo subdivisiones específicas según el modelo y fabricante como, por ejemplo, la zona de memoria de datos de 16 bits (DM) o la zona de memoria no volátil accesible a nivel de bit (HR) en los autómatas de OMRON.

### Ø CPU.

La unidad central de procesamiento (CPU) o microprocesador, es la unidad de toma de decisiones en el PLC. La CPU toma decisiones basadas en las instrucciones sostenidas en su memoria de programa, realiza conmutación, conteo, temporización comparación de datos y operaciones secuenciales. Es el corazón del autómata programable.

Sus funciones son:

- Vigilar que el tiempo de ejecución del programa de usuario no exceda un determinado tiempo máximo (tiempo de ciclo máximo). A esta función se le suele denominar Watchdog.
- Ejecutar el programa de usuario.
- Actualizar los contadores y temporizadores programados.

- Crear una imagen de las entradas, ya que el programa de usuario no accede directamente a dichas entradas.
- Renovar el estado de las salidas en función de la imagen de las mismas obtenida al final del ciclo de ejecución del programa de usuario.
- Chequeo del sistema.

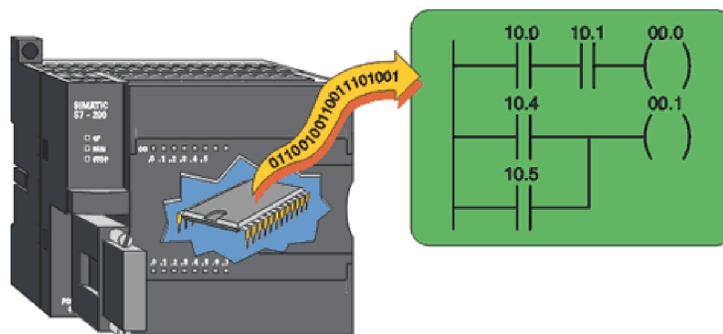


Figura 1.3. CPU de un PLC.

### Ø Unidades de E/S.

Generalmente vamos a disponer de dos tipos de E/S:

- Digital.
- Analógica.

Las E/S digitales se basan en el principio de todo o nada. Estas E/S se manejan a nivel de bit dentro del programa de usuario. Pueden ser de tipo relé o de tipo diodo. Las E/S analógicas pueden poseer cualquier valor dentro de un rango determinado especificado por el fabricante. Se basan en convertidores A/D y D/A aislados de la CPU (ópticamente o por etapa de potencia). Estas señales se manejan a nivel de byte o palabra (8/16 bits) dentro del programa de usuario.

Las E/S son leídas y escritas dependiendo del modelo y del fabricante, es decir, pueden estar incluidas sus imágenes dentro del área de memoria o ser manejadas a través de instrucciones específicas de E/S.

### Ø El bus interno.

Se conoce como bus interno al conjunto de líneas y conexiones que permiten la unión eléctrica entre la unidad de control, las memorias y las interfaces de entrada y salida. Un bus se compone de un conjunto de líneas utilizadas para intercambiar datos u órdenes (por ejemplo el contenido de celdas de memoria o las instrucciones de la unidad de control). Permite minimizar el número de conexiones entre subsistemas y el acceso de los dispositivos al mismo es controlado por la unidad de control.

Los tres buses característicos de un sistema digital son:

- bus de datos, por el que tienen lugar las transferencias de datos del sistema.
- bus de direcciones, a través del cual se direccionan la memoria y el resto de los periféricos.
- bus de control, constituido por todas las conexiones destinadas a gobernar los intercambios de información.

El número de líneas de este bus interno depende de cada fabricante. Se considera también como bus del autómatas cualquier conexión entre bloques o módulos que no necesite de procesadores específicos de comunicaciones, en sus extremos, como, por ejemplo, el cable de conexión entre el autómatas y una unidad externa de expansión de E/S.

#### 1.2.5 Interfaces.

Todo autómatas, salvo casos excepcionales, posee la virtud de poder comunicarse con otros dispositivos (como una PC o un MODEM). Lo normal es que posea una E/S serie del tipo RS-232 / RS-422. A través de esta línea se pueden manejar todas las características internas del autómatas, incluido la programación del mismo y suele emplearse para monitorización del proceso en otro lugar separado.

#### 1.2.6 Equipos o unidades de programación.

El autómatas debe disponer de alguna forma de programación, la cual se suele realizar empleando alguno de los siguientes elementos:

- Unidad de programación: suele ser en forma de calculadora. Es la manera más simple de programar el autómatas, y se suele reservar para pequeñas modificaciones

del programa o la lectura de datos en el lugar de colocación del autómata. (Figura 1.4)

- Consola de programación: es un terminal a modo de ordenador que proporciona una forma más cómoda de realizar el programa de usuario y observar parámetros internos del autómata.



Figura 1.4. Terminal de programación portátil.

- PC: es el modo más potente y a su vez más empleado en la actualidad. Permite programar desde un ordenador personal estándar, con todo lo que ello supone: herramientas más potentes, posibilidad de almacenamiento en soporte magnético, impresión, transferencia de datos, monitorización mediante software SCADA, etc.(Figura 1.5)

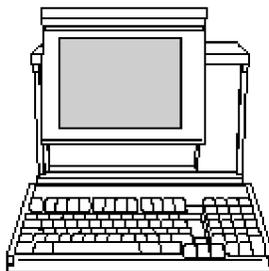


Figura 1.5. Terminal de programación compatible PC.

Para cada caso el fabricante proporciona lo necesario, bien el equipo o el software/cables adecuados. Cada equipo, dependiendo del modelo y fabricante, puede poseer una conexión a uno o varios de los elementos anteriores. En el caso de los micro-PLC se escoge la programación por PC o por unidad de programación integrada en la propia CPU.

### **1.2.7 Dispositivos periféricos.**

El autómata programable, en la mayoría de los casos, puede ser ampliable. Las ampliaciones abarcan un gran abanico de posibilidades, que van desde las redes internas (LAN, buses de campo como Profibus, CAN-Bus, etc.), módulos auxiliares de E/S, memoria adicional, hasta la conexión con otros autómatas del mismo modelo. Cada fabricante facilita las posibilidades de ampliación de sus modelos, los cuales pueden variar incluso entre modelos de la misma serie.

### **1.2.8 Funcionamiento del AP.**

Los autómatas programables son máquinas de funcionamiento secuencial que ejecutan las instrucciones de programa que se les introduce, una detrás de otra y continuamente mientras el autómata está funcionando. Utilizan, en la ejecución del programa, las señales de entrada al PLC y generan unas señales de salida para el control de la planta.

El programa y una serie de acciones comunes de funcionamiento del autómata como por ejemplo vigilar que el tiempo de ejecución del programa de usuario no excede un determinado tiempo máximo (tiempo de ciclo máximo), actualizar los contadores y temporizadores programados, etc., se ejecutan de manera periódica y cíclica en lo que se suele denominar el ciclo de “scan” o ciclo de operación. La secuencia de operación suele ser la siguiente:

- Lectura de señales de entrada desde interfaces de entrada.
- Escritura de señales de salida a interfaces de salida.
- Procesado del programa.

Las señales de entrada a través de los interfaces de entrada son copiadas a una memoria intermedia de manera que el programa que se ejecuta no accede nunca directamente a las entradas. Se crea una imagen de la entrada. Lo mismo se hace con las salidas pero esta vez la CPU escribirá la memoria con las salidas (imagen de la salida) en la salida o interfaz de salida. Estas dos acciones se ejecutan de una sola vez para ahorrar tiempo y ser más eficiente. Además, un autómata tiene varios modos de funcionamiento como son:

**RUN:** El autómata ejecuta el programa de usuario y hace funcionar contadores y temporizadores evolucionando normalmente.

STOP: El autómata está parado y listo para ser programado o para labores de mantenimiento.

ERROR: Ante una situación de mal funcionamiento del aparato, éste se detiene y bloquea a la espera de atención por el programador o salida del error que causó la detención.

Algunos autómatas tienen otro modo denominado PROGRAM. La diferencia entre RUN y PROGRAM es que en el primero ningún dispositivo de programación puede forzar o modificar ning

una posición de memoria en el PLC y en el segundo sí. El paso entre estos modos de funcionamiento se hace desde consolas o botones. El modo actual suele estar indicado mediante LEDs en la carcasa del PLC.

Ø Fase de procesos comunes:

En esta fase se comprueba periódicamente que el programa de usuario no tarde más de un tiempo en ejecutarse (watchdog), que no existan errores de conexiones y que no existan errores de sintaxis del programa. El watchdog es un reloj interno no accesible por el usuario (salvo para ser inicializado o programar su tiempo de cuenta) que genera una interrupción al transcurrir el tiempo establecido si no es re-inicializado. Por tanto, comprueba que el programa alcanza la instrucción END antes de un tiempo dado, y que no se ha producido una detención por un bucle infinito u otras causas. La comprobación de conexiones comprueba los niveles de tensión en las conexiones y en la pila, si existe, y en los buses. En la comprobación del programa se revisa si el mismo está correctamente escrito y si se mantienen los datos.

Ø Fase de ejecución de programa:

En esta fase se atienden las entradas leyendo su contenido y se actualizan las salidas escribiendo los nuevos valores al finalizar la fase. También se ejecuta el programa guardado en su memoria. El tiempo que tarde el PLC en ejecutarlo depende del procesador utilizado, de la longitud del programa, del tipo de instrucciones usadas en el programa y del número y ubicación de las interfaces de E/S. Este tiempo suele ser del orden de 5ms para un autómata compacto con interfaces locales. No cuesta lo mismo ejecutar instrucciones AND, OR, que otras de movimiento de datos o de transmisión a través de un puerto.

Ø Fase de atención a periféricos:

Esta fase sólo se atiende si existen peticiones de intercambio de información pendientes con los periféricos conectados a la CPU o a procesadores auxiliares a éste. El autómata contiene rutinas cíclicas de auto-chequeo y de arranque en ROM con las que comprueban el programa de usuario y el propio hardware y en caso de detectar algún problema, registrarlo y, si es grave, detener la ejecución e informar del mismo mediante LEDs o un número de error en un registro.

### **1.2.9 Introducción a la programación de PLC.**

Hasta ahora sólo se ha tratado el hardware de los autómatas, por lo que a partir de este momento nos introduciremos en la parte de software.

El software asociado a un autómata consta de dos partes; la primera conforma el sistema operativo (firmware) residente en el autómata que ejecuta las órdenes del programa de usuario y vigila el correcto funcionamiento del equipo; por otra parte tenemos el software de edición y depuración de programas que permite escribir e introducir el código en el autómata o en un soporte físico adecuado por parte del programador desde un dispositivo como un PC o una consola de programación.

Tenemos entonces una unidad de programación sobre la que se escribe un programa en un determinado lenguaje y que luego se conecta al PLC para traspasarle la información. Posteriormente, el firmware del PLC interpretará el código y ejecutará el programa.

Para realizar un programa comprensible por el PLC, se ejecutan diversos pasos.

Una vez que se sabe lo que se quiere hacer, en qué orden y los medios de que se dispone (E/S, memoria, etc.) se ha de realizar una representación del sistema de control con un modelo que indique las funciones que se van a realizar mediante una representación que puede ser algebraica o gráfica.

Después se asignan las salidas y entradas a los distintos elementos de nuestro modelo y se codifica el modelo anterior en un lenguaje entendible por el PLC. Por último, el código se transfiere al autómata.

Las representaciones de los sistemas de control son una manera de expresar mediante un lenguaje lo que hace exactamente el proceso que queremos estudiar o lo que queremos que

haga. Se puede expresar de diversas maneras como por ejemplo con lenguaje usual (descripción literal) pero puede que no nos permita ser muy exactos en las descripciones debido a la complejidad de los sistemas y de la necesidad de dejar perfectamente especificadas las condiciones de trabajo, lo que nos obliga a utilizar útiles simbólicos de representación que nos permitan ser más precisos y dar toda la información necesaria.

