



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

---

---

FACULTAD DE INGENIERÍA

APLICACIÓN DE UN PLC AL CONTROL DE  
UNA BANDA TRANSPORTADORA

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECÁNICO ELECTRICISTA  
AREA : ELÉCTRICA ELECTRÓNICA

PRESENTAN:

LEOBARDO CANO REYES  
MARCO ANTONIO SERVÍN CASTILLO  
JESÚS MARIA FRANCISCO HERNÁNDEZ MORALES



Director de tesis:

Ing. Francisco J. Rodríguez Ramírez

CIUDAD UNIVERSITARIA, MÉXICO D.F. 2003

## Tabla de contenido

<b>TABLA DE CONTENIDO</b> .....	<b>I</b>
<b>INTRODUCCIÓN</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>CAPÍTULO 1 PRINCIPIOS DE OPERACIÓN</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
DEFINICIÓN DE UN CONTROLADOR LÓGICO PROGRAMABLE (PLC) .....	¡ERROR! MARCADOR NO DEFINIDO.
BREVE SINOPSIS DEL PLC .....	¡ERROR! MARCADOR NO DEFINIDO.
ARQUITECTURA TÍPICA DE UN PLC .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Módulos de entrada y salida.</b> .....	¡Error! Marcador no definido.
<b>Unidad central de proceso</b> .....	¡Error! Marcador no definido.
<b>Memoria de programa</b> .....	¡Error! Marcador no definido.
<b>Bus de datos</b> .....	¡Error! Marcador no definido.
<b>La fuente de alimentación</b> .....	¡Error! Marcador no definido.
VENTAJAS DE LOS PLC .....	¡ERROR! MARCADOR NO DEFINIDO.
DESVENTAJAS DE LOS PLC .....	¡ERROR! MARCADOR NO DEFINIDO.
OPERACIÓN DEL PLC.....	¡ERROR! MARCADOR NO DEFINIDO.
ELEMENTOS DE HARDWARE DEL SIMATIC S5 100-U .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Características generales</b> .....	¡Error! Marcador no definido.
<b>Bloques Funcionales</b> .....	¡Error! Marcador no definido.
<b>Sistemas de Entradas y Salidas</b> .....	¡Error! Marcador no definido.
<b>MÓDULOS DE ENTRADA Y SALIDA DIGITAL</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
Módulo de entrada digital 8 x 24VDC .....	¡Error! Marcador no definido.
Módulo de salida digital 4 x 24VDC /2A .....	¡Error! Marcador no definido.
Módulos de entrada y salida analógicas .....	¡Error! Marcador no definido.
Módulo de entrada analógica 4 x + 50mV.....	¡Error! Marcador no definido.
Módulo de salida analógica 2 x 1 5V .....	¡Error! Marcador no definido.
<b>MÓDULOS ESPECIALES</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
Módulo de temporizadores .....	¡Error! Marcador no definido.
Módulo de contadores .....	¡Error! Marcador no definido.
Módulo de comparadores .....	¡Error! Marcador no definido.
Módulos simuladores de entradas ó salidas digitales .....	¡Error! Marcador no definido.
Módulos de memoria .....	¡Error! Marcador no definido.
El Bus Simatic.....	¡Error! Marcador no definido.
<b>CONSTRUCCIÓN DEL BUS EN EL S5-100U</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
Concepto de dirección .....	¡Error! Marcador no definido.
Guías para la conexión de elementos de entrada y salida .....	¡Error! Marcador no definido.
<b>CAPÍTULO 2 PRINCIPIO DE PROGRAMACIÓN DE CONTROLADORES LÓGICOS PROGRAMABLES.</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Funciones lógicas</b> .....	¡Error! Marcador no definido.
<b>LA FUNCIÓN AND (Y Ó PRODUCTO BOOLEANO)</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>LA FUNCIÓN OR (O Ó TAMBIÉN SUMA BOOLEANA)</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>LA FUNCIÓN OR (O Ó TAMBIÉN SUMA BOOLEANA)</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>LA FUNCIÓN NOT (NEGACIÓN)</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
PRINCIPIOS DE LÓGICA Y ÁLGEBRA BOOLEANA.....	¡ERROR! MARCADOR NO DEFINIDO.
<b>OPERACIONES BÁSICAS.</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
Compuertas básicas .....	¡Error! Marcador no definido.
Compuertas combinadas .....	¡Error! Marcador no definido.
<b>REGLAS DEL ÁLGEBRA BOOLEANA</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
Ley de conmutatividad .....	¡Error! Marcador no definido.
Ley de asociatividad.....	¡Error! Marcador no definido.
Ley de distributividad .....	¡Error! Marcador no definido.
Ley de absorción.....	¡Error! Marcador no definido.
Leyes de DeMorgan .....	¡Error! Marcador no definido.
<b>PROCEDENCIA DE LOS OPERADORES Y AGRUPACIÓN DE SIGNOS.</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
El concepto de señales binarias.....	¡Error! Marcador no definido.
Simbología de contactos .....	¡Error! Marcador no definido.
<b>CONTACTOS NORMALMENTE ABIERTOS</b> .....	¡ERROR! MARCADOR NO DEFINIDO.

<b>CONTACTOS NORMALMENTE CERRADOS</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Lenguajes de programación</b> .....	¡Error! Marcador no definido.
Lenguaje STEP 5 .....	¡ERROR! MARCADOR NO DEFINIDO.
TIPOS DE REPRESENTACIÓN. ....	¡ERROR! MARCADOR NO DEFINIDO.
DEFINICIONES .....	¡ERROR! MARCADOR NO DEFINIDO.
Ciclo de Barrido .....	¡Error! Marcador no definido.
Instrucción .....	¡Error! Marcador no definido.
Estructura de Bit, Byte y Word.....	¡Error! Marcador no definido.
OPERACIONES BÁSICAS .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Combinaciones</b> .....	¡Error! Marcador no definido.
<b>OPERACIÓN A (AND)</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>OPERACIÓN AN (NAND)</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>OPERACIÓN O (OR)</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>OPERACIÓN ON (NOR)</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Operaciones de Memoria</b> .....	¡Error! Marcador no definido.
<b>Operaciones con Temporizadores</b> .....	¡Error! Marcador no definido.
<b>SP TIPO PULSO</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>SE TIPO PULSO ALARGADO</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>SD RETARDO DE PULSO AL ENCENDIDO</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>SS TIPO RETARDO CON MEMORIA</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>SF TIPO DESCONEXIÓN CON RETARDO</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Operaciones con contadores</b> .....	¡Error! Marcador no definido.
OPERACIONES DIGITALES CON PALABRAS .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Operaciones de carga y transferencia</b> .....	¡Error! Marcador no definido.
<b>Combinaciones digitales</b> .....	¡Error! Marcador no definido.
<b>Operaciones aritméticas</b> .....	¡Error! Marcador no definido.
<b>Operaciones complementarias</b> .....	¡Error! Marcador no definido.
<b>OPERACIONES DE TRANSFORMACIÓN</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>OPERACIONES DE DESPLAZAMIENTO</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>OPERACIONES DE SALTO</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
ORGANIZACIÓN DE UN PROGRAMA EN STEP 5 .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Tipos de programación</b> .....	¡Error! Marcador no definido.
<b>PROGRAMAS LINEALES</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>PROGRAMAS ESTRUCTURADOS</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Tipos de módulos de programa</b> .....	¡Error! Marcador no definido.
<b>MÓDULOS DE ORGANIZACIÓN (OB)</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
Ejecución cíclica del programa usando el OB 1.....	¡Error! Marcador no definido.
Módulos de organización para arranque: OB 21 y OB 22.....	¡Error! Marcador no definido.
Fallo de batería. usando el OB 34 .....	¡Error! Marcador no definido.
Perro guardián empleando el OB 31.....	¡Error! Marcador no definido.
<b>MÓDULOS DE PROGRAMA (PB)</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>MÓDULOS DE PASO (SB)</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>MÓDULOS FUNCIONALES (FB)</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Módulos funcionales integrados</b> .....	¡Error! Marcador no definido.
<b>CONVERSOR DE CÓDIGO B4</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>CONVERTIDOR DE CÓDIGO 16</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>MULTIPLICADOR 16</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>DIVISOR 16</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>LECTURA Y NORMALIZACIÓN DE UN VALOR ANALÓGICO</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>SALIDA DE UN VALOR ANALÓGICO</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>MÓDULOS DE DATOS</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
DOCUMENTACIÓN DE PROGRAMAS .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>CAPÍTULO 3 PRESENTACIÓN DE LA APLICACIÓN</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
PLANTEAMIENTO DEL PROBLEMA .....	¡ERROR! MARCADOR NO DEFINIDO.
DESARROLLO DEL PROTOTIPO .....	¡ERROR! MARCADOR NO DEFINIDO.
Bosquejo del prototipo .....	¡Error! Marcador no definido.
PARTES INVOLUCRADAS EN SU FUNCIONAMIENTO .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Dispositivos mecánicos</b> .....	¡Error! Marcador no definido.
<b>Módulos de control</b> .....	¡Error! Marcador no definido.
<b>Dispositivos de monitoreo</b> .....	¡Error! Marcador no definido.
<b>Dépósitos de entrada y salida</b> .....	¡Error! Marcador no definido.