Estas representaciones deben de ser comunes a los que la utilicen y coherentes en las reglas que empleen (sintaxis).

Formas de representación:

- Proposicional: descripciones literales.
- Algebraica: funciones booleanas y aritméticas.
- Gráfica: esquemas, diagramas, grafcet.

ØDescripciones literales:

Esta manera de describir el proceso mediante el lenguaje corriente, no será exacta en las especificaciones aunque explique correctamente el proceso a nivel general. Así, es difícil expresar relaciones entre variables, etc.

ØFunciones algebraicas:

Esta representación utilizaría funciones booleanas y aritméticas para establecer condiciones entre variables de uno o varios bits. Aunque este sistema es exacto, adolece de no poder expresarse en función del tiempo.

ØDiagramas lógicos (o Plano de funciones):

Utiliza símbolos que representan las funciones lógicas: and, or, etc., o biestables, registros, contadores, etc., tal y como se muestra en la Figura 1.8. Este modo de representación es útil pero está alejada de muchos usuarios finales del autómata, quedando circunscrito a la representación del sistema en alto nivel, a efectos de documentaciones y especificaciones.

ØOrdinogramas:

Este modelo refleja con especial claridad la evolución y toma de decisiones del proceso aunque no es tan claro si se intentan representar variables y señales que intervienen y las

relaciones entre ellas a no ser que se utilicen expresiones algebraicas lo que va en contra de la claridad del gráfico del modelo. Figura 1.9.

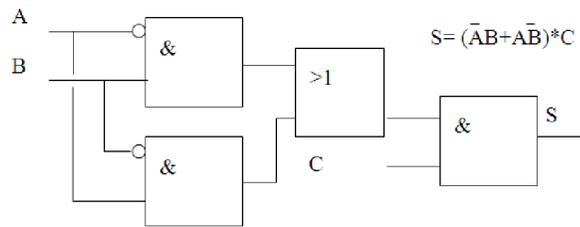


Figura 1.8. Diagrama lógico

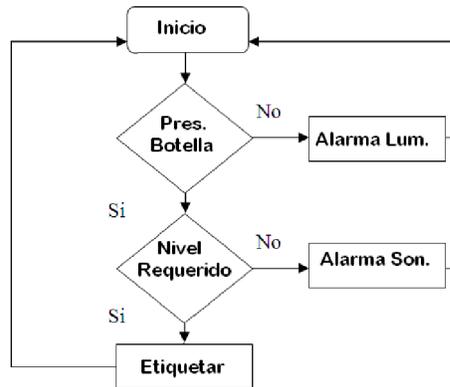


Figura 1.9. Ordinograma.

#### Ø Representación GRAFCET:

Es una evolución del diagrama de flujos que representa la sucesión de los distintos estados, separados por transiciones, por los que pasa el sistema. Comienza con una etapa inicial y evoluciona por el resto según las condiciones de las transiciones. En cada etapa se representan las distintas acciones a realizar en ella y que se pueden representar aparte con cualquiera de las representaciones anteriores (por ejemplo con diagrama de relés).

#### Ø Lenguaje de Lista de instrucciones (Nemónico):

Este lenguaje de nemónicos utiliza básicamente las funciones del álgebra de Boole aunque como resultan insuficientes para todas las tareas que puede realizar el autómata, se amplían con un juego de instrucciones mayor como son de contadores (CNT), temporizadores (TIM), manipulación de datos (comparar datos CMP, mover datos MOV), gestión de programa (final de programa END, salto a otro bloque JMP) o aritméticas (suma ADD, multiplicación MUL).

### ØLenguaje de Diagramas de contactos (*Ladder Diagram*):

Es un lenguaje gráfico que expresa relaciones entre señales binarias como una sucesión de contactos en serie y en paralelo (ver Figura 1.10). Estos contactos, con dos estados, (abierto/cerrado) permiten su combinación formando todo tipo de funciones lógicas equivalentes directa e inmediatamente a funciones en lenguaje nemónico.

Esta manera de representación nació como representación del diagrama de relés de uso común en automatización y contiene contactos y bobinas a modo de relés de salida o internos además de otras funciones, desde timers y contadores hasta funciones de comunicaciones.

Al igual que el lenguaje de Lista de instrucciones, éste ha tenido que ampliarse con otras instrucciones mediante bloques funcionales complejos que permiten la manipulación de datos, gestión del programa, funciones aritméticas y otras muchas operaciones, ya que era un tanto deficiente en representación de funciones secuenciales complejas como son los secuenciadores, los registros de memoria, etc., y, sobre todo las señales digitales de varios bits.

Este lenguaje es muy utilizado en todo el mundo y será el que usaremos en nuestro proyecto gracias a las comodidades visuales de comprensión que presenta.

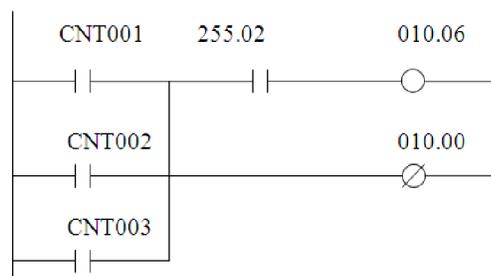


Figura 1.10. Diagrama de contactos.

Los lenguajes de programación vuelven a ser, como las distintas maneras de representaciones anteriores, algebraicos y gráficos y, aunque sean distintos, intentan con unas reglas sintácticas definidas, construir un programa coherente y comprensible por el autómeta.

Todos los lenguajes poseen instrucciones con una estructura similar:

### **“Campo de operación” “Operando(s)”**

El Campo de Operación indica qué hacer como el tipo de operación, etc., mientras que el Operando dice con quién hay que realizarlo como por ejemplo una dirección de memoria (DM150.13) o un valor numérico (134).

Los lenguajes son lo suficientemente completos y complejos como para realizar cualquier tipo de operación desde las más sencillas, como las booleanas, hasta las más complejas, como movimiento de datos, rotaciones de bits o manipulación de ficheros.

Además del lenguaje propiamente dicho puede ser necesario pasarle al autómatas ciertos parámetros para configurar su entorno como por ejemplo si está conectado a red o cómo actuará en caso de fallo de tensión o al volver a arrancar.

### **1.3 Campos de aplicación.**

EL PLC por sus especiales características de diseño tiene un campo de aplicación muy extenso. La constante evolución del Hardware y Software amplía continuamente éste campo para poder satisfacer las necesidades que se detectan en el aspecto de sus posibilidades reales. Su utilización se da fundamentalmente en aquellas instalaciones donde es necesario realizar procesos de maniobra, control, señalización, etc. y por lo tanto, su aplicación abarca desde procesos de fabricación industrial de cualquier tipo hasta de transformaciones industriales, control de instalaciones, etc.

Sus reducidas dimensiones, las extremas facilidades de montaje, la posibilidad de almacenar los programas para su posterior y rápida utilización, la modificación o alteración de los mismos, etc., hace que su eficiencia se aprecie fundamentalmente en procesos en que se producen necesidades tales como: Espacio reducido, Procesos de producción periódicamente cambiantes, Maquinaria de procesos variables, Instalación de procesos complejos y amplios, entre otros.(Guille, 2001)

### **Campos de acción.**

Ø Maniobra de máquinas.

- Maquinaria industrial del mueble y la madera.

- Maquinaria en proceso de grava, arena y cemento.
- Maquinaria en la industria del plástico.
- Máquinas-herramientas complejas.
- Maquinaria de ensamblaje.
- Máquinas de transferencia.

Ø Maniobra de instalaciones.

- Instalaciones de aire acondicionado y calefacción.
- Instalaciones de seguridad.
- Instalaciones de almacenamiento y transporte.
- Instalaciones de plantas embotelladoras.
- Instalaciones en la industria automotriz
- Instalación de tratamientos térmicos.
- Instalaciones de la industria azucarera

Ø Señalización y control.

- Chequeo de programas.
- Señalización del estado de procesos.

Es interesante hacer notar que aunque el PLC fue originalmente diseñado como un dispositivo de reemplazo de control industrial, cumple las necesidades de los usuarios. Las necesidades de la aplicación pueden ser definidas solamente por un análisis detallado del sistema completo. Esto significa que los exámenes detallados deben ser ejecutados en todas las facetas de la máquina u operación del proceso. De nuevo, como toda aplicación es diferente, no hay una rutina clara y concisa que evalúe las necesidades de todas las aplicaciones. Una consideración importante en la aplicación de un PLC es el futuro crecimiento del sistema. Los PLC están diseñados modularmente y por lo tanto con posibilidades de poder expandirse para satisfacer las necesidades de la industria.

### 1.3.1 Ejemplos de sus aplicaciones en la industria nacional y extranjera.

Como veníamos explicando, dada las potencialidades que brindan los PLC's los encontramos diseminados por todo el mundo, formando parte desde los más sencillos automatismos, hasta complejos procesos industriales. Un ejemplo lo podemos encontrar en el sistema de control de la planta de empaque de productos farmacéuticos de la empresa MERCK, en Costa Rica. Los Controladores Lógicos Programables que se encuentran en el proyecto de líneas de empaque en MERCK son parte de un sistema llamado BAS (*Building Automation System*). Este sistema se encarga de controlar todo lo referente al ambiente que se encuentra por debajo de las máquinas, donde procesos de manufactura muy delicados (debido a las condiciones de higiene y almacenamiento) se llevan a cabo.

También lo podemos encontrar en la industria azucarera, en nuestro país y en muchos países de América Latina, por ejemplo, en Costa Rica. La Quebrada Azul es un Ingenio azucarero ubicado en la zona norte de este país, que ha utilizado a la empresa Siemens para automatizar las diferentes partes del proceso, donde el PLC tiene una participación muy importante. Primero en el ingenio, se realiza una supervisión de las calderas (utilizando S7-300 CPU 315-2DP, una computadora con una tarjeta Profibus-CP5613 y un panel operador OP1708). En una computadora se pueden visualizar el comportamiento de las calderas y demás procesos. (Sánchez, 2004)

El PLC como elemento de control, también, es utilizado en esta industria, adicionando algunos elementos como sensores de velocidad de tipo inductivo para mejorar la sincronía entre las diferentes etapas del centrifugado y las condiciones de trabajo en cada una de ellas; así como otros elementos mecánicos del sistema de control. Permite además la sincronía de operación de diferentes centrifugas aprovechando las funciones de comunicación inherentes de los PLC, por medio de lo cual puede conectarse a algún sistema con fines de supervisión y control.

En la industria cubana también encontramos los autómatas formando parte desde los más simples hasta los más complejos procesos, por ejemplo en la Planta IMS, perteneciente a la Empresa de Prefabricados #4 (EPP4 de la provincia Villa Clara) posee un sistema de control el cual está implementado sobre un autómata programable de la serie C200HX-

CPU44 de la OMRON el cual permite seleccionar recetas del cemento prediseñadas o hacer otras nuevas, donde el operador le pasa al PLC los valores a los cuales desea fijar los diferentes parámetros. La planta puede ponerse a trabajar de forma automática o manual y utiliza un Terminal NT11 como interfaz con el operador para entrada de datos y visualización del estado del proceso y demás parámetros prefijados. (Montejo, 2005)

Otra aplicación de los autómatas en nuestra industria la podemos encontrar en el pasteurizador de la fábrica de cerveza de Manacas "Antonio Díaz Santana", donde son capaces de gobernar todo el proceso de la línea de pasteurizado. Esta línea fue modernizada recientemente implementándose un sistema supervisor.

Este sistema tiene como objetivos tener una visualización del proceso de pasteurización así como controlar y manipular a través de este todas las variables existentes del proceso.