<b>Depósito de desvío</b> .....	¡Error! Marcador no definido.
<b>Proceso de llenado</b> .....	¡Error! Marcador no definido.
DESCRIPCIÓN DEL PROCESO.....	¡ERROR! MARCADOR NO DEFINIDO.
MÓDULO DE CONTROL.....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Tarjetas de control</b> .....	¡Error! Marcador no definido.
<b>El VCO</b> .....	¡Error! Marcador no definido.
<b>Tarjetas de potencia</b> .....	¡Error! Marcador no definido.
MOTORES DE PASOS.....	¡ERROR! MARCADOR NO DEFINIDO.
Construcción.....	¡Error! Marcador no definido.
<b>Funcionamiento de un motor de pasos</b> .....	¡Error! Marcador no definido.
<b>Disposición de las bobinas</b> .....	¡Error! Marcador no definido.
<b>Motores bipolares</b> .....	¡Error! Marcador no definido.
<b>Diferentes tipo de excitación de las bobinas</b> .....	¡Error! Marcador no definido.
<b>SECUENCIA NORMAL</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>SECUENCIA TIPO WAVE DRIVE</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>SECUENCIA DE MEDIO PASO</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>Secuencia para motores bipolares</b> .....	¡Error! Marcador no definido.
CONTROL DE LOS MOTORES DE DESPLAZAMIENTO.....	¡ERROR! MARCADOR NO DEFINIDO.
<b>CLASIFICACIÓN DE LOS CONTROLES</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
Tipo i.....	¡Error! Marcador no definido.
<b>Tipo ii</b> .....	¡Error! Marcador no definido.
<b>Tipo iii</b> .....	¡Error! Marcador no definido.
<b>Tipo iv</b> .....	¡Error! Marcador no definido.
<b>Tipo v</b> .....	¡Error! Marcador no definido.
DETECTORES DE BOTELLAS.....	¡ERROR! MARCADOR NO DEFINIDO.
CONTROL MAESTRO DE ARRANQUE Y PARO.....	¡ERROR! MARCADOR NO DEFINIDO.
<b>CAPÍTULO 4 PROGRAMACIÓN DEL PLC</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
ENTRADAS Y SALIDAS DEL PLC.....	¡ERROR! MARCADOR NO DEFINIDO.
BANDA TRANSPORTADORA.....	¡ERROR! MARCADOR NO DEFINIDO.
DESPACHADOR DE BOTELLAS.....	¡ERROR! MARCADOR NO DEFINIDO.
DESVIADOR DE BOTELLAS.....	¡ERROR! MARCADOR NO DEFINIDO.
LLENADO DE BOTELLAS.....	¡ERROR! MARCADOR NO DEFINIDO.
RETIRADOR.....	¡ERROR! MARCADOR NO DEFINIDO.
<b>CAPÍTULO 5 CONCLUSIONES</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>EVALUACIÓN DE LA CONSTRUCCIÓN DEL PROTOTIPO</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
CONCLUSIONES.....	¡ERROR! MARCADOR NO DEFINIDO.
<b>BIBLIOGRAFÍA</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>APENDICE</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>EL LENGUAJE DE REPRESENTACIÓN DE UN PROGRAMA EN STEP5, TIENE TRES POSIBLES FORMAS DE REPRESENTACIÓN:</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
INSTRUCCIONES BÁSICAS EN STEP5.....	¡ERROR! MARCADOR NO DEFINIDO.
PROGRAMADOR.....	¡ERROR! MARCADOR NO DEFINIDO.
ACCESO A STEP 5.....	¡ERROR! MARCADOR NO DEFINIDO.
EDICIÓN DE UN PROGRAMA NUEVO: MODO INPUT.....	¡ERROR! MARCADOR NO DEFINIDO.
EDICIÓN EN REPRESENTACIÓN CSF.....	¡ERROR! MARCADOR NO DEFINIDO.
EDICIÓN EN REPRESENTACIÓN LAD.....	¡ERROR! MARCADOR NO DEFINIDO.
EDICIÓN EN REPRESENTACIÓN STL.....	¡ERROR! MARCADOR NO DEFINIDO.
LISTAR PROGRAMA YA CREADO: MODO OUTPUT.....	¡ERROR! MARCADOR NO DEFINIDO.
INSERCIÓN Y BORRADO DE SEGMENTOS: MODO INSERT Y DELETE.....	¡ERROR! MARCADOR NO DEFINIDO.
CORRECCIÓN DE UN SEGMENTO: MODO CORRECT.....	¡ERROR! MARCADOR NO DEFINIDO.
TRANSFERENCIA DE UN PROGRAMA: MODO TRANSFER.....	¡ERROR! MARCADOR NO DEFINIDO.
PRUEBA DE U PROGRAMA: MODO TEST/STATUS.....	¡ERROR! MARCADOR NO DEFINIDO.
TIPO DE OPERACIONES.....	¡ERROR! MARCADOR NO DEFINIDO.
OPERACIONES COMPLEMENTARIAS.....	¡ERROR! MARCADOR NO DEFINIDO.
OPERACIONES DE SISTEMA.....	¡ERROR! MARCADOR NO DEFINIDO.
OPERACIONES BÁSICAS.....	¡ERROR! MARCADOR NO DEFINIDO.

# Introducción

La necesidad de realizar medios de fabricación más eficientes y productivos, requiere el uso de técnicos y sistemas capaces de ser modificables con el propósito de manipular cambios en el proceso al menor tiempo posible a un bajo costo. Con la creación y evolución de las computadoras aplicadas al control y el desarrollo de las comunicaciones, nace un nuevo control automático, el Controlador Lógico Programable (Programmable Logic Controller (PLC)). El cual cumple con las necesidades de control que son cada vez más exigentes y en donde se requieren altas velocidades de procesamiento, consumo bajo de energía eléctrica, poco espacio de instalación y un bajo costo.

La diferencia entre una computadora y un PLC refiriéndonos al ambiente industrial, es el ser capaz de tolerar las circunstancias de un medio hostil, por tratarse del uso de maquinaria tales como motores y sistemas mecánicos, que ocasionan un medio con presencia de vibraciones, polvo interferencias electromagnéticas, altas temperaturas, condiciones extremas de humedad, etc. por esta razón los PLC están fabricados de acuerdo a características de tipo industrial tales como:

- Opto acoplamientos para módulos de entrada y salida así como para los dispositivos periféricos que estén unidos al control.
- Fuentes de poder especialmente construidas para soportar variaciones bruscas en tensión.
- Sistemas continuos de almacenamiento de datos que permiten al sistema no perder información.
- Los circuitos impresos deben tener un recubrimiento especial para evitar la humedad y el polvo del medio ambiente, también estarán diseñados para soportar altas temperaturas aún en condiciones extremas de trabajo.

Para conocer las características del PLC, nos hemos dado a la tarea de mostrar su funcionamiento con la puesta en marcha de un proceso controlado por lo que el desarrollo de la presente tesis es con base a una estructuración que se considera necesaria para dar a conocer tanto las características como la forma de aprender a utilizar de manera práctica un PLC; en este caso contamos con el SIMATIC S5-100U de la marca SIEMENS. El empleo de este PLC nos permite realizar el control de un proceso industrial real, este consiste en un proceso de llenado de botellas contemplando a diversas secciones durante el recorrido sobre una banda transportadora, estas secciones se detallaran explícitamente en su momento. El propósito de la tesis no es más que el plantear un problema y ofrecer una solución, ya que nos acercamos lo más posible a un proceso de llenado como el utilizado en la industria y que en un momento dado resulta ser solo demostrativo con opción a crecer.

Siguiendo una metodología para empezar a enfocarnos al tema de los PLC se desarrolla la siguiente temática: en primer lugar con el Capítulo 1 conoceremos el principio de operación del PLC SIMATIC S5-100U, como referencia se abordan los conceptos necesarios como son la definición y una breve sinopsis, arquitectura, ventajas y desventajas, y la operación del PLC, así como también conocer los elementos del hardware; el Capítulo 2 nos enfoca directamente al principio de la programación, se habla de las funciones lógicas necesarias para la utilización de la programación, los diferentes tipos de lenguajes de programación STEP 5, operaciones básicas, operaciones digitales con palabras, organización de un programa, programas lineales y estructurados; posteriormente en el Capítulo 3 se hace la presentación de la aplicación o del proceso de control que se eligió para ejemplificar de manera real la utilidad de un PLC, la forma de cómo se llegó a determinar y desarrollar cada una de sus secciones, los elementos que las constituyen, los diferentes tipos de controles adicionales al proceso, en que consiste cada uno y su programa de ejecución de control en conjunto; el Capítulo 4 esta dedicado a la programación del PLC aplicado al proceso de control, refiriéndonos a cada una de sus secciones por separado para que resulte más sencillo visualizar el procedimiento empleado en el desarrollo de cada etapa (sub programa), y al final mostrar el programa completo tal cual como es y como se ejecuta realmente; por último en el Capítulo 5 se hacen los comentarios acerca de los diferentes tipos de problemas que se presentaron y las soluciones a las que se fueron tomando, además se mencionan las conclusiones relevantes a las que finalmente se llegó.

# Capítulo 1 Principios de operación

El avance en el control de procesos industriales, siempre va a la par del desarrollo tecnológico, de esta forma podemos observar que la historia de los controladores empieza con la era de las máquinas de vapor y fue avanzando hasta que llegó a la época de los controles por relevadores, los cuales eran muy confiables, sin embargo resultaron ser sistemas poco flexibles. Con el nacimiento de las computadoras el desarrollo de técnicas de control tuvo un avance significativo tanto en la tecnología empleada como en la flexibilidad y otras prestaciones de controladores como los PLC.

## Definición de un controlador lógico programable (PLC)

Un controlador lógico programable (PLC, del inglés Programmable Logic Controller) está definido por la International Electromechanical Commission como:

Un sistema eléctrico operado digitalmente, el cual está diseñado para su uso en ambientes industriales, contiene una memoria programable para el almacenamiento de instrucciones (del usuario), para implantar funciones específicas de lógica, secuenciación, temporización, conteo y aritméticas, para llevar a cabo el control de diversos tipos de máquinas y procesos a través de entradas (analógicas y/o digitales), produciendo las correspondientes salidas (analógicas y/o digitales). El control lógico programable y sus periféricos asociados han sido diseñados para su fácil integración y uso en sistemas de control en el ámbito industrial, para llevar a cabo las funciones para que fue programado.

## Breve sinopsis del PLC

Las especificaciones para diseñar el primer controlador lógico fueron establecidas en 1968 por la división Hydromatic de la General Motors Corporation. El primer punto a considerar era eliminar el alto costo asociado con los sistemas poco flexibles basados en relevadores. Las especificaciones requerían de un sistema de estado sólido con la flexibilidad de los equipos de cómputo, capaz de trabajar en ambientes industriales, que fuera fácilmente programado y mantenido por ingenieros y técnicos de la planta.