En este caso la propuesta consistió en el uso de los software empleados en el desarrollo de sistemas supervisores y aplicaciones de control que son capaces de intercambiar datos con el proceso a través de la comunicación con PLC's, tarjetas de entrada / salida, entre otras. La implementación de tal tipo de software ayudaría a los operadores de procesos a tener una clara idea de lo que está pasando dentro del proceso de pasteurización a través de la visualización de las variables más importantes, que a su vez pueden ser manipuladas por los operadores desde una PC en casos necesarios.

La implementación del supervisor no sólo trajo mejores condiciones de trabajo, sino también aumentó en gran nivel el control, la automatización y de forma general, la productividad del proceso. (Ambrose, 2004)

## **CAPÍTULO 2. IDENTIFICACION DE VARIABLES Y SISTEMA SUPERVISOR**

### **2.1 Características y funciones básicas del CPM2A.**

Los PLC's CPM2A incorporan una variedad de características en una unidad compacta que incluye control sincronizado de pulsos, entradas de interrupción, salidas de pulsos, selecciones analógicas y una función de reloj. Además la CPU CPM2A es una unidad compacta que puede gestionar un amplio rango de aplicaciones de control de máquina, lo que la hace ideal para ser integrada en la propia máquina como unidad de control.

El CPM2A dispone de funciones de comunicación con ordenadores personales, otros PLC's OMRON y Terminales Programables OMRON. Estas capacidades de comunicación permiten al usuario diseñar sistemas de producción distribuidos de bajo costo.

#### **2.1.1 CPU.**

La CPU contiene 20, 30, 40 ó 60 puntos de E/S y se pueden añadir unidades de expansión de E/S hasta obtener un total de 120 puntos de E/S. También se pueden conectar unidades de E/S analógicas y unidades I/O Link de CompoBus/S. Hay 3 tipos de salidas disponibles (salidas relé, salidas transistor NPN y salidas transistor PNP) y 2 tipos de fuentes de alimentación (100/240 Volts CA. ó 24 Volts CC.). Ver Tabla 2.1.

Tabla 2.1. CPU con 20/30 puntos de E/S.

Número de puntos de E/S.	Fuente de A.	Entradas.	Salidas.	Modelos.
20 puntos de E/S (12 entradas y 8 salidas)	100 a 240 VCA.	24 VCC.	Relé.	CPM2A-20CDR-A
		24 VCC.	Relé	CPM2A-20CDR-D
	24 VCC.	24 VCC.	Transistor negativo común o NPN	CPM2A-20CDT-D
		24 VCC.	Transistor positivo común o PNP	CPM2A-20CDT1-D
30 puntos de E/S (18 entradas y 12 salidas)	100 a 240 VCA.	24 VCC.	Relé	CPM2A-30CDR-A
		24 VCC.	Relé	CPM2A-30CDR-D
	24 VCC.	24 VCC.	Transistor negativo común o NPN	CPM2A-30CDT-D
		24 VCC.	Transistor positivo común o PNP	CPM2A-30CDT1-D

### 2.1.2 Unidades de E/S de expansión.

Hasta 3 unidades de E/S de expansión se pueden conectar a la CPU para aumentar la capacidad de E/S del PLC hasta un máximo de 120 puntos. Hay 3 tipos de unidades de E/S de expansión disponibles: una unidad de 20 puntos de E/S, una unidad de 8 puntos de entrada y una unidad de 8 puntos de salida. La capacidad máxima de E/S se obtiene conectando 3 unidades de expansión de 20 puntos a una CPU con 60 puntos de E/S incorporados.

### 2.1.3 Unidades de E/S analógicas.

Para disponer de entradas y de salidas analógicas se pueden conectar hasta 3 unidades de E/S analógicas. Cada unidad dispone de 2 entradas analógicas y 1 salida analógica. (Si se combinan los puntos de E/S analógica con las instrucciones PID (----) y PWM (----) se puede efectuar un control de tiempo proporcional)

- El rango de entrada analógica se puede fijar a uno de 0 a 10Vc.c., a 5Vc.c., ó 4 a 20 mA con una resolución de 1/256. (La función de detección de circuito abierto se puede utilizar con las selecciones de 1 a 5 V C.C. y de 4 a 20 mA).
- El rango de salida analógica se puede establecer a 0 a 10 V C.C., -10 a 10 VC.C., ó 4 a 20 mA con una resolución de 1/256.

#### 2.1.4 Contadores de alta velocidad e interrupciones.

El CPM2A tiene un total de cinco entradas de contador de alta velocidad. La entrada de contador de alta velocidad tiene una frecuencia de respuesta de 20 kHz/5 kHz y las cuatro entradas de interrupción (en modo contador) tienen una frecuencia de respuesta de 2 kHz.

El contador de alta velocidad se puede utilizar en uno de los cuatro modos de entrada siguientes: modo de diferencia de fase (5 kHz), modo de entrada de pulso más dirección (20 kHz), modo de pulso más/menos (20 kHz), o modo incremental (20 kHz). Las interrupciones se pueden lanzar cuando el contaje coincide con un valor fijado o cae en un rango especificado.

Las entradas de interrupción (modo contador) se pueden utilizar para contadores incrementales o contadores decrementales (2 kHz) y lanzar una interrupción (ejecutar el programa de interrupción) cuando el contaje coincide con el valor objeto.(OMRON, 2004)

#### 2.1.5 Resumen de sus principales características.

§ Juego de instrucciones:

- 14 instrucciones básicas.
- 105 instrucciones especiales y 185 variaciones.

§ Capacidad de programa.

- 4096 palabras.

§ Número máximo de E/S:

- CPU: 20, 30 Y 60 puntos.
- CPU con unidades de expansión de E/S: 90, 100, o 120 puntos máximo.

§ Unidades de expansión.

- Número máximo de unidades: Un máximo de 5 unidades se pueden conectar a cualquier CPU.
- Unidades disponibles: Unidades de expansión de E/S, unidad de E/S analógica y unidad I/O Link de CompoBus/S.

A continuación se muestra en la tabla 2.2 la memoria de E/S del autómatas SYSMAC-CPM2A.

En el Anexo I y II se mostrarán otras características de dicho PLC que es necesario conocer para el trabajo con el mismo, así como una comparación con otro AP de OMRON, de similares características y prestaciones.

Tabla 2.2. Memoria de E/S del CPM2A.

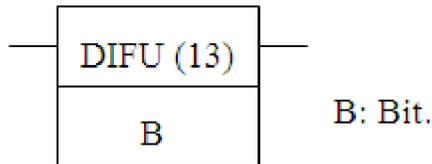
Memoria de E/S.	Bit de entrada.		IR 00000 a IR 00915
	Bit de Salida.		IR 01000 a IR 01915
	Bit de Trabajo.		IR 02000 a IR 04915, IR 20000 a IR 22715: 928 bits:
	Área SR (relés especiales)		SR 22800 a SR 25515: 448 bits:
	Área TR (relés temporales)		8 bits: TR0 a TR7
	Área AR (relés auxiliares)		AR 0000 a AR 2315: 384 bits:
	Área LR (relés de enlace)		LR 0000 a LR 1515: 256 bits:
	Área de TIM/CNT.		TIM/CNT 000, TIM/CNT 255: 256 bits:
	Área DM (Memoria de datos)	Área de R/W.	(DM 0000 a DM 2047): 2,048 palabras
		Área de sólo lectura	(DM 6144 a DM 6599): 456 palabras
Setup del PLC		(DM 6600 a DM 6655): 56 palabras	

**2.2 La ingeniería inversa.**

**2.2.1 Funciones empleadas en el programa del autómatas.**

A continuación se explicarán las funciones más importantes que presenta el código del PLC, para comprender su funcionamiento:

- DIFU(13):

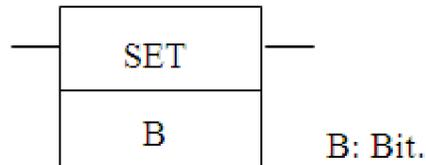


Descripción: DIFU (013) pone en ON el bit designado (B) durante un ciclo de scan cuando la señal de entrada pasa a ON. La próxima vez que se ejecute DIFU (13) se pondrá en OFF el bit B sin tener en cuenta el estado de la señal de entrada.

Rangos:

B: Bit IO, AR, HR, LR.

• SET:

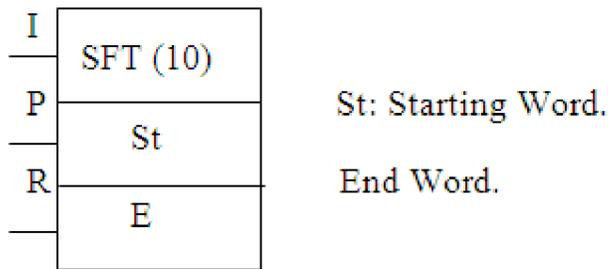


Descripción: Pone el bit operando a ON cuando la condición de ejecución es ON y no afecta al estado del operando cuando la condición es OFF.

Rangos:

B: Bit IO, AR, HR, LR.

• SFT(10):



Descripción: Se controla por tres condiciones de ejecución, I, P y R.

Si SFT (10) se ejecuta y 1) la condición de ejecución P está en ON y 2) R está en OFF, la condición de ejecución I se desplaza al bit menos significativo (LSB) de un registro de desplazamiento definido entre St y E, es decir, si I está en ON, se desplaza 1 en el registro; si I es OFF, se desplaza un 0. Al desplazar el estado I en el registro, todos los bits del registro se desplazan una posición a la izquierda, perdiéndose el bit de la izquierda. Ver figura 2.1.

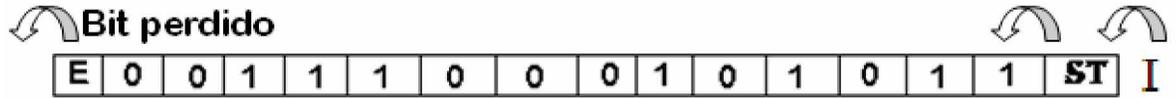


Figura 2.1. Registro desplazamiento.

§ La condición de ejecución en P funciona como una instrucción diferenciada, es decir, I se desplazará en el registro sólo cuando P esté en ON y la vez anterior que se ejecutó SFT (10) estaba en OFF. Si no cambia la condición de ejecución P, o lo hace de ON a OFF, el registro de desplazamiento no sufre variación.

§ St designa el canal de la derecha del registro de desplazamiento; E designa el de la izquierda. El registro de desplazamiento incluye los dos canales anteriores y todos los comprendidos entre ambos. El mismo canal puede designarse para St y E para crear un registro de desplazamiento de 16 bits (es decir, un canal).

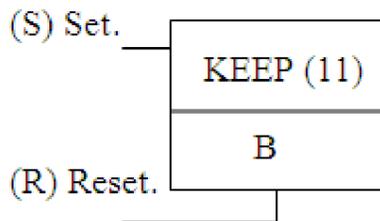
§ Cuando la condición de ejecución R se pone en ON, todos los bits en el registro de desplazamiento se pondrán a OFF y el registro no operará hasta que R se ponga en OFF de nuevo.

Rangos:

St: Canal inicial IO, AR, HR, LR.

E: Canal final IO, AR, HR, LR.

• KEEP

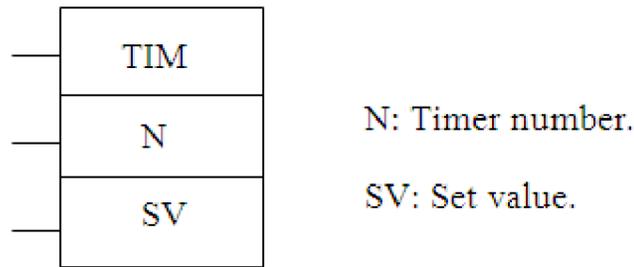


Descripción: Define un bit (B) como de enclavamiento, activado por una entrada de set (S) y por una de reset (R).

Rangos:

B: Bit IO, AR, HR, LR.

- TIM:



Descripción:

Un temporizador se activa cuando su condición de ejecución es ON, y se resetea (a SV) cuando la condición de ejecución se pone en OFF. Una vez activado, TIM mide en unidades de 0,1 segundo desde el SV.