Los primeros controladores programables fueron sólo sistemas basados en relevadores con capacidad de control de dos posiciones (encendido y apagado) solamente. De 1970 a 1974, las primeras innovaciones en la tecnología de fabricación de los microprocesadores contribuyeron grandemente a incrementar la flexibilidad y capacidad para la toma de decisiones de los controladores programables, así como el contador con una interfase con el operador más amigable, funciones aritméticas de mayor sofisticación, manipulación de datos y comunicación con computadoras. El programador apoyado en equipos de cómputo permitió al usuario realizar programas de control utilizando símbolos de relevadores más conocidos por el personal de la planta, más que cualquier otro lenguaje de programación.

Los avances obtenidos entre 1975 y 1979, en hardware y software agregan características, tales como gran capacidad de memoria, entradas y salidas remotas, control analógico y de posición, comunicación con el operador y enriquecimiento del software. Estos avances hicieron a los controladores programables apropiados para un rango amplio de aplicaciones y contribuyeron grandemente a la reducción de alambreado y costo de instalación. Los sistemas de expansión de memoria permitieron almacenar grandes cantidades de datos y programas más extensos.

A principios de la década de los años 80, se tuvieron grandes avances en los controladores programables en forma paralela a la tecnología de microprocesadores debido a la competitividad de los fabricantes de estos equipos. Se produjo un intento de estandarización de las comunicaciones con el protocolo MAP (Manufacturing Automation Protocol) de General Motors. También fue un tiempo en el que se redujeron las dimensiones del PLC y se pasó a programar con programación simbólica a través de ordenadores personales en vez de los clásico terminales de programación.

El mercado de los nuevos PLC creció de \$ 80 millones de dólares en 1978 a un billón de dólares de forma anual hasta 1990, y aún hoy día continúa creciendo. Inclusive la industria de máquinas y herramientas usó en el pasado computadoras de control numérico (CNC), incursionaron en el uso de PLC. Los PLC fueron también usados ampliamente en la construcción y desarrollo de sistemas de energía y seguridad. Otros usos no tradicionales del PLC fueron el hogar y los equipos médicos, los que aumentaron en uso en la década de los 90.

### Arquitectura típica de un PLC

Un PLC se compone de las siguientes partes funcionales:

- Módulos de entrada.
- Módulos de salida.
- Unidad Central de Proceso (CPU)
- Memoria de Programa.
- Bus de Datos.
- Fuentes de Alimentación.

Estas unidades son fundamentales en el PLC. Cada una de ellas cumple con una función específica.

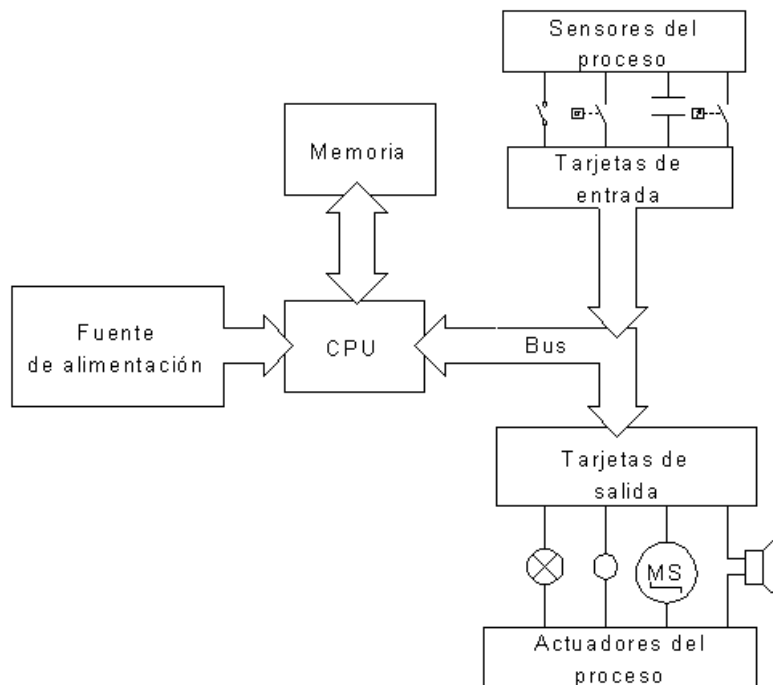


Figura 1-1 Partes fundamentales de un PLC.

### Módulos de entrada y salida.

Los módulos de entrada y salida constituyen la interfase entre los emisores de señal y los actuadores de la máquina o proceso que se va a controlar.

### **Módulos de entrada.**

Reciben las señales del proceso y las adaptan a los niveles de señal internos del equipo. Estas señales ya acondicionadas viajan a través de un bus hasta el CPU. Este ejecuta el programa control y basándose en él se producen señales de comando o de respuesta.

### **Módulos de salida.**

Toman las señales de respuesta elaboradas por el CPU y realizan la conversión en sentido contrario. Es decir, a partir del nivel de señal interno, los módulos producen el nivel adecuado para alimentar relevadores, electro válvulas, lámparas y en general elementos finales de control.

## **Unidad central de proceso**

La unidad central de proceso, el cerebro del PLC. Se encarga de ejecutar el programa existente en la memoria.

Dependiendo de lo que el programa contenga, el CPU consulta si las entradas al aparato tienen o no tensión aplicada. Basándose en esto se elabora un resultado y le ordena a los módulos de salida la conexión o desconexión de determinadas bornes. Con ello, los elementos de control, como actuadores y lámparas, también son conectados o desconectados.

### **Memoria de programa.**

El programa de control se almacena en la memoria del programa. Con la ayuda de un programador se escriben una a una de las instrucciones del programa de control.

Cuando se desean hacer cambios en la lógica o secuencia de control, basta modificar el programa contenido en la memoria.

### **Bus de datos**

El bus de datos es la vía por la que se intercambian todas las señales entre el CPU y las tarjetas de entrada y salida.

### **La fuente de alimentación**

Genera, a partir de la tensión de suministro, la tensión de operación necesaria para que todos los elementos electrónicos del PLC funcionen, tales como el CPU, módulos de entrada y salida, etc.

El voltaje de operación normalmente es mucho menor que el voltaje de suministro, del orden de 5 a 24 V. Sin embargo los circuitos de los dispositivos de control requieren de un voltaje de trabajo superior (24V, 115V, 220V, etc.). Este voltaje es suministrado por fuentes de alimentación externas, no por la fuente de alimentación del PLC.



## Ventajas de los PLC

### Independencia del cableado con respecto a la estrategia de control

Debido a que las interfases de entrada y de salida están aisladas galvánicamente unas con respecto a otras y de la CPU. Es posible definir la relación de estas por medio de un programa, lo cual facilita la operación, modificación y expansión del sistema automatizado. Con esto es posible variar procesos mediante un PLC.

### Capacidad analógica y digital

La mayoría de los PLC están orientados hacia el control de elementos de dos posiciones, y de hecho en muchas aplicaciones se utilizan para reemplazar sistemas de control basados en relevadores.

La posibilidad de manipular variables analógicas, ya sea corriente o voltaje, amplía el campo de aplicación de estos sistemas, debido a la existencia de una gran variedad de equipo de medición con salidas de este tipo.

### Durabilidad

El reemplazo de estructuras y dispositivos de hardware por elementos de software permite una vida útil ilimitada y libre de mantenimiento de temporizadores, contactos auxiliares, y contadores.

### Flexibilidad

Los sistemas son de tipo modular por lo que cuando es necesario incrementar el número de entradas / salidas, basta con agregar nuevas tarjetas. Debe tomarse en cuenta que existe un número máximo de entradas y salidas para cada modelo, y una vez alcanzado ese límite empieza a destacar la importancia de la comunicación con otras unidades, ya sean de la misma marca o de otras. En este aspecto se debe considerar utilización de expansiones esclavas o inteligentes.

### Ejecución Cíclica

Todos los PLC realizan la ejecución del programa de usuario en forma cíclica, llamado ciclo de barrido (SCAN), con la finalidad de tener un mejor control del microprocesador. Esto se tiene como consecuencia velocidades de respuesta limitadas así como la incapacidad de detectar en forma consistente señales cuya duración sea mayor que la del ciclo de barrido.

Aunque por supuesto los fabricantes de PLCs han resuelto este inconveniente con la fabricación de módulos especiales de respuesta rápida y de alarma.

### Economía

De espacio: En el área ocupada por la CPU pueden existir tantos contactos auxiliares como líneas de programa en la memoria (depende de la cantidad de memoria disponible), así como Temporizadores, Contadores, etc.

De energía: Los contactos auxiliares ubicados en la memoria del PLC no tienen bobinas. Su consumo de energía no depende de que estén actuados o no.

De tiempo: Los indicadores y el uso de aparatos de programación y/o monitoreo, tiene como resultado menores tiempos de instalación, de arranque, de paro, de detección de fallas, de mantenimiento, etc.

En costos: El mantenimiento de equipo de estado sólido es menor, lo cual reduce el costo de operación; el cableado también es reducido, pues sólo se unen las tarjetas de entradas / salidas con los dispositivos de campo.

## **Operaciones aritméticas**

La capacidad de cálculo de los PLC actuales va más allá de un simple conjunto de instrucciones orientadas a operaciones booleanas, con habilidades que van desde una aritmética simple de números de coma fija, hasta modelos matemáticos de punto flotante.

## **Comunicaciones**

Existen equipos que posibilitan la supervisión y control por parte de operadores, utilizando para esto teclados y monitores.

Muchos modelos de controles programables pueden ser conectados a una red de comunicación, lo cual facilita la integración de sistemas de control de grandes magnitudes, compuestos generalmente por PLC de distintas marcas, tamaños y capacidades debido a la existencia de estándares de comunicación.

## **Corrección de errores y facilidad de cambios**

Anteriormente cuando se empleaban circuitos con relevadores, cualquier cambio de control ó corrección del programa, consumía gran cantidad de tiempo para el recableado de panel ó los dispositivos que intervenían dentro del proceso. Con el PLC un cambio de programa se realiza mediante un teclado, en tan solo algunos minutos, todo esto se lleva a cabo sin la necesidad desconectar máquina ó dispositivo alguno.

## **Velocidad de respuesta**

Los relevadores pueden tener un tiempo de respuesta inadecuado para algunas situaciones de control, en este aspecto el PLC tiene una velocidad de respuesta muy rápida, la velocidad de las operaciones lógicas de un PLC está determinadas por el tiempo del ciclo de barrido (SCAN), el cual se lleva en unas cuantas milésimas de segundo.

## **Desventajas de los PLC**

Desde cierto punto de vista lo siguiente no son desventajas, sino precauciones que se deben tomar en cuenta cuando se emplean los PLC.

## **Nueva tecnología**

Muchas ocasiones se tiene que capacitar a personas que ya han trabajado previamente con sistemas con relevadores, aunque hoy en día ya es común observar plantas ó procesos controlados por PLC.

## **Aplicaciones de programa fijo**

Algunas aplicaciones son a base de funciones sencillas. En estos casos no es conveniente el uso de un PLC, cuya capacidad no es del todo aplicable. Algunos fabricantes de equipo aún incluyen dispositivos mecánicos ó de otro tipo para controlar algún proceso sencillo, lo cual se ve reflejado en un menor precio. En este tipo de máquinas se tiene una misma secuencia de operación, por lo cual no es recomendable el uso de un PLC para su control.

## **El entorno**

Algunos entornos relacionados con procesos pueden causar daños a los componentes electrónicos de los PLC. Tal es el caso de procesos que estén relacionados con altas temperaturas ó vibraciones, que puedan limitar su vida útil.

## **Operación a prueba de fallas**

En sistemas de relevadores, el botón de paro desconecta eléctricamente al circuito; si se presenta una falla en el suministro de energía, el sistema se detendrá, además cuando se restablece el suministro, el sistema no reiniciará automáticamente. Aunque esto puede ser programado en un PLC, existirán algunos programas en los cuales, para detener al sistema ó una máquina se tiene que aplicar un voltaje, esto en sí no es un sistema a prueba de fallas. La manera

más sencilla de solucionar esta desventaja es simplemente agregarle relevadores de seguridad al PLC.

## Operación del PLC

La operación de la mayoría de los controladores programables consiste en un ciclo repetitivo de cuatro pasos principales:

1. Todas las entradas provenientes de las interfases, controladores en lazo cerrado o de algún otro dispositivo de control en el bus de entrada / salida son leídas a fin de producir una “imagen” consistente de éstas, denominadas “imágenes de entrada”.
2. Las imágenes de entrada son leídas por el controlador y el programa del usuario, lo que ha sido denominado “ciclo de barrido o ciclo de scan” para generar las nuevas imágenes, que son las de las salidas deseadas, así como las variables internas del programa. A partir de las imágenes de entrada, las variables internas y las de imágenes de salida, el programa en este ciclo de barrido genera las variables de salida. Este proceso consiste de varios pasos que enseguida se detallan:
  - a. Primero se determinan los pasos activos (en este nivel del programa) de la carta secuencial de funciones (SFC, del inglés Sequential Function Chart) si es que existen. Esta información está contenida en el programa.
  - b. Cálculo de las salidas indicadas en las acciones activas del la SFC, si es que éstas existen (sí el programa del usuario no contiene SFC, entonces todas las instrucciones del programa se consideran como acciones activas). La ejecución de programas en diagramas de escalera o bloques de funciones se lleva a cabo típicamente de izquierda a derecha y de arriba hacia abajo. Algunas instrucciones están situadas de manera tal que ciertas secciones sean saltadas o no ejecutadas o también para forzar las salidas a un estado determinado.
  - c. Evaluación de las condiciones de transición de la SFC (sí existen) al final del ciclo de barrido del programa, en preparación del paso 2(a) para el siguiente ciclo de barrido.
3. Los datos actualizados de las imágenes de salida se transfieren a las interfases, controladores de lazo cerrado ó algún otro dispositivo de control.
4. Por último se ejecutan las tareas de manipulación final, entre las que se pueden mencionar principalmente las de comunicación con el operador o con un controlador de supervisión o con algún otro dispositivo de control.

Después de llevar a cabo estas tareas de manipulación final de operación cíclica del PLC inicia de nuevo. Esto puede ser determinante después de la ejecución de las tareas o funciones de manipulación final o también puede ser previamente programado.

Algunos controladores programables con secciones de entrada / salida por separado ó con procesadores de comunicaciones tienen la capacidad de traslapar las funciones de ejecución del programa y lectura de entradas (paso 1) y la generación de las salidas (paso 3) y las funciones o tareas de comunicación. En estos casos, se requerirán de mecanismos de programación especiales para alcanzar la concurrencia y sincronización entre la ejecución del programa, la lectura de entradas y la generación de salidas, y entre la ejecución del programa y las comunicaciones.

El ciclo de operación básico de un PLC se muestra en la siguiente figura:

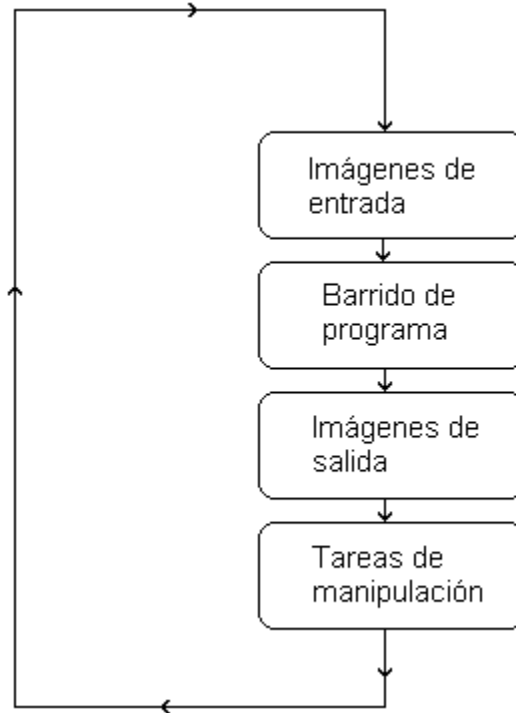


Figura 1-2 Ciclo de operación de un PLC

### Elementos de hardware del Simatic S5 100-U

El Simatic 100-U es un controlador programable (PLC) parte de una familia de controladores programables, siendo el equipo diseñado para las tareas de automatización de la gama media de aplicación como son aquellos procesos en los cuales el conteo de las señales de entrada / salida digitales no sobrepasan un máximo de 256.

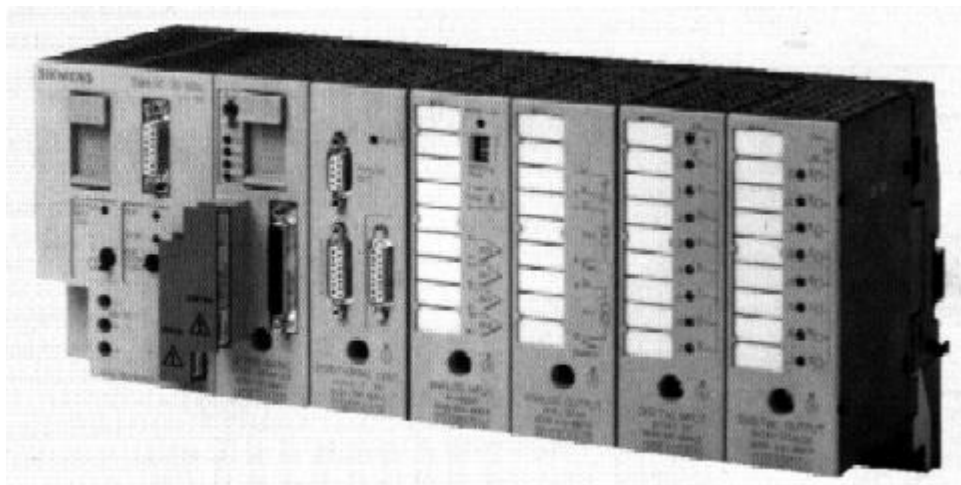


Figura 1-3 PLC Simatic S5 100-U

La familia Simatic S5-100U provee de uno de los autómatas PLC más flexibles debido a su configuración modular, y a la gran variedad de tarjetas de entradas / salidas y a la existencia de módulos autónomos configurables. Este sistema es pues idóneo para la automatización de máquinas y pequeños procesos, permitiendo el control de periferia digital, analógica, comunicaciones seriales, entradas de alta velocidad, y sus aplicaciones típicas son secuenciadores, lógica combinacional, control de lazo cerrado, y sistemas de identificación automática, pesaje, y muchos más.

### Características generales

Como hemos dicho, este PLC modular. Para este equipo, la mínima configuración de elementos de Hardware requeridos para completar un sistema consiste en lo siguiente:

- 1 CPU
- 1 fuente de alimentación
- 1 elemento de Bus
- 1 módulo de entrada
- 1 módulo de salida

La configuración máxima consiste en utilizar hasta 32 módulos de entrada / salida, que pueden estar en una o varias filas, de modo que esto podría enlistarse así:

- 1 CPU
- 1 Fuente de alimentación
- 1 Módulo de interfase para expansión en varias filas
- Hasta 16 elementos de bus
- Hasta 32 módulos de entrada / salida

Existen para el modelo Simatic 100 U tres distintas CPU, que se refieren en este texto y son CPU 100, CPU 102 y CPU 103. La siguiente tabla muestra algunas de sus características más relevantes.

CPU	100	102	103
Bloques de organización	4	4	8
Bloques de programa	64	64	256
Bloques de funciones	64	64	256
Bloques de secuencia			256
Bloques de datos	62	62	254
Bloques integrados	no	Sí	Sí
Coprocesadores	no	no	Sí
Memoria de usuario	2Kb	2Kb	20Kb
Banderas	1024	1024	2048
Temporizadores	16	32	128
Contadores	16	32	128
Entradas digitales	256	448	448
Entradas analógicas	8	16	32
Sinec L1	no	Sí	Sí
Reloj de hardware	no	no	Sí
Control PID	no	no	Sí
Parametrización de FB	no	no	Sí

Tabla 1-1 Elementos que integran las CPU

Pues obsérvese que las CPU más avanzadas poseen una mayor cantidad de elementos direccionables (temporizadores, contadores, banderas, bloques de programas) así como también funciones especiales como reloj de tiempo real, módulos de funciones parametrizables estándar y de usuario, algoritmo de control PID.