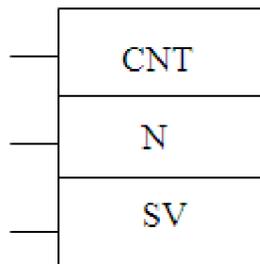
Si la condición de ejecución permanece en ON lo suficiente para que transcurra el tiempo fijado en TIM, se pondrá a ON el indicador de finalización del número de TC utilizado y permanecerá en dicho estado hasta que se resetee TIM (es decir, hasta que su condición de ejecución se ponga en OFF).

Rangos:

N: Número TC      000 - 511

SV: Valor seleccionado (canal, BCD)    IO, AR, DM, HR, #

- CNT:



Descripción:

CNT se utiliza para descontar a partir de SV cuando la condición de ejecución en el impulso de conteo, CP, pase de OFF a ON, es decir el valor presente (PV) será reducido en uno, siempre que CNT se ejecute con una condición de ejecución ON

para CP y la condición de ejecución fuera OFF para la última ejecución. Si la condición de ejecución no cambió o cambió de ON a OFF, el PV de CNT no cambiará. El indicador de finalización para un contador se pone a ON cuando el PV alcanza cero y permanecerá en ON hasta que el contador se resetee.

CNT se resetea con una entrada de reset, R. Cuando R pasa de OFF a ON, el PV se resetea a SV. El PV no decrementará mientras R esté en ON. El descuento desde SV comenzará de nuevo cuando R se ponga en OFF. El PV para CNT no se resetea en secciones de programa enclavadas o por cortes de alimentación.(OMRON, 2004)

Rangos:

N: Número de TC 000 - 511

SV: Valor seleccionado (canal, BCD) IO, AR, DM, HR, #.

### 2.2.2 Ingeniería inversa al código del programa.

El código del programa extraído de la memoria del PLC presenta los comentarios en italiano, por lo que, para mejor comprensión, primeramente se procederá a la traducción de los distintos comentarios de todos los segmentos de código del programa.

Como se había mencionado con anterioridad, del código del programa del autómatas que se puede encontrar en el Anexo III, se desea ver si es posible obtener los siguientes resultados:

- El número de etiquetas, contraetiquetas y collarines que se ponen correctamente.
- El número de botellas que entran a la etiquetadora.
- El número de botellas que salen sin etiqueta, contraetiqueta y/o collarín.
- Máquina en marcha o detenida.

A continuación analizaremos el funcionamiento de la máquina etiquetadora a partir del análisis del programa del autómatas que la controla y de aspectos empíricos obtenidos de la experiencia de los operadores de dicha máquina.

Inicialmente al ponerse en marcha el proceso, el sensor k140 detecta si hay presencia de botella y envía un “1” lógico por el pin de entrada 0001 del PLC. Al detectarse esta señal se ejecutará la función DIFU (13) (línea 1-2) la cual activará el bit 200.00 durante un scan

poniéndose a “1” los bits 210.00, 212.00 y 214.00 de los registros 210, 212 y 214 respectivamente, los cuales se encargarán del control del collarín, la etiqueta y la contraetiqueta en ese propio orden.

A partir de este momento sólo se analizará lo relacionado al registro 212, encargado de controlar la puesta de la etiqueta, debido a que durante todo el programa se tratan de igual forma a los demás aditamentos pero con la excepción de que emplean los registros 210 y 214.

En la práctica se determinó que la etiquetadora trabaja por pasos, es decir, que se detiene momentáneamente en el momento en que las botellas están justo al frente de los diferentes sensores y luego da un paso hasta caer nuevamente en el mismo estado. Después de detectarse la presencia de la botella y registrarse (línea 1-2 y 5-8) la máquina da un paso. Este momento de transición de un estado a otro, o sea, cuando está en movimiento, también es detectado por un transductor (B41). La señal enviada por este al PLC es recibida por la entrada 00002, la cual, con el pulso de subida, ejecutará la instrucción DIFU (13) poniéndose a “1” el bit 200.01 durante un scan (línea 3-4), el cual a su vez activará la función SFT (10) (línea 9-10-11), la cual tal y como se había explicado con anterioridad, insertará en el bit menos significativo (LSB) del registro 212, es decir en 212.00, el valor de I, que en este caso es la bandera 253.14 (siempre OFF) es decir, un “0” lógico. Resumiendo, cada vez que la máquina da un paso se inserta un cero en el bit 212.00 del registro 212 y se desplazan a la izquierda los demás valores, y cuando se detiene si hay presencia de botella se fija a “1” el propio bit 212.00. En la secuencia del programa encontramos ahora en la línea 23-24 la lectura del sensor B24 que detecta la presencia de etiqueta en la botella, la señal de presencia de la misma es cargada por el autómatas a través del contacto 0.06, que procede a resetear el bit 212.04. En caso de no detectarse etiqueta no se resetea este bit y al dar un paso la máquina este “1” se desplazará hasta llegar al bit 212.09 el cual, como se aprecia en la línea 28, ejecutará una función KEEP, la que pondrá en “1” el bit 220.00, que a su vez dará salida para activar una alarma sonora intermitente durante cuatro segundos (TIM00), por ausencia de etiqueta en la botella. Esta última operación se realiza para avisar al operador de la máquina que debe sacar de la línea una botella defectuosa (sin etiqueta).

Este mismo “1” que se está desplazando por todo el registro 212, debido a que no fue reseteado por no haber etiqueta en la botella, será leído además por en el bit 212.05 y será contado, en caso de ser “1” por CNT001, el cual contará hasta tres ausencias consecutivas (si no es de forma consecutiva se resetea) y luego se pondrá en ON y activará la salida 10.06, la que controla una lámpara roja intermitente, indicando al operador problemas técnicos, y activará además la salida 10.00 la cual hará que se detenga la máquina (línea 64-66).

De las variables que se quieren mostrar en el SCADA ninguna se puede obtener directamente del programa residente en el autómatas. Para obtener estos valores deseados será necesario hacerle algunas modificaciones al código del programa, las cuales se propondrán más adelante.

### **2.3 Sistema supervisor.**

Para la programación de SCADA podemos encontrar en el mercado una gran variedad de software. Entre los más populares y más usados se pueden mencionar al LabVIEW, WinCC de Siemens y MOVICON. En el presente epígrafe se pretende hacer un estudio de estos software y se hará hincapié fundamentalmente en el MOVICON X, el cual se empleará para diseñar la propuesta del sistema supervisor que se desea lograr. Ya una vez realizada la ingeniería inversa al programa del autómatas, conociendo así su funcionamiento, y ubicación en las distintas zonas de memoria de todas las variables, y propuestos además los posibles códigos que se deben agregar a dicho programa, sólo nos resta hacer un estudio de los diferentes sistemas supervisores que se pudieran implementar y específicamente al MOVICON.

#### **2.3.1 LabVIEW.**

LABVIEW, es una herramienta diseñada por la *National Instruments* especialmente para monitorear, controlar, automatizar y realizar cálculos complejos de señales analógicas y digitales capturadas a través de tarjetas de adquisición de datos, puertos serie y GPIBs (Buses de Intercambio de Propósito General).

Es un lenguaje de programación de propósito general, como es el Lenguaje C o Basic, pero con la característica que es totalmente gráfico, facilitando de esta manera el entendimiento y manejo de dicho lenguaje para el diseñador y programador de aplicaciones tipo SCADA.

Incluye librerías para la adquisición, análisis, presentación y almacenamiento de datos, GPIB y puertos serie.

Está basado en la programación modular, lo que permite crear tareas muy complicadas a partir de módulos o sub-módulos mucho más sencillos. Estos módulos pueden ser usados en otras tareas, lo cual permite una programación más rápida y provechosa.

#### ØFilosofía de LABVIEW.

Existen dos conceptos básicos en LabVIEW: el *Front Panel* (Panel Frontal) y el *Block diagram* (Diagrama de Bloque). El Panel Frontal es la interfaz que el usuario está viendo y puede ser parecido al proceso del cual se están recogiendo los datos, de esta manera el usuario sabe exactamente cuál es el estado actual de dicho proceso y los valores de las señales que se están midiendo. El diagrama de bloques es el conexionado de todos los controles y variables, que tendría cierto parecido al diagrama del esquema eléctrico del instrumento.

Es un sistema abierto, en cuanto a que cualquier fabricante de tarjetas de adquisición de datos o instrumentos en general puede proporcionar el controlador de su producto en forma de VI dentro del entorno de LabVIEW.

#### 2.3.2 WinCC.

¿Qué es WinCC?

WinCC, es un potente sistema HMI (*Human Machine Interface*) que se utiliza bajo Microsoft Windows. El control sobre el proceso en sí lo tiene el autómeta. Es decir, por un lado hay una comunicación entre WinCC y el operador, y por otro lado entre WinCC y los autómetas programables. Con WinCC se visualiza el proceso y se programa la interfaz gráfica de usuario para el operador.(SIEMENS, 2005)

- WinCC permite que el operador observe el proceso, mediante la visualización gráfica en la pantalla. En cuanto cambia un estado en el proceso, se actualiza la visualización.

- WinCC permite que el operador maneje el proceso; así, desde la interfaz gráfica de usuario, él puede predeterminar un valor de consigna, abrir una válvula, etc.
- Cuando se presenta algún estado crítico en el proceso se activa automáticamente una alarma; si se rebasa un valor límite predeterminado, por ejemplo, aparece un aviso en la pantalla.
- Los avisos y los valores de proceso, se pueden imprimir y archivar en formato electrónico. El usuario documenta así la evolución del proceso y puede acceder posteriormente a los datos de producción del pasado.

#### Ø Sistema básico WinCC.

- Sistema de gráficos
- Sistema de avisos
- Sistema de archivos
- Sistema de informes
- Comunicación
- Administración de usuario

### 2.3.3 MOVICON.

MOVICON (Monitorio, Visión y Control), es una herramienta para compañías que trabajan en el campo de la automatización y el control de procesos. Es un producto de Progea, compañía italiana que ha desarrollado productos software para la automatización industrial desde 1990.(Progea, 2007)

MOVICON permite la adquisición de datos a través de su comunicación con el PLC, red y bus de campo, así como la configuración de herramientas y sensores. Los datos adquiridos son coleccionados dentro de una base de datos en tiempo real (RTDB) y luego están disponibles para todos los objetos y recursos para crear de forma animada, sinópticos, alarmas, recetas, gráficos y reportes.

ØMOVICON tiene las siguientes características:

- Sistema SCADA/ HMI para Win32.

- Trabaja bajo sistemas operativos desde Win2000 hasta WinCE.
- Es un sistema Cliente/Servidor de 32 bit.
- Contiene una amplia biblioteca de símbolos, objetos y gráficos.
- OPC<sup>5</sup> cliente y servidor.
- OPC-DA (Data Access.)
- OPC-AE (Alarma y Eventos.)
- Programación en Basic Script que es 100% compatible con VBA (*Visual Basic for Application*).
- Contiene editores de menú y cajas de diálogo.
- Posibilidad de usar lenguaje de PLC en las lógicas que él dispone.
- Tiene un administrador de alarmas.
- Soporte de tecnología ActiveX.
- Soporte de OLE<sup>6</sup>, ODBC<sup>7</sup>, DDE, DAO<sup>8</sup>/ADO<sup>9</sup>, SQL y OPC.
- Red de Cliente/Servidor de TCP/IP.
- Objetos PID integrados.
- Gráficos y hojas de trabajo.
- Un depurador (*debugger*) integrado.
- Administración de estadísticas de evento o producción.

---

<sup>5</sup> OLE para el control de procesos. Conjunto de protocolos para el intercambio de datos en tiempo real.

<sup>6</sup> Object Linking and Embedding.

<sup>7</sup> Open Database Conectivity.

<sup>8</sup> Data Access Object.