## Bloques Funcionales

Analizaremos primero la constitución interna de la CPU. Para todas las marcas de PLC y dentro de cada familia, se dispone de varios modelos para la unidad central de procesamiento. En el caso concreto del Simatic 100U, se tienen diferentes CPU que son: 100U, 102U y 103U.

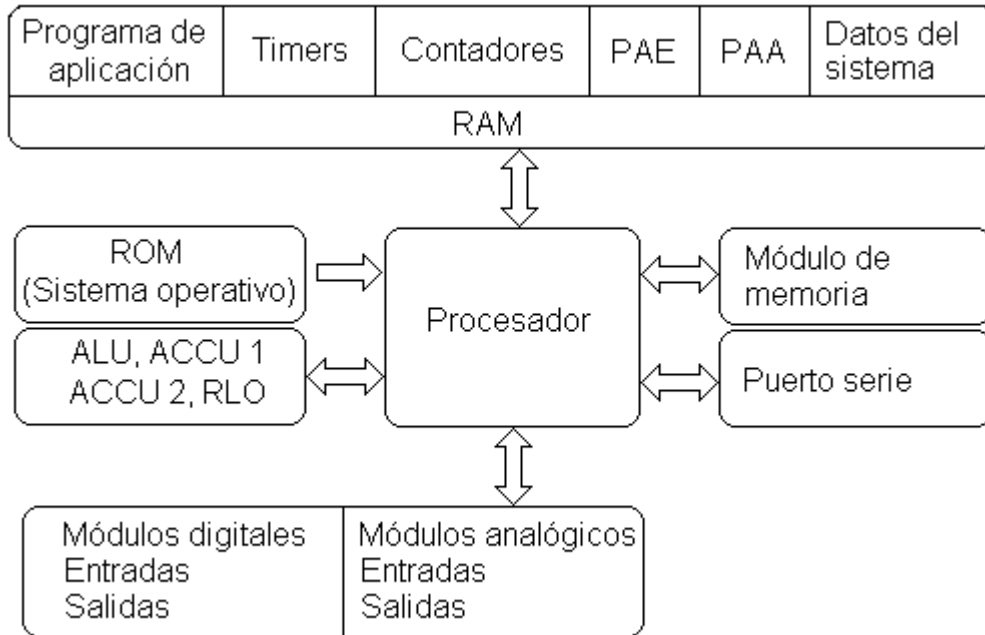


Figura 1-4 Intercambio de datos entre la CPU y su periferia

La CPU se compone básicamente de un procesador, una unidad lógica aritmética, y memoria. Adicionalmente se dispone de un puerto para un cartucho o módulo de memoria y un puerto serial.

La función principal del procesador es comandar las actividades de todo el sistema. Esto se logra a través de la interpretación y ejecución de un grupo de programas contenidos en el sistema operativo, el cual supervisa y controla los dispositivos internos y externos. De esta forma la unidad de procesamiento gobierna y coordina todo el autómata. Siguiendo el módulo de organización cíclica, llama sucesivamente las instrucciones contenidas en otros módulos de programa, y las ejecuta; para ello se procesan las informaciones contenidas en la imagen de proceso de entrada (PAE) y se consideran los valores de los temporizadores y contadores internos así como los estados de señal de las banderas internas. Los resultados se registran en la imagen de proceso de salida (PAA).

La unidad lógica y aritmética se compone de dos acumuladores de 16 bits, llamados ACCU1 y ACCU2. A través de estos se procesan las operaciones por byte y por palabra. También se dispone de un registro tipo bit en el cual se guardan los resultados de todas las operaciones binarias, el cual se conoce como RLO ( de sus siglas en ingles Result of Logic Operation).

Los sistemas programables necesitan de un espacio en el cual almacenar toda la información con la cual operan, es decir datos de entrada, datos intermedios, el programa, el sistema operativo, y datos de salida. Este espacio de almacenamiento constituye la memoria del sistema.

El mapa de memoria contiene una parte de ROM, en la que se ubica el sistema operativo y el compilador, una sección de EEPROM para el programa de aplicación, y una parte de RAM, en la cual se guardará el programa de aplicación, los datos del sistema, las imágenes del proceso, y los

valores de los contadores y temporizadores. A continuación veremos el mapa de memoria para dos diferentes CPU, de la serie Simatic.

CPU	102	103
Memoria de programa	D000... DFFF	8000...CFFF
Módulo memoria	4000... 5FFF	0000...7FFF
PAE digital	EF00... EF1F	EF00... EF1F
PAE analógica	EF40... EF7F	EF40... EF7F
PAA digital	EF80... EF9F	EF80... EF9F
PAA analógica	EFC0... EFFF	EFC0... EFFF
Temporizadores	EC00... EC39	EC00... ECFF
Contadores remanentes	ED00... ED0F	ED00... ED0F
Contadores no remanentes	ED10... ED3F	ED10... ED3F
Banderas remanentes	EE00... EE3F	EE00... EE3F
Banderas no remanentes	EE40... EE7F	EE40... EE7F

Tabla 1-2 Memoria del PLC (fragmento)

El mapa de memoria en realidad es una parte transparente para el usuario común. Nosotros podemos notar que es evidente la existencia de dos diferentes características en el mapa ilustrado arriba, y que se define como remanencia. Esto es, la característica de que una variable conserve su valor después de haber apagado el PLC. En ocasiones puede ser ventajoso, en otras no. Para el usuario de Simatic basta con elegir el temporizador, contador o bandera específicos dentro de cada CPU para hacer uso del mejor. En cualquier caso, la remanencia está condicionada al respaldo por batería.

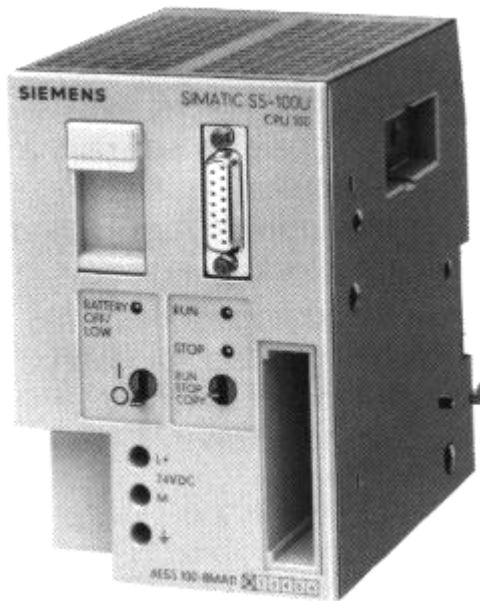


Figura 1-5 Simatic S5 100-U / CPU 100

Receptáculo para módulo de memoria. Este puerto se utiliza para conectar un cartucho de memoria EPROM / EEPROM como unidad de respaldo del programa de usuario, por ser una unidad de solo lectura, desde el punto de vista del PLC.

Puerto de programación. El puerto serie trabaja bajo Protocolo TTY, el cual es un lazo de corriente que permite la comunicación hasta distancias de 1000 metros. A través de este puerto se

posibilita la conexión de programadores Simatic, computadoras y dispositivos de visualización y control tales como los OP 393/5/15/25/35 del sistema COROS, y la construcción de redes SINEC.

### Sistemas de Entradas y Salidas

El sistema de entradas / salidas forma la interfase por la cual los dispositivos de campo son conectados al controlador. La función de este sistema es acondicionar las señales, tanto las que se reciben como las que se envían, a niveles de voltaje y corriente apropiados para cada componente del sistema.

Se les llama señales de entrada a la proveniente de todos aquellos dispositivos emisores de señal como son botones pulsadores, interruptores de fin de carrera, sensores analógicos, selectores y botones; pudiendo recibir valores de proximidad, posición, movimiento, niveles, temperatura, presiones, etc.

Cada módulo de entrada / salida es alimentado por una fuente suplementaria que no necesariamente es del mismo tipo o magnitud que la salida correspondiente, esto permite tener dos voltajes en un mismo sistema, por ejemplo toda la parte operativa puede estar a 24 VCD, mientras que la parte de mando trabaja a 115 VCA.

### Módulos de entrada y salida digital

Las interfaces de entrada / salida más comunes son aquellas que manejan señales digitales, es decir, sensan señales de dos estados, ON/OFF, OPEN/CLOSE o sus equivalentes. A continuación se listan algunas de los elementos de entrada / salida más frecuentes utilizados.

Dispositivos de entrada digital	Dispositivos de salida digital
Selectores	Alarmas sonoras
Botones	Control de relevadores
Sensores fotoeléctricos	Ventiladores
Retro alimentación de contactos	Lámparas
Contactos	Contactos
Interruptores de proximidad	Válvulas
Indicadores de nivel	Motores
Arranque / paro de motor	Válvulas Solenoides

Tabla 1-3 Algunos elementos de entrada / salida

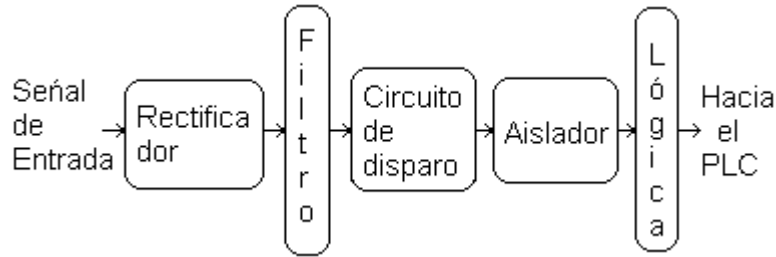
### Elementos de entrada y salida

Los estados típicos de entradas y salidas puede definirse con palabras, tales como:

- Manual / Fuera / Automático*
- Encendido / Apagado*
- Adelante / Fuera / Reversa*
- Falla / Normal*
- Alto / Normal / Bajo*
- Apagado / Abajo / Medio / Alto*



Un diagrama de bloques típico para la interfase de entrada / salida digital de AC /DC es como se muestra:

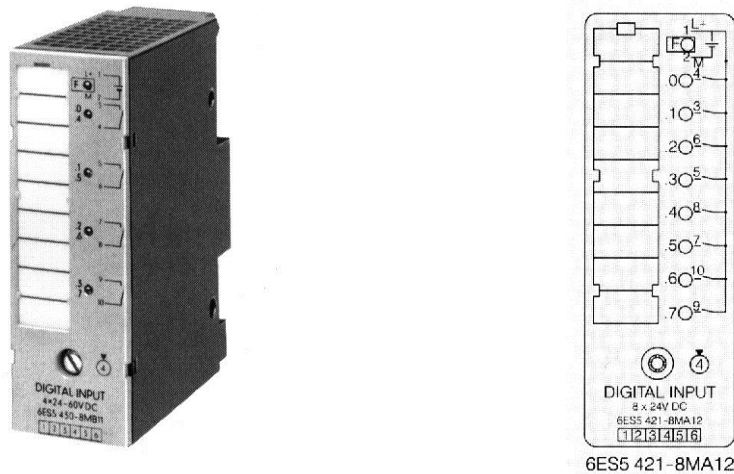


**Figura 1-6 Diagrama de la interfase de entrada digital**

Se puede observar en el dibujo que se dispone de un filtro. Este es importante para limitar los rebotes de la señal. También, una de las partes más importantes, es el aislador, que por lo general se realiza con un opto acoplador. Este nos provee de separación galvánica, que limita la propagación de una sobre tensión, polaridad inversa, o corto circuito, protegiendo de esta forma todo sistema.

Similarmente, para una salida digital, se suele disponer de una red de protección, a través de un opto acoplador de tiristor del tipo MOC3011.

### Módulo de entrada digital 8 x 24VDC 6ES5 421-8MA12



**Figura 1-7 Módulo de entrada digital 6ES5-421-8MA12**

Este módulo tiene la función de acoplar señales provenientes de contactos secos (por ejemplo interruptores, pulsadores). A través de este módulo se pueden conectar ocho entradas binarias de 24 VDC. La impedancia de entrada es tal que la corriente demandada es típicamente de 7 mA.

Los puntos de entrada están aislados en grupos de 8, por lo que es necesaria una fuente de alimentación externa, la que se debe conectar en los bornes 1 y 2. Las entradas (o señales provenientes de los emisores) se conectan en los bornes 4,3,6,5,8,7,10 y 9, y corresponden a los canales 0, 1,... 6, y 7 respectivamente.

El estado de cada una de las entradas se señaliza por medio de un diodo emisor de luz, en la parte frontal de la tarjeta.

Direccionamiento: Este módulo puede ser direccionado como entrada únicamente, en las tres modalidades de Word, Byte y también por puntos individualmente como Bit, que es lo más usual.

### Módulo de salida digital 4 x 24VDC /2A 6ES5 440-8MA21

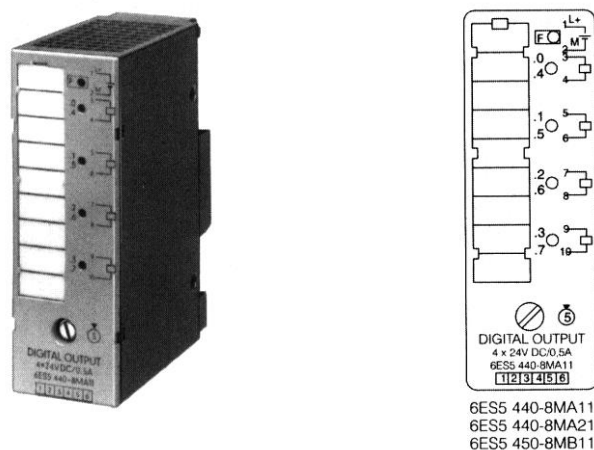


Figura 1-8 Módulo de salida digital 6ES5 440-8MA21

Esta interfase de salida permite energizar cuatro puntos con un voltaje de 24 VDC a una intensidad máxima de 2A. El voltaje de salida se provee a través de transistores. Se requiere de una fuente de alimentación de 24 Volts, la cual se conecta en los bornes 1 y 2.

Los bornes de salida 3, 5, 7 y 9 están conectados a los colectores de los transistores, y los bornes 2, 4, 6, 8 y 10 están unidos y conectados al potencial 0 de la fuente externa. De esta forma una salida (por ejemplo un contacto) se conecta entre un borne de salida y el potencial 0.

#### Notas:

El cableado de las señales de módulos digitales pueden tenderse en un conducto común, sin blindaje.

Se tiene protección contra corto circuito, con reconexión automática.

El voltaje de salida bajo máximo es de 3V para un 0 lógico.

Direccionamiento: Este módulo puede ser direccionado como Salida únicamente, tanto como Word, Byte y Bit.

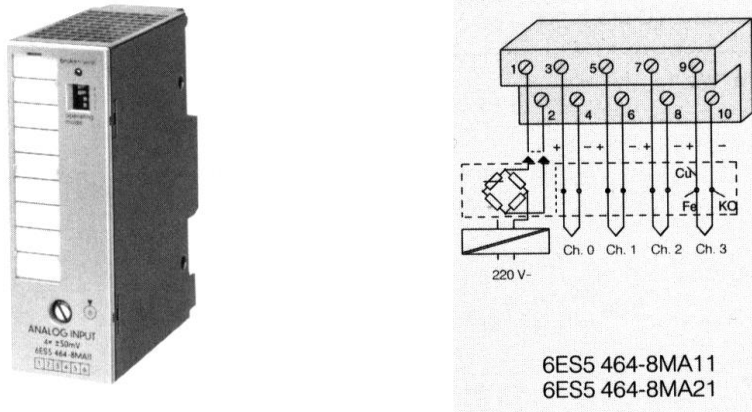
Hay que destacar un punto referente a las tablas de especificaciones. Si bien es válido hacer circular la corriente nominal en cada una de los puntos, no se vale energizar todas las salidas al mismo tiempo. Es decir, la carga total admisible para este módulo no es la suma de la corriente admitida por cada punto. De esto se deriva la práctica frecuente de conectar en un módulo señales que no pueden coexistir simultáneamente.

### Módulos de entrada y salida analógicas

Las interfases de entrada / salida analógicas permiten el monitoreo y control de intervalos de voltaje y corriente provenientes de señales continuas como sensores de temperatura, presión, celdas de carga, humedad, flujo y potenciómetros; y envían señales a válvulas y actuadores analógicos, controladores de motores (drivers) e instrumentos de proceso.

Las entradas / salidas analógicas están generalmente disponibles en intervalos normales (4-20mA, 0-20ma, 1-5V, 0-10V, RTD 50mV, etc.). En algunos casos un módulo de entrada / salida puede alojar varios tipos de intervalos al mismo tiempo, lo cual le da una mayor flexibilidad al PLC. Los intervalos que se manejan pueden ser unipolares (solo valores positivos) o bipolares.

**Módulo de entrada analógica 4 x + 50mV**  
**6ES5 464-8MA11**



**Figura 1-9 Módulo de entrada analógica 6ES5 464-8MA11**

Esta interfase permite la adquisición de datos analógicos, dentro del intervalo de más menos 50 mV y está diseñada específicamente para la medición de temperaturas por medio de termopares.

El módulo contiene un convertidor de 12 bits por integración doble rampa, que realiza con un tiempo de conversión máximo de 80 ms por entrada. La impedancia de cada entrada es de 10 MΩ, y se tiene separación galvánica de las entradas con respecto a tierra, pero no entre entradas. La máxima diferencia de potencial entre estas no debe exceder 1 V, es decir, se tiene un margen de sobrecarga de 10 entradas es a dos hilos, y se puede seleccionar el número de entradas utilizadas por medio de un conjunto de interruptores (selectores) colocado en la parte frontal.

50	60	Frecuencia
		Interruptor 1
		Interruptor 2
No	Si	Señalización

**Tabla 1-4 Interruptores de selección del módulo 6ES5 464-8MA11**

Los interruptores 2 y 3 se utilizan para definir el número de canales activos, de acuerdo con lo mostrado en la tabla 1-4.

Cuando se utiliza solamente un canal, y quedan tres canales sin utilizar, se deberán cortocircuitar todos los canales no utilizados con el propósito de reducir el tiempo de conversión. El canal único deberá ser el canal 0.

En caso de utilizar dos canales, estos deberán ser el 0 y el 1, es decir, se eligen en orden ascendente, y se cortocircuitan los bornes no utilizados. En caso de no hacer esto, se tiene una operación errónea y no se logra hacer ninguna medición. Esto se debe a que se tiene un solo

convertidor A/D, y las entradas se multiplexan. Este módulo de entrada no requiere de una fuente de alimentación externa, puesto que se toma la energía del bus.

**Nota:**

No mover los selectores mientras el autómata esté encendido, puesto que solamente se leen cuando el PLC pasa de Stop a Run.

Número de canales	Interruptor 1	Interruptor 2
1	Libre	Activado
2	Activado	Libre
4	Activado	Activado

Tabla 1-5 Selección del número de canales del módulo 6ES5 464-8MA11

El direccionamiento de un canal de este módulo se hace tomando en cuenta la posición que ocupa dentro de las primeras 8 ranuras para conexión, siendo el primer canal analógico direccionable el PW128.

Módulo de salida analógica 2 x 1 5V

**6ES5 470-8MD12**

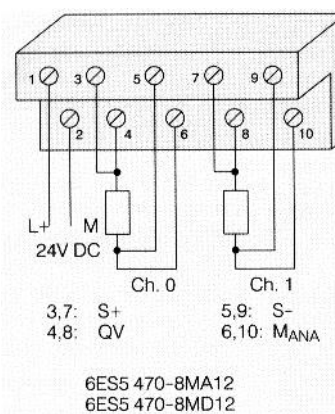
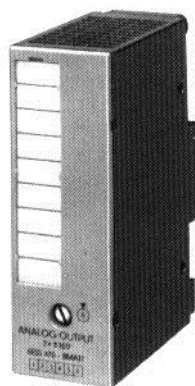


Figura 1-10 Módulo de salida analógica 6ES5 470-8MD12

Este módulo contiene un convertidor D/A de 11 bits + signo con lo cual es posible la salida de 2048 distintos valores de voltaje dentro del margen 1 a 15 Volts. La entrada al convertidor es un número binario representado en complemento a dos, y este realiza la conversión en un tiempo máximo de 0.1 mS. Se tiene separación galvánica entre las salidas y también con respecto a tierra. La salida tiene protección contra cortocircuito, con una intensidad máxima de 30 mA.