<sup>9</sup> ActiveX Data Object.

Los datos de un proyecto hecho en MOVICON son llevados a una base de datos de variables (*Real Time DataBase* (RTDB)). La base de datos de la variable, puede ser importada o exportada a través de ODBC, colecciona todos los datos de los drivers y los distribuye a los recursos del proyecto. La base de datos de variables puede disparar de forma directa a los controles o alarmas. Esta base de datos permite la conexión a través de TCP/IP a estaciones remotas y se conecta dinámicamente a bases de datos externos a través de ODBC, o son disponibles a otras aplicaciones gracias a la funcionalidad del Servidor OPC integrado.(Vidal, 2002)

La tecnología MOVICON no se degrada en funcionamiento cuando se aumenta el número de variable porque ellas son conectadas directamente sin que se acceda a la base de datos durante el tiempo de corrida. Así se pueden implementar proyectos de gran cantidad de datos sin que se afecte la eficiencia y funcionamiento del programa.

Algunas variables en MOVICON pueden considerar retentiva para mantener su estado, aun cuando la PC ha sido apagada. Las variables pueden ser de tipo byte, word, double word, float y array. El modo de comunicación estándar es OPC.

MOVICON fue una de las primeras aplicaciones SCADA que implementó el uso de OPC integrado. (Progea, 2006)

Para la comunicación usa los siguientes estándares:

- Cliente OPC versión 1.0 y 2.0
- Servidor OPC versión 1.0 y 2.0
- OPC DA (*Data Access*).
- OPC AE (*Alarms & Events*).

Se compone de un ambiente formado por un editor de objetos, que en conjunto de librerías gráficas pueden implementar mímicos animados. Los mímicos también ofrecen una interfaz a Basic Script VBA, suministrando al programador eventos, métodos y propiedades. De esta manera el usuario puede implementar cualquier tipo de objeto gráfico a través de la manipulación de las funciones. En la figura 2.2 se pueden observar algunos de los objetos que se pueden usar para la conformación de mímicos empleando este software.

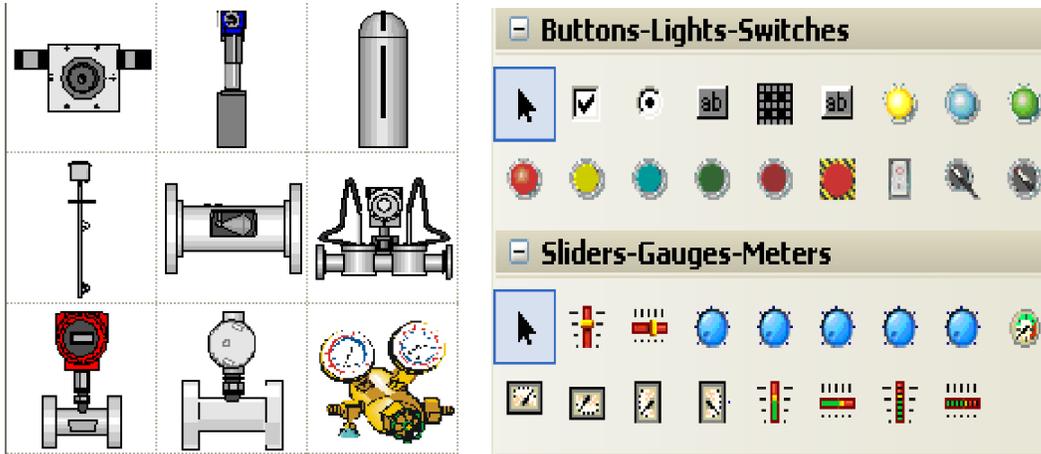


Figura 2.2. Algunos de los objetos disponibles en el MOVICON.

MOVICON soporta dos tipos de elementos: símbolos y objetos. Los símbolos están organizados en librerías con clases que son expandibles. Los objetos son elementos vectoriales que tienen una función y pueden ser configurados con funciones de estilo y control. MOVICON también tiene un componente que implementa la interfaz gráfica para el diálogo con el operador del sistema. Este componente, que es una caja de diálogo, se usa para poder manipular los puntos de ajuste, ajustar datos, selecciones y opciones.

El editor de alarma permite al usuario identificar las anomalías y extraer algunas sugerencias. El Historical Log registra las alarmas y eventos del sistema, así mismo los tiempos que fueran activados, quienes lo atendieron y lo resetearon. El *Data Logger* permite que datos de producción, variables de procesos continuos y valores que serán analizados, sean documentados y registrados por tiempo, evento o cambio de estado, dentro de la base de datos ODBC. También el *Data Logger* permite al usuario ver el comportamiento gráfico de los datos (*Trends*) y los reportes del proceso que son fundamentales para el análisis de la productividad de la planta.

### **CAPÍTULO 3. IMPLEMENTACION DEL SISTEMA SUPERVISOR Y ANALISIS DE LOS RESULTADOS**

Como se había planteado en capítulos anteriores, el principal objetivo de este trabajo es la realización de la ingeniería inversa al programa del autómeta SYSMAC-CPM2A, que es encargado del control de la máquina etiquetadora. Además se había planteado proponer el código de programación que garantice conocer las variables antes mencionadas, que no estén disponibles en el programa; y como último objetivo teníamos la implementación de una pequeña aplicación SCADA, empleando el software de programación MOVICON X, que represente el proceso de etiquetado y se realicen tareas básicas de supervisión y control en esta planta. En función de esto, en el presente capítulo, dando cumplimiento a los objetivos, se mostrará la tabla de memoria del PLC con las diferentes direcciones de sus variables, tipo, nombre que se le da en el programa y una breve descripción de su función. Además de esto se pretende mostrar la propuesta de reprogramación del autómeta, describiendo el objetivo a conocer y la acción propuesta, para aquellas variables que no fueron posible obtener. Y por último expondremos como se implementó el sistema supervisor y el resultado final.

### 3.1 Mapa de memoria

A continuación se presenta la tabla de memoria obtenida del autómata de la etiquetadora, donde se muestra la dirección, el tipo, el nombre y la descripción de las variables empleadas en el programa.

#### 3.1.1 Tabla de memoria obtenida del programa.

Tabla 3.1. Tabla de memoria obtenida del programa del PLC.

Dirección	Tipo	Nombre	Descripción y observaciones.
000.01	Entrada	K140	Sensor que detecta la presencia de botella.
000.02	Entrada	B41	Sensor que detecta paso de máquina. Avance de la botella.
000.04	Entrada	B23	Sensor que detecta la presencia de collarín en la botella.
000.06	Entrada	B24	Sensor que detecta la presencia de etiqueta en la botella.
000.08	Entrada	B25	Sensor que detecta la presencia de contraetiqueta en la botella.
001.01	Entrada	S16	Pulsante que resetea los CNT 1,2, y 3 desde el panel de control.
010.00	Salida	OUTMINUSBLOCCO	Parada del sistema o bloque del gato. Esta salida es negada: "1" en marcha, "0" parada.
010.06	Salida	H16MINUS	Lámpara roja intermitente por falta de 3 adhesivos en la botella de forma

			consecutiva.
010.07	Salida	H18MINUS	Alarma sonora por ausencia de adhesivo en la botella, durante 4 segundos intermitente.
TIM000	TIMER	TIM000	<i>Timer</i> que limita a 4 segundos la alarma sonora por falta de adhesivo.
CNT001	COUNTER	CNT1COLLARIN ERROR	Cuenta 3 ausencias de collarín en la botella, sólo si son de forma consecutiva. Da salida para bloqueo de máquina.
CNT002	COUNTER	CNT2ETIQUETA ERROR	Cuenta 3 ausencias de etiqueta, sólo si son de si son de forma consecutiva. Da salida para bloqueo máquina.
CNT003	COUNTER	CNT3CONTRA ETIQUETAERROR	Cuenta 3 ausencias de etiqueta si son de forma consecutiva. Da salida para bloqueo máquina.
200.00	BIT	DIFUK140	Memoria auxiliar 200.00. Se pone en "1" durante 1 scan cuando se detecta presencia de botella.
220.00	BIT	MEMMINUSH18	Memoria auxiliar 220.00. Se pone en "1" durante alarma sonora por ausencia de adhesivo.
253.14	BIT	P_Off	Indicador siempre Off.
255.02	BIT	P_1	Bit de pulsos de reloj de 1.0 segundo.

210	WORD		Control del collarín. Registro desplazamiento, para leer y escribir a cada instante diferentes bits. Ej. 210.05, 210.06, 210.09, etc. (ver más adelante)
212	WORD		Control de la etiqueta. Registro desplazamiento, para leer y escribir a cada instante diferentes bits. Ej. 212.05, 212.06, 212.09. (ver más adelante)
214	WORD		Control de la contraetiqueta. Se emplea como registro desplazamiento, para leer y escribir a cada instante diferentes bits. Ej. 212.07, 212.06, 212.09. (ver más adelante)
210.04	BIT		Este bit es reseteado cada vez que se detecta presencia de collarín.
210.05	BIT		Este bit se pone a "1" cada vez que se desplaza el registro y no se reseteó en 210.04 por no presencia de collarín. Es leído y se toma como una ausencia de collarín.
212.04	BIT		Ídem a 210.04 pero para etiqueta.
212.05	BIT		Ídem a 210.05 pero para etiqueta.
214.06	BIT		Ídem a 210.04 pero para contraetiqueta.

214.07	BIT		Ídem a 210.05 pero para contraetiqueta.
--------	-----	--	---

### 3.2 Propuesta de programación.

Para implementar el sistema supervisor y controlar las variables antes mencionadas es necesario agregar varias líneas al código del programa del autómeta SYSMAC-CPM2A que controla la máquina etiquetadora, ya que no se pueden leer directamente, debido a que el PLC no las trae implementadas.

#### 3.2.1 Desarrollo de la propuesta programación.

En la tabla 2.4 que mostramos a continuación se registra el código propuesto para garantizar conocer las variables que no están disponibles en el código actual almacenado en memoria.

Tabla 3.2 Propuesta de programación.

<b>OBJETIVO A CONOCER</b>	<b>ACCION PROPUESTA</b>
<b>Número de etiquetas puestas correctamente.</b>	LOAD 000.06 y contar.
<b>Número de contraetiquetas puestas correctamente.</b>	LOAD 000.08 y contar.
<b>Número de collarines puestos.</b>	LOAD 000.04 y contar.
<b>Número de botellas que en entran a la máquina.</b>	LOAD 200.00 y contar.

<b>Número de botellas que salen sin etiqueta</b>	LOAD 212.09 y contar.
<b>Número de botellas que salen sin collarín</b>	LOAD 210.09 y contar.
<b>Número de botellas que salen sin contraetiqueta.</b>	LOAD 214.09 y contar.

Los contadores que presenta el autómata en cuestión sólo pueden llegar a contar como máximo 65535 pulsos, y la etiquetadora procesa un promedio de 6000 botellas por hora. Estos contadores se pretenden resetear cuando finalice cada jornada de trabajo (de 8 a 12 horas), por tanto la máquina procesa un promedio de 72000 botellas por jornada, por lo que con un solo contador no se logra contabilizar toda la producción, y se necesita poner dos en serie.

### **3.3 Desarrollo del sistema supervisor.**

Ya una vez realizada la ingeniería inversa al programa del autómata, conociendo así su funcionamiento, y ubicación en las distintas zonas de memoria de todas las variables, y propuestos además los posibles códigos que se deben agregar a dicho programa, sólo resta implementar el sistema supervisor.

Como habíamos mencionado con anterioridad el sistema supervisor lo desarrollaremos con el software de programación de SCADA MOVICON X, cuya apariencia podemos ver en la figura 3.1.