Este módulo se utiliza para enviar señales de control a dispositivos que posean entrada analógica, ó para monitoreo de variables en medidores convencionales de montaje en tablero.

Precaución: Solo es admisible una resistencia de carga mayor de 3.3kΩ.

La carga se conecta entre los bornes 4(+) y 5 para el canal 0 y los bornes 8(+) y 9 para el canal 1. Los bornes 3, 6, 7 y 10 se conectan a la carga y sirven para compensar las caídas de voltaje en los cables que alimentan la carga. Logran esto a través de una medición del voltaje en la carga, la cual se realiza a través de cables de alta impedancia.

**Notas:**

Las señales analógicas deben utilizar cable blindado, con la malla conectada a tierra en uno solo de los extremos.

Los cables que conducen señales pueden tenderse con cables de energía de hasta 380 V sin necesidad de observar una separación mínima.

El módulo se alimenta con 24VCD, soporta un rizo máximo de 3.6 Vpp.

## Módulos Especiales

Los módulos que se describen a continuación no son de uso convencional. Algunos de ellos se utilizan como interfaz con el operador para capturar señales de alta velocidad, o para simulación en la etapa de pruebas. Veamos a continuación los siguientes módulos:

### Módulo de temporizadores

#### 6ES5 380-8MA11

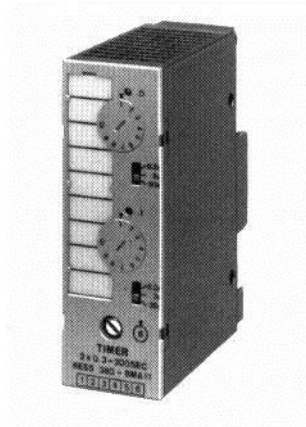


Figura 1-11 Módulo temporizador 6ES5 380-8MA11

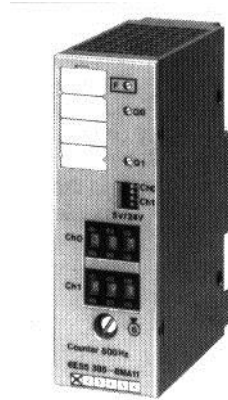
Este módulo se utiliza como una interfase hombre máquina, para el ajuste de tiempos cuando no se dispone de una terminal de programación ó un panel de operador. Contiene dos temporizadores SP (tipo pulso). No se requiere de ninguna conexión en los bornes, por lo cual se dejan abiertos. Se direcciona como un módulo de salida digital de 2 canales, para las operaciones de arranque, borrado, e inicialización. Para la consulta del estado de la señal se direcciona como un módulo de entrada digital.

Mientras corre la temporización (es decir, mientras el valor del contador está disminuyendo, pero aun no llega a cero) se enciende el diodo emisor de luz del canal respectivo, y la consulta resulta en estado "1". El valor de temporizado se ajusta con un potenciómetro y un selector de base de tiempo.

**Notas:**

Este módulo toma su alimentación del bus. No posee conexiones externas.

## Módulo de contadores **6ES5 385-8MA11**



**Figura 1-12** Módulo de contadores 6ES5 385-8MA11

Para un cambio de estado en una entrada digital sea detectado, ésta debe mantenerse por lo menos durante un ciclo de barrido completo. Con los módulos de entrada de contador se posibilita el conteo de pulsos eléctricos a frecuencias mayores que las admitidas por una entrada digital normal. Este tipo de señal proviene generalmente de codificadores de posición (Enconders). El modelo 385-8MB acepta frecuencias de entrada de hasta 500 kHz. La tarjeta de módulo de contadores que se ilustra aquí consiste en dos contadores decrementables que aceptan una frecuencia de entrada máxima de 500 Hz. Se debe conectar la entrada en los bornes 3(+) y 4 para el canal 0, y 7(+) y 8 para el canal 1. El voltaje de la señal de entrada puede seleccionarse entre 5 y 24 VCD. Se tiene separación galvánica entre las entradas y la CPU.

Las salidas se conectan entre los bornes 5 y 6 para el canal 0, y 9 y 10 para el canal 1. Note que la fuente de alimentación está en serie con la carga.

Se tiene protección electrónica contra cortocircuito, dicho evento se señaliza con un diodo emisor de luz.

Se envía una señal a la CPU cuando el contador llega a cero; para esto se examina como una entrada digital.

El borrado del contador se hace como una salida digital, y con esta operación carga el número que ha sido ajustado en el panel frontal del módulo.

La intensidad típica de entrada es de 1.58/8.5 mA (TTL/24V), la salida está protegida contra cortocircuito, y soporta una intensidad máxima de 500 mA.

### Módulo de comparadores **6ES5 461-8MA11**

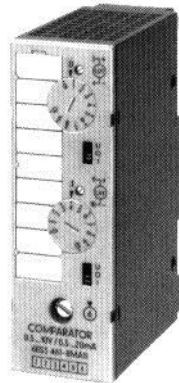


Figura 1-13 Módulo de Comparadores 6ES5 461-8MA11

Este módulo contiene dos comparadores con separación galvánica que pueden medir corrientes o voltajes, y al llegar al límite ajustado, se señaliza a la CPU y al usuario por medio de un diodo emisor de luz.

El direccionamiento se hace como un módulo de entrada digital. El canal 0 se conecta en los bornes 3 y 4, y el canal 1 en los bornes 9 y 10. No se requiere de una fuente de alimentación externa, y las entradas no utilizadas pueden permanecer abiertas. El margen de corriente de entrada es de 0.5 a 20 mA, y el de tensión de 0.5 a 10 V. La impedancia de entrada es de 47 k $\Omega$  para medición de voltaje, y de 500  $\Omega$  para medición de corriente. La histéresis es menor o igual al 10%.

### Módulos simuladores de entradas ó salidas digitales **6ES 788-8MA11**

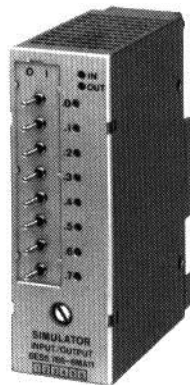


Figura 1-14 Módulos simuladores de entradas ó salidas digitales 6ES 788-8MA11

El simulador 788 es una tarjeta de 8 bits con aplicaciones de simulación y de diagnóstico en la puesta en marcha de un sistema 100 U.

Estos módulos pueden configurarse como entradas o salidas por medio de un selector ubicado en la parte posterior del mismo. No requieren de conexión externa, toman la alimentación del bus, y no poseen llave.

En la configuración de entradas los 8 canales son actuados por los interruptores frontales, y se confirma el estado de cada entrada por medio de un diodo emisor de luz.

En la configuración de salida los diodos emisores de luz indican el estado de cada canal, al estar encendido indica que la salida esta alta o que el bit contiene un "1" lógico. Los interruptores son inoperables cuando el módulo ha sido configurado como salida.

## Módulos de memoria

El módulo de memoria es el cartucho de color naranja que se observa en la parte frontal de la CPU hacia el lado derecho.

Funciona como respaldo del programa de control en prevención a una falla en el suministro de energía. Con estos módulos es posible transferir programas al autómatas, simplemente se inserta el cartucho en la ranura y se opera la perilla "copy". Esto tiene como resultado una carga del programa que está en el cartucho hacia la RAM interna del CPU. Una vez copiado el programa es posible retirar el cartucho, siempre y cuando no este encendido el PLC, para hacer múltiples copias en tantas máquinas como se quiera, o para guardarlo en un lugar seguro. El programa en la RAM se mantendrá por la acción de la batería.

Se tienen dos tipos de cartuchos:

Módulo EPROM: Para el borrado de este tipo de módulo es preciso utilizar un borrador de luz ultravioleta. La grabación se hace con un PG.

Módulo EEPROM: Este dispositivo puede ser borrado y grabado directamente por los aparatos de programación o por el autómatas Simatic S5-100U.

En los aparatos de programación se lleva a cabo mediante el paquete denominado EEPROM contenido en el Software Step 5.

El método que se utiliza únicamente el PLC consiste en una secuencia de manejo de las perillas, y tiene dos restricciones de seguridad: se debe tener insertada la batería y debe haber un programa ejecutable en la RAM.

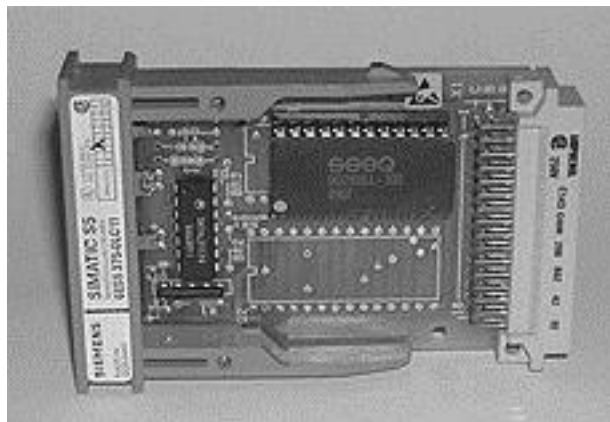


Figura 1-15 Cartucho de memoria.



## El Bus Simatic

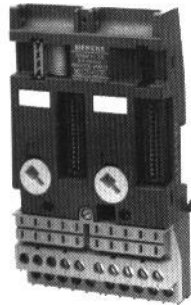


Figura 1-16 Bus Simatic.

El bus sirve para comunicar las interfases de entrada / salida con la CPU. La información contenida en los registros de cada dirección se transfieren a localidades de la RAM durante cierta parte del ciclo scan, por medio del bus. Esto permite una gran reducción del cableado porque todas las tarjetas de entrada / salida envían información a través de los mismos hilos, de acuerdo al orden de llamadas y envíos que ha sido determinado por el fabricante del sistema. En algunos casos se utiliza un par trenzado para un bus serial, mientras que en otros sistemas se utilizan más hilos para conformar un bus con señales en paralelo.