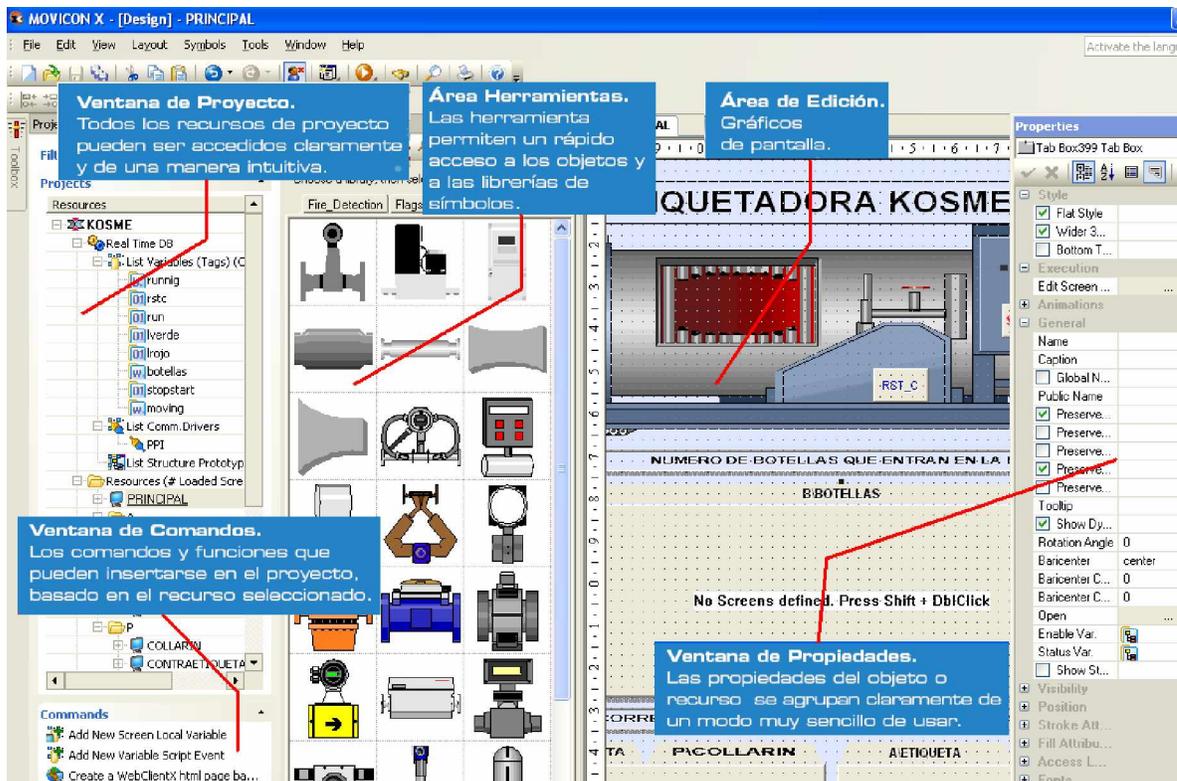


Figura 3.1. Software de programación de SCADA MOVICON X.

Este software posibilita al programador crear una aplicación que le permita al operador supervisar y controlar desde la pantalla de la PC cualquier proceso industrial en tiempo real. Con él se puede realizar un mímico del proceso que se desea supervisar y agregarle animaciones, de forma tal, que represente lo más real posible el proceso que se desea supervisar y/o controlar.

### 3.3.1 Implementación del sistema supervisor.

Para comenzar, se necesita tener instalado el software MOVICON X, para luego pasar a la creación de un proyecto nuevo.

Para la creación del nuevo proyecto es necesario especificar el fabricante con el que se va a trabajar, el driver de comunicación, y el tipo de base de datos a emplear, entre otros requerimientos menos importantes.

Ya una vez creado el proyecto es necesario establecer la comunicación con el PLC (u otro dispositivo a usar). Esto se logra en la Ventana de Proyectos (Figura 3.1), específicamente, en la pestaña *Real Time DB*. En este menú se selecciona el driver y se le añade una

estación. Al añadir esta estación deberá especificarse la velocidad de comunicación y la dirección del hardware con el que se va a establecer la comunicación. Ya establecida la misma se puede pasar a la fase de diseño y creación de variables.

Estas variables son las que van a guardar los valores que se desean conocer del programa que se encuentra “corriendo” en el autómatas. Para su creación podemos dirigirnos hasta la ventana de proyectos, en la pestaña *Real Time DB* y añadir una nueva variable usando el clic derecho del mouse. A la misma se le puede añadir el nombre deseado, preferentemente uno que se corresponda con la función que va a desempeñar, ej. *On\_Off*, *Start*, *Counter* etc. Además de esto se le configura si se va a usar para leer del PLC o para escribir en él, especificándole entonces Input u Output en el caso que corresponda. Si no se va a usar con ninguno de estos fines, y es sólo de uso temporal a nivel de software, se declara de tipo *No Shared*. Una vez creada, en caso que sea de tipo Input u Output, se pasaría a especificar la dirección de donde va a leer o a donde va escribirse el valor de la misma. Esta operación se realiza en el panel de Proyectos, específicamente en *List. Comm.Drivers/PPI, Add New Task*, donde se selecciona dicha variable y se le da la dirección del dispositivo (PLC en nuestro caso). Todas las variables creadas se pondrán a disposición de los diferentes objetos que conformarán el mímico del proceso.

En la ventana de Proyectos, específicamente en el menú *Resource* se añade la ventana o *Screen* principal, la cual mostrará el mímico general del proceso que se desea representar. Para la conformación de este mímico simplemente hay que ir arrastrando los diferentes elementos que sean necesarios del área de herramientas ej. Tanques, motores, tuberías etc. Una vez obtenido el mímico deseado pasamos a conectar los diferentes objetos con sus variables correspondientes. Para esto nos dirigimos a las propiedades del objeto y le asignamos la variable previamente diseñada para manejarlo. (Progea, 2007)

En el presente trabajo, como habíamos visto anteriormente vamos a mostrar el número de etiquetas, contraetiquetas y collarines que se ponen correctamente, el número de botellas que salen sin alguno de estos adhesivos, el número total de botellas que entran en la línea y el estado de la máquina. En este sentido el mímico del proceso de etiquetado quedaría como se muestra a continuación.

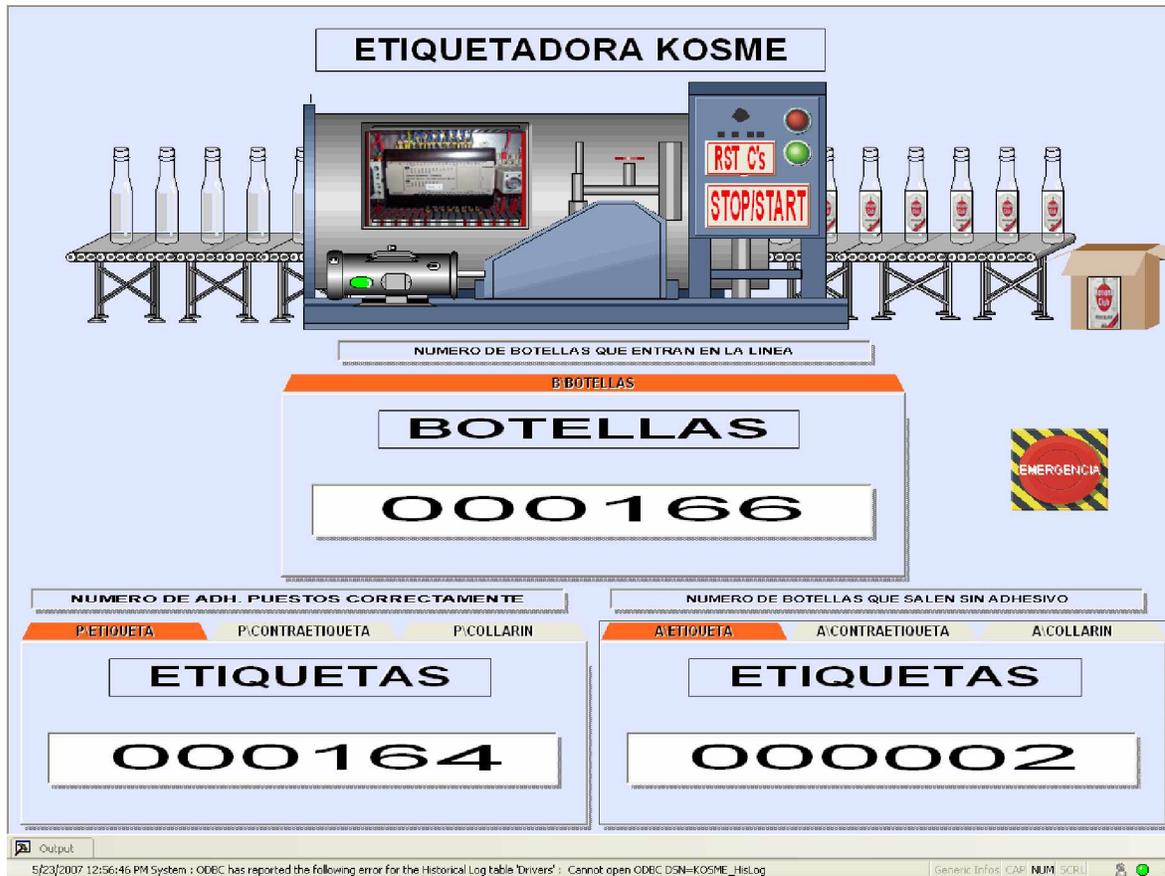


Figura 3.2. Propuesta del mímico del proceso de etiquetado.

### 3.4 Análisis económico.

Un punto fundamental para la realización de un proyecto es la valoración económica del mismo. En este análisis se tendrán en cuenta los gastos por compras de productos, superación del personal y de forma general la relación costo-beneficio de la implementación de este sistema.

Para el desarrollo del proyecto se hace necesaria la compra de los siguientes productos:

Tabla 3.3 Gastos en productos para la inversión inicial.

Producto	Precio (USD)	Cantidad	Valor total USD
Microcomputadora Pentium IV	600.00	1	600.00
Licencia de <i>Run-Time</i> (MOVICON)	1965.00	1	1965.00
Transportación			20.00
Total			2585.00

También es necesario, para superación del personal que se va encargar de la realización del proyecto, invertir un determinado presupuesto inicial. Este debe suplir los gastos en cuanto a cursos básicos de capacitación y superación, para obtener un trabajo con óptima calidad. En la tabla 3.4 se muestra una relación de estos gastos.

Tabla 3.4 Gastos en superación del personal en la inversión inicial.

Nombre	Tipo de postgrado	Cantidad Especialistas	Valor total USD
Autómatas.	Curso	1	70.00
CX-Programmer	Curso	1	130.00
MOVICON-X	Curso	1	200.00
Total			400.00

De acuerdo con lo anteriormente expuesto, la Inversión inicial del proyecto quedaría de la siguiente forma:

$$\begin{aligned} \text{Inversión inicial del proyecto} &= \text{Productos} + \text{Superación} \\ &= 2585.00 + 400.00 = \$ 2985.00 \text{ USD} \end{aligned}$$

A continuación, en la tabla 3.5 se realiza un análisis para una muestra de cinco años, de los flujos de fondo estimados por período.

Tabla 3.5 Flujos de fondo estimados por período (beneficios y costos por períodos)

Año	Costos		Beneficios		Flujos de caja	Flujo acumulado
	Costos	acumulados	Beneficios	acumulados		
0	2985	2985	0	0	-2985	-2985
1	120	3105	3084.134	3084.134	2964.134	-20.866
2	80	3185	450	3534.134	370	349.134
3	100	3285	250	3784.134	150	499.134
4	40	3325	150	3934.134	110	609.134
5	100	3425	200	4134.134	100	709.134
<b>VAN</b>			<b>241 \$</b>	<b>TIR</b>		<b>18%</b>

- **Costos:** Inversión inicial del proyecto y servicio post-venta.
- **Costos acumulados:** Costo más costo acumulado.
- **Beneficios:** Están estimados de manera tal que se evite la pérdida del producto final debido a desperfectos técnicos, errores humanos o ilegalidades. Teniendo en cuenta el costo de producción de una botella lista para la comercialización y la recuperación de diez de estas diarias.
- **Beneficios acumulados:** Beneficios más beneficios acumulados.
- **Flujos de caja:** Beneficios menos costos.
- **Flujo acumulado:** Beneficios acumulados menos costos acumulados.
- **VAN:** Devuelve el valor neto presente en una inversión a partir de una tasa de descuento fija (10%) en un período y una serie de pagos futuros. Función financiera calculada en el Excel.

- **TIR:** Devuelve la tasa interna de retorno de una inversión para una serie de valores en efectivo. Función financiera calculada en el Excel.

Luego, graficando los costos acumulados contra los beneficios acumulados se obtiene el gráfico de retorno de la inversión, el cual se muestra en la figura 3.3.

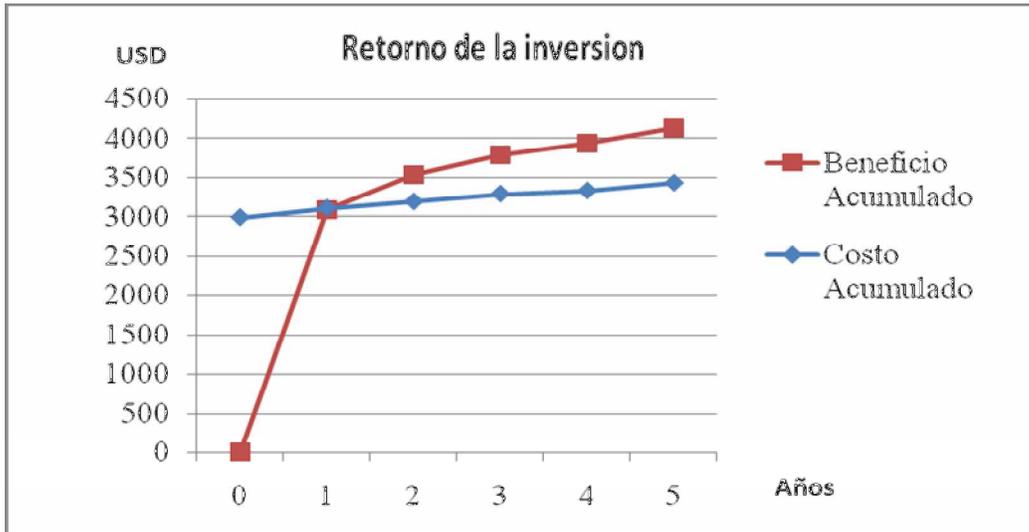


Figura 3.3 Retorno de la inversión.

En la misma se puede verificar que el punto de intersección de las curvas se produce alrededor del año de iniciado el proyecto y para confirmar su veracidad se muestra la siguiente expresión que utiliza los datos de la tabla 3.5:

$$\begin{aligned}
 \text{PRI (período de recuperación de la inversión)} &= N-1 + \text{ABS} (\text{FAN}-1 / \text{FN}) \\
 &= 2 - 1 + \text{ABS} (-20.866/370) \\
 &= 1 + 0.056 \approx 1 \text{ años.}
 \end{aligned}$$

Donde

**N:** Año en el cual se comienza a obtener beneficios.

**ABS:** Valor absoluto.

**FAN:** Fondos acumulados netos.

**FN:** Fondos netos.

También es necesario realizar una comparación entre los gastos con y sin el nuevo sistema incorporado y esto se representa en la tabla 3.6. Los valores reflejados en la misma tienen en cuenta que, con la nueva aplicación, dada las facilidades que brinda, es posible prescindir de uno de los tres operarios que trabajan actualmente en la línea. Los mismos devengan un salario de cuatrocientos veinticinco pesos en moneda nacional y esto representa en una hora alrededor de 0.0885 dólar y que por jornadas de ocho horas se iban a recuperar diez botellas por lo que en una hora se ahorran un promedio de 1.25 de estas, conociendo que el valor del costo de producción de una botella es de un dólar.

Tabla 3.6 Comparación entre los costos con y sin el nuevo sistema, para una hora de producción.

<b>Indicador</b>	<b>Costo estimado actual USD</b>	<b>Costo Estimado con el sistema USD</b>
Producción Final	6000.000	5998.750
Salarios	0.266	0.177
Total	6000.266	5998.927

También es factible conocer a partir de qué volumen de producción este sistema comienza a ofrecer beneficios y por esto se realiza la comparación entre el costo estimado actual y el costo estimado con el sistema añadido, lo cual se muestra en las ganancias de la tabla 3.7.

Tabla 3.7 Horas de producción para la recuperación de la inversión.

<b>Cantidad de Horas de Producción</b>	<b>Costo Estimado Actual USD</b>	<b>Costo Estimado con el Sistema USD</b>	<b>Ganancias USD</b>
8	48002.125	47991.416	10.709
40	240010.624	239957.080	53.544
96	576025.498	575896.992	128.506
1152	6912305.971	6910763.904	1542.067
2320	13920616.192	13917510.640	3105.552

El volumen de producción requerido para recuperar la inversión es de dos mil trescientas veinte horas lo que representa un poco más de un años laboral (8 horas \* 24 días\* 12 meses = 2304 horas).

Y también se presenta una grafica de ganancias contra horas de producción utilizando datos de la tabla 3.7.

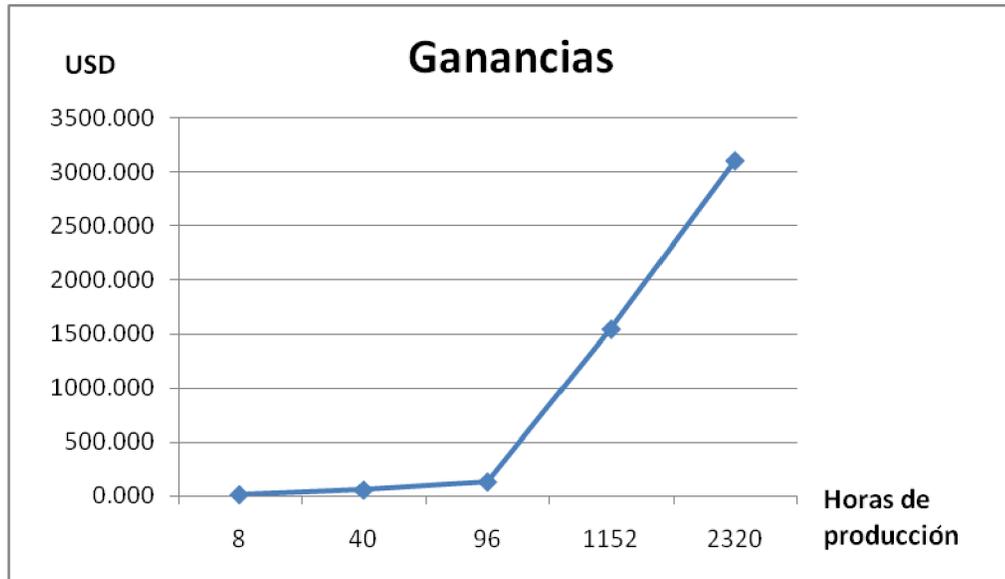


Figura 3.4. Estimado de ganancias por hora de producción.

Por lo tanto, con el análisis realizado, se demuestra como en un corto período de tiempo (aproximadamente un año) la inversión inicial es recuperada y se comienza rápidamente a obtener ganancias, garantizando mayor seguridad para los trabajadores, un aumento del control de los diferentes insumos, lo cual muestra que este proyecto es factible y su implementación es ampliamente beneficiosa para esta línea de producción y para el desarrollo económico del país.

## CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

Los siguientes puntos reflejan de forma precisa el cumplimiento de los objetivos trazados:

- Se logró obtener de forma íntegra la distribución de las variables de E/S y variables internas empleadas por el PLC en el control de la máquina etiquetadora.
- Se añadieron nuevos segmentos de códigos al programa de usuario del autómeta, para obtener información de variables que no estaban disponibles en el mismo.
- Se implementó exitosamente la aplicación SCADA, donde se mostraron las principales variables del autómeta que controla la etiquetadora.
- Se comprobó el correcto funcionamiento del sistema supervisor realizado, pues se conectó, a nivel de laboratorio, a un PLC de la firma Siemens, lográndose los objetivos de supervisión y control.
- Se mostró el amplio campo de aplicación que tienen los PLC's en la industria nacional, y las potencialidades que presenta el software de programación de SCADA MOVICON-X.

**Recomendaciones**

Se recomienda, para dar continuidad a este trabajo, los siguientes aspectos:

- Hacer las pruebas de conexión del SCADA con el autómata real que está montado en la máquina etiquetadora, redireccionando los diferentes objetos empleados en el supervisor, con las direcciones reales del PLC de la máquina en cuestión.
- Incluir en el sistema supervisor gráficas de reportes históricos.
- Gestionar la posibilidad de ampliar la enseñanza de SCADA en nuestra Facultad.

## REFERENCIAS BIBLIOGRÁFICAS

- Ambrose, A. G. (2004). Sistema de supervisión y control del pasteurizador en la cervecería "Antonio Díaz Santana", Santa Clara, Cuba, UCLV.
- Bayón, R. M. (2005). "Sistemas domóticos basados en PLC's." Accedido, Disponible.
- Canto, C. E. (2006). "Autómatas Programables." Accedido el 5 de Marzo, 2007, Disponible en: <http://gpds.uv.es/plc/plc.pdf>.
- Castellanos, E. I. (2007). Sistemas a base de PLC.
- Diéguez, J. J. R. (2005). "La evolución del control de procesos y la telemetría." Boletín Electrónico No. 02(Facultad de Ingeniería. Universidad Rafael Landívar.).
- Guille, M. (2001). Autómatas Programables: arquitectura y aplicaciones. Barcelona, España.
- Montejo, L. (2005). "Automatización de Planta Productora de Hormigón para Elementos Prefabricados." SIE'2005: 1-3.
- OMRON. (2004). "CX-Programmer Introduction Guide." Accedido el 15 de Febrero, 2007, Disponible en: <http://downloadcentre.omron-industrial.com/dlc3/files/Products/CX-One/Programming/CX-Programmer/R120/R120-ES2-01+CX-Progr+IntroGuide.pdf>.
- OMRON. (2004). "SYSMAC CPM1A/CPM2A/CPM2C Programming Manual." Accedido el 10 de Febrero, 2007, Disponible en: <http://downloadcentre.omron-industrial.com/dlc3/files/IAB/Products/Automation%20Systems/Programmable%20Logic%20Controllers/Compact%20PLC%20Series/CPM1A/W353/W353-E1-05.pdf>.
- Progea, C. (2006). "Visualization and Control for Embedded System Based on Windows CE." Accedido el 18 de Mayo, 2007, Disponible en: [http://www.movicon.info/docum/SPA/MoviconX1CE\\_ESP.pdf](http://www.movicon.info/docum/SPA/MoviconX1CE_ESP.pdf).
- Progea, C. (2007). "Movicon Programmer Guide." Accedido el 15 de Mayo, 2007, Disponible en: [http://www.movicon.info/download/eng/Movicon\\_Programmer\\_Guide\\_955.pdf](http://www.movicon.info/download/eng/Movicon_Programmer_Guide_955.pdf).
- Progea, C. (2007). "Products Online." Accedido el 2 de Marzo, 2007, Disponible en: [www.progea.com](http://www.progea.com).

- Sánchez, J. P. (2004). "Aplicaciones de los Controladores Lógicos Programables." Accedido el 20 de Marzo, 2007, Disponible en: <http://www2.eie.ucr.ac.cr/~valfaro/docs/monografias/0401/ucr.ie431.trabajo.2004.02.grupo01.pdf>.
- SIEMENS. (2005). "WinCC Flexible. Communication Part One." Accedido el 24 de Mayo, 2007, Disponible en: [http://www1.siemens.cz/ad/current/layers/data/downloads/c004hmi/wincc\\_flex/wincc\\_flex\\_komunikace/UM\\_WinCCflexible2005Com2\\_2005\\_en.pdf](http://www1.siemens.cz/ad/current/layers/data/downloads/c004hmi/wincc_flex/wincc_flex_komunikace/UM_WinCCflexible2005Com2_2005_en.pdf).
- Vidal, P. (2002). "OPC: Un estándar en las redes industriales y buses de campo." Accedido el 20 de Mayo, 2007, Disponible en: <http://www.senacitel.cl/downloads/senacitel2002/te001.pdf>.

## BIBLIOGRAFIA

Balcells, J. (1997). Autómatas Programables. Barcelona, España.

Castillo, J. M. (2003). "Autómatas programables." Accedido el 20 de Marzo 2007, Disponible en: <http://www.olmo.pntic.mec.es>.

Erickson, K. (1996). "Programmable logic controllers." IEEE.

Hellgren. A. (2002). "Modelling and PLC-based Implementation of Modular Supervisory Control." IEEE: 371-376.

Mayol, A. (1998). Autómatas Programables. Barcelona, España.

Porras, A. (1994). Autómatas programables: fundamentos, manejo, instalación y prácticas.

Salder, A. G. (2003). Introducción a los Autómatas Programables.

## ANEXOS

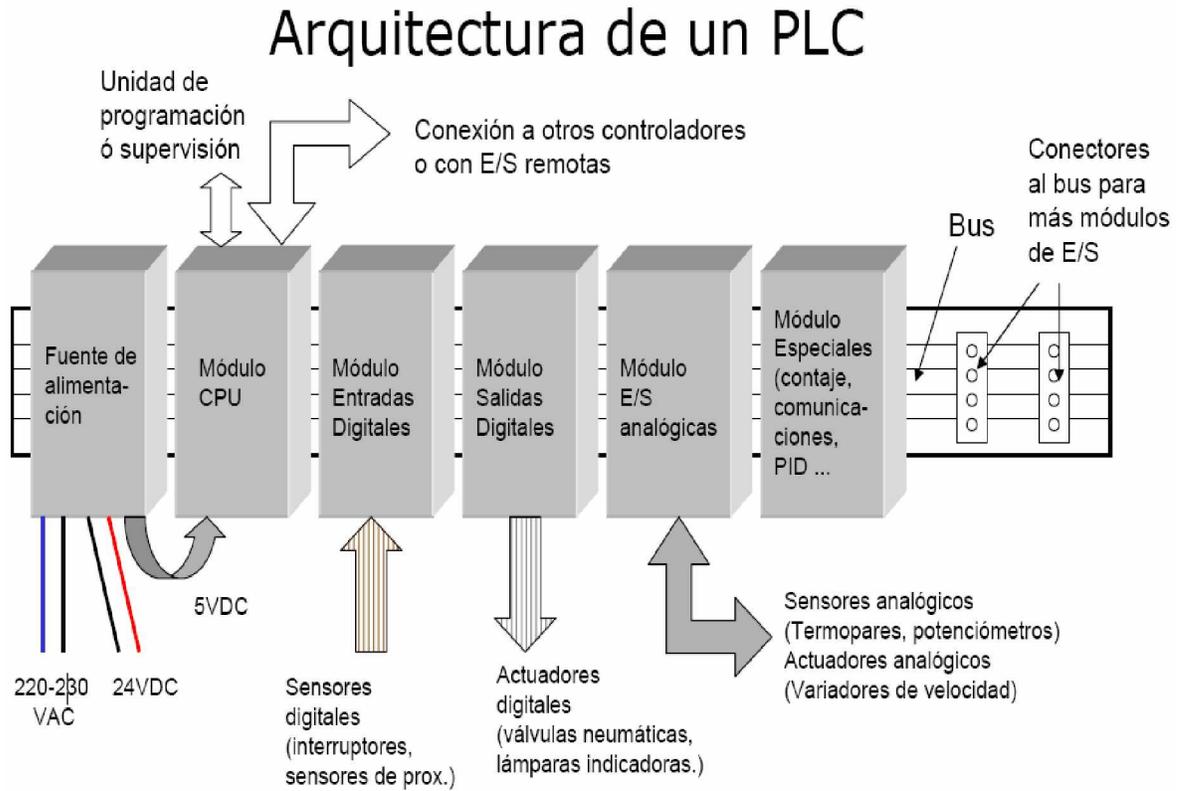
### Anexo I Características del A.P SYSMAC- CPM2A de OMRON.

Item		Especificaciones			
Método de control		Método de programa almacenado			
Método de control de E/S		Scan cíclico con salida directa (se puede efectuar refresco inmediato con IORF(97).)			
Lenguaje de programación		Diagrama de relés			
Longitud de instrucción		1 paso por instrucción, 1 a 5 palabras por instrucción			
Instrucciones		Instrucciones básicas: 14 Instrucciones especiales: 105 instrucciones, 185 variaciones			
Tiempo de ejecución		Instrucciones básicas: 0.64 $\mu$ s (instrucción LD) Instrucciones especiales: 7.8 $\mu$ s (instrucción MOV)			
Capacidad de programa		4,096 palabras			
Capacidad de E/S máx.	Sólo CPU	20 puntos	30 puntos	40 puntos	60 puntos
	Con unidades de expansión de E/S	80 puntos máx.	90 puntos máx.	100 puntos máx.	120 puntos máx.
Bits de entrada		IR 00000 a IR 00915 (Los canales no utilizados para bits de entrada se pueden utilizar como bits de trabajo)			
Bits de salida		IR 01000 a IR 01915 (Los canales no utilizados para bits de salida se pueden utilizar como bits de trabajo)			
Bits de trabajo		928 bits: IR 02000 a IR 04915 y IR 20000 a IR 22715			
Bits especiales (área SR)		448 bits: SR 22800 a SR 25515			
Bits temporales (área TR)		8 bits (TR0 a TR7)			
Bits de retención (área HR)		320 bits: HR 0000 a HR 1915 (Canales HR 00 a HR 19)			
Bits auxiliares (área AR)		384 bits: AR 0000 a AR 2315 (Canales AR 00 a AR 23)			
Bits de enlace (área LR)		256 bits: LR 0000 a LR 1515 (Canales LR 00 a LR 15)			
Temporizadores/ Contadores		256 temporizadores/contadores (TIM/CNT 000 a TIM/CNT 255)  Temporizadores de 1-ms: TMHH(--) Temporizadores de 10-ms: TIMH(15) Temporizadores de 100-ms: TIM Temporizadores de 1-s/10-s : TIML(--) Contadores descendentes: CNT Contadores reversibles: CNTR(12)			

**Anexo II Comparación del A.P CPM2A con el CPM2C de OMRON.**

	Item	CPM2C	CPM2A
Protección de memoria	Área de programa, área DM de sólo lectura	Protección de memoria Flash	Igual que CPM2C.
	Área DM de lectura/escritura, área de HR, área AR y contadores	CPU con reloj: Batería interna (2 años a 25°C, se puede cambiar)  CPU sin reloj: Condensador (10 días a 25°C) o batería opcional (5 años a 25°C, se puede cambiar)	Batería interna (5 años a 25°C, se puede cambiar)
Entradas de interrupción (modo de entrada de interrupción)		4 (CPU de 20 puntos), 2 (CPU de 10 puntos)	4
Entradas de interrupción (modo contador)	Modo contador	Contador incremental Contador descendente	Igual que CPM2C.
	Límite superior de contador	2 kHz	Igual que CPM2C.
	SR 244 a SR 247	Contiene el PV del contador.	Igual que CPM2C.
	Método(s) para leer el PV del contador	Lectura de SR 244 a SR 247. Ejecutar PRV(62).	Igual que CPM2C.
	Método para cambiar el PV del contador	Ejecutar INI(61).	Igual que CPM2C.
Temporizador de intervalo	Modo de un impulso	Sí	Igual que CPM2C.
	Modo de interrupción programada	Sí	Igual que CPM2C.

## Anexo II. Arquitectura de un PLC.



**Anexo III Programa del autómata SYSMAC-CPM2A.**

DIFU-B40. PRESENCIA DE BOTELLA.

1.LD K140

2.DIFU(13) DIFUK140

DIFU-B41. MEM.AUX.20001 DIFU B41 SENSOR DE PASO.

3.LD B41

4.DIFU(13) DIFUB41

SET SFT. CONTROL DE COLLAR., ETIQ., Y CONTRAE., EN LA BOT.

5.LD DIFUK140

6.SET 210.00

7.SET 212.00

8.SET 214.00

SFT CONTROL DEL COLLARIN EN LA BOTELLA.

9.LD P\_Off

10. LD DIFUB41

11. LD P\_Off

12. SFT(10) 210 211

SFT-CONTROL DE LA ETIQ.

13. LD P\_Off

14. LD DIFUB41

15. LD P\_Off

16. SFT(10) 212 213

SFT- CONTROL DE LA CONTRAETIQUETA.

17. LD P\_Off

18. LD DIFUB41

19. LD P\_Off

20. SFT(10) 214 215

RESET BIT 210.04 SFT CONTROL DEL SELLO EN LA BOTRLLA.

21. LD B23

22. RSET 210.04

RESET BIT 212.04 SFT CONTROL DE LA ETIQ. EN LA BOT.

23. LD B24

24. RSET 212.04

RESET BIT 214.06 SFT CONTROL DE LA CONTRAE. EN LA BOT.

25. LD B25

26. RSET 214.06

H18 ALARMA SONORA POR FALTA DE COLL, ETIQ., O CONTRAE. EN LA BOT.

27. LD 210.09

28. OR 212.09

29. OR 214.09

30. ANDNOT B41

31. LD RSTMINUSH18\_Estado

32. KEEP(11) MEMMINUSH18

H18-ALARMA SONORA PARA CONTROL DE COLLARIN, ETIQUETA Y CONTRAETIQUETA EN LA BOTELLA.

33. LD MEMMINUSH18

34. AND P\_1s

35. OUT H18MINUS

RST-H18-TIM0-TIM0 RESET ALARMA SONORA H18

36. LD MEMMINUSH18

37. TIM 000 #0040

CNT1-COLLAR-CNT1 CONTEO DE AUSENCIA CONSECUTIVA DE COLLARIN EN LA Y PARADA DEL SISTEMA.

38. LD 210.05

39. ANDNOT B41

40. LD S16

41. LD B23

42. ANDNOT CNT1MINUSBOLLO\_Estado

43. ORLD

44. CNT 001 #0003

CNT2-ETICH.-CNT2 CONTEO CONSECUTIVO DE CARENCIA DE ETIQ. Y PARADA DEL SISTEMA

45. LD 212.05

46. ANDNOT B41

47. LD S16

48. LD B24

49. ANDNOT CNT2MINUSETICHMINUS\_Estado

50. ORLD

51. CNT 002 #0003

CNT3-RETROETIC-CNT3 CONTEO DE AUSENCIA DE CONTRAETIQUETAS CONSECUTIVAS Y PARADA DEL SISTEMA CHIUSURA BLOCCO BOTT.

52. LD 214.07

53. ANDNOT B41

54. LD S16

55. LD B25

56. ANDNOT CNT3MINUSRETROEMINUS\_Estado

57. ORLD

58. CNT 003 #0003

H16-H16 LUZ ROJA POR FALTA DE ETIQUETAS CONSECUTIVAS.

59. LD CNT1MINUSBOLLO\_Estado

60. OR CNT2MINUSETICHMINUS\_Estado

61. OR CNT3MINUSRETROEMINUS\_Estado

62. OUT TR0

63. AND P\_1s

64. OUT H16MINUS

65. LD TR0

66. OUTNOT OUTMINUSBLOCCO

END PROGRAM

67. END(01)

## Anexo IV Software de programación de PLC CX-Programmer V3.0.