### Construcción del Bus en el S5-100U

Los elementos de bus son parte fundamental para la implementación de un sistema Simatic S5-100U, debido a que sobre ellos se conectan los módulos de entrada y salida. Cuenta con un bloque de conexiones (clemas) aislado galvánicamente, y además con una llave en hardware que permite la entrada de un determinado tipo de módulo únicamente. Esta llave se puede girar para dar acceso a módulos analógicos de entrada y salida, y módulos digitales de entrada / salida, así como también módulos inteligentes. La llave es una pieza de plástico que se encuentra ubicada en el centro interior del elemento de bus.

En cada CPU pueden conectarse hasta 16 elementos de bus, los cuales se conectan por medio de un cable plano, y dos terminales ya integradas. Se pueden colocar dos módulos en cada elemento, con lo cual se tiene capacidad máxima de 32 módulos de entrada / salida en un Simatic S5-100U.

El elemento de Bus consta de las siguientes partes mecánicas:

- Gancho de sujeción al riel DIN
- Cable plano para intercomunicación
- Borne para cable plano
- Dos peines para interconexión de tarjetas entrada / salida
- Ganchos para colocar los módulos de entrada / salida
- Tuercas para asegurar en su lugar los módulos de entrada / salida
- Bornes de conexión para alambrar los elementos de campo

Los bornes de conexiones forman parte del elemento de bus y no de los módulos, por lo cual no es necesario desconectar el cableado para reemplazar una tarjeta de entrada / salida dañada, lo que reduce el tiempo de reemplazo y evita tener errores de cableado.

Para transferir datos entre la CPU y los módulos periféricos el autómatas cuenta con un bus serial con las siguientes características:

- Su estructura modular permite una exacta adaptación a cualquier tarea de mando.
- No es necesario ajustar direcciones en los módulos periféricos.
- no es necesario ningún dispositivo terminal.
- no es posible acceder directamente a la periferia (a un módulo individual).

La transmisión de datos se realiza a través de una cadena de registros de corrimiento. Cada puesto de enchufe de un elemento de bus tiene asignados cuatro bits de datos y un bit de verificación (para vigilar la transmisión por el bus). Todos los módulos que precisen más de 4 bits de datos tienen un registro de corrimiento propio, que sustituye al registro de corrimiento del puesto de enchufe.

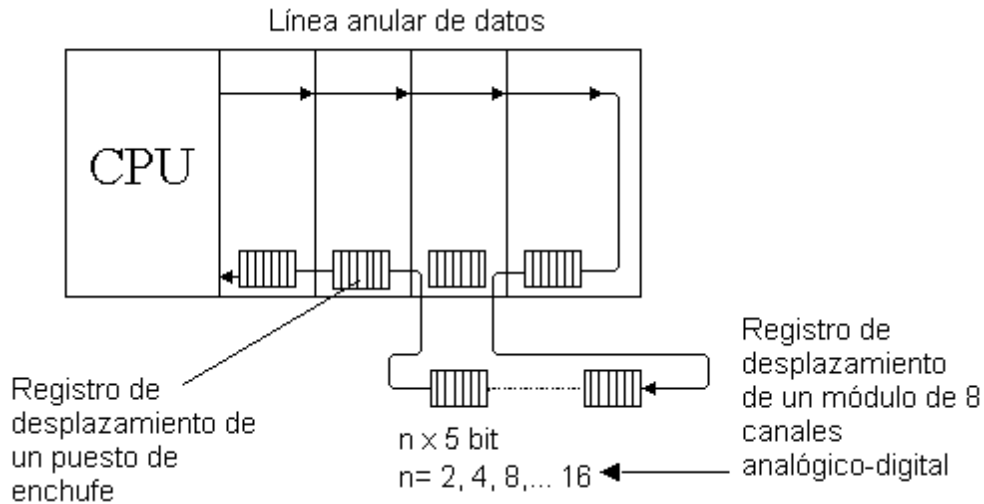


Figura 1-17 Registro de corrimiento.

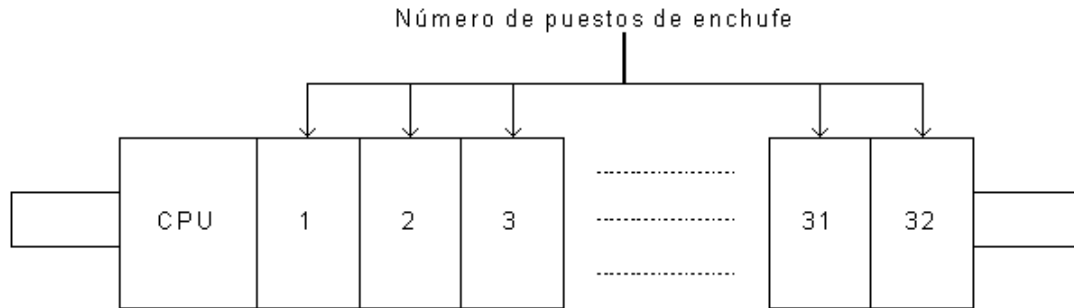
En la figura se puede observar cómo el registro de corrimiento del elemento de bus no es utilizado sino se reemplaza por los registros del módulo conectado, en este caso un módulo analógico.

### Concepto de dirección

La dirección es un concepto de software. En nuestro caso, consiste en una notación utilizada en el software STEP 5 para acceder un punto de entrada y salida. La dirección de una señal estará definida por la posición y tipo de tarjeta en cuestión dentro del bus del PLC Simatic.

En algunos casos estas direcciones de las tarjetas pueden ser fijadas por software, y en otros casos, sólo depende de la ubicación física de los módulos. En el primer caso, se dice que el direccionamiento es variable y es necesario que el programador establezca lo que denomina mapa de direcciones. Las direcciones contemplan siempre información referente a un canal de alguna tarjeta específica. La configuración de las entradas y las salidas es un dato importante que se debe tener en mente al realizar un programa específico.

Para la serie Simantec S5-100U, el direccionamiento es fijo, es decir, dependerá exclusivamente de la posición física del módulo. Para intercambiar información con un módulo, es preciso conocer su posición con respecto al bus, esto determinará su dirección, lo cual se verá reflejado en el operando. La organización de estos direccionamientos comienza en la posición cero "0", que es el módulo contiguo al CPU, esta posición se incrementará a la derecha. Estos números se conocen como puestos de enchufe. La siguiente figura ilustra como se ordenan los puestos de enchufe en el PLC Simantec S5-100U.



**Figura 1-18 Posición de los enchufes en el PLC**

Se puede observar que cada puesto de enchufe corresponde a un byte, y cada elemento de bus a una palabra. (El primer elemento del bus siempre corresponderá a la palabra cero "0", cuya parte alta pertenecerá a la palabra cero, mientras que su parte baja corresponderá al byte 1). Cuando se conecta una tarjeta analógica de cuatro canales, se utilizan 8 bytes en total en el puesto de enchufe, porque en cada canal analógico se utiliza una palabra. La notación utilizada para el direccionamiento se observa en el siguiente capítulo.

Cuando el PLC se configura en varias filas, por medio de la utilización de interfaces IM315/IM316, la numeración de las filas de ampliación prosiguen en el puesto de enchufe situado más a la izquierda.



**Figura 1-19 Módulo de interfase IM 316**

La interfase de comunicación IM 316 permite configurar al autómata hasta 4 filas, en una distribución centralizada con una distancia máxima entre filas de 10 metros.

### **Guías para la conexión de elementos de entrada y salida**

Los módulos digitales pueden enchufarse en cualquier puesto (0-31). A un módulo digital ó desde un módulo digital, sólo es posible transmitir por canal dos estados: "0" (desconectado) ó "1" (conectado); esto puede realizarse con operaciones combinatoriales (lógicas) binarias ó con operaciones de carga y transferencia, para el acceso por medio de palabras.

Los módulos analógicos sólo pueden enchufarse en los puestos 0 a 7. La comunicación es un estado "0" ó "1" como en los módulos digitales, sino es posible comunicar por cada canal 65 536 informaciones diferentes (1 palabra = 2 bytes = 16 bits). Los módulos pueden direccionarse byte a byte ó palabra a palabra usando operaciones de carga o transferencia.

Es valida cualquier combinación de módulos de entrada y salida analógica en los puestos de enchufe 0 a 7.

## Capítulo 2 Principio de programación de controladores lógicos programables.

Los PLC son dispositivos programables y obedecen a principios de lógica Booleana, en este capítulo se analizarán los principios lógicos, así como las bases de la programación del PLC de la familia Simantic.

### Funciones lógicas

El concepto de señal binaria es el de aquella cantidad física que sólo puede adoptar dos posibles valores, representándolos típicamente como verdadero (o uno), y falso (o cero). Al álgebra que describe este tipo de valores se le llama álgebra booleana en honor a Charles Boole. Este tipo de álgebra describe a través de relaciones simples llamadas funciones booleanas, cómo se combinan dos o más variables binarias para dar como resultado un nuevo valor binario o booleano. Eventualmente el controlador programable tomará decisiones basadas en este tipo de funciones. En esta sección se verán los diferentes tipos de funciones, su definición, su simbología (representación), su significado y el cómo se puede utilizar para formar relaciones más complejas aún para la toma de decisiones en equipos de tipo digital como los PLC.

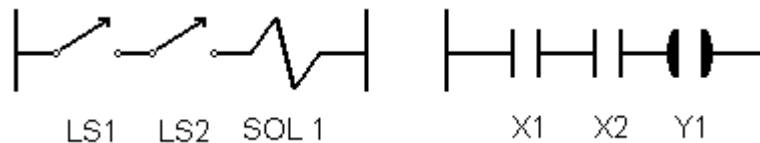
### La función AND (Y ó producto booleano)

La figura siguiente muestra el símbolo de una compuerta AND empleando para representar gráficamente la función AND, así como su tabla de verdad:



Figura 2-1 Función AND y su tabla de verdad

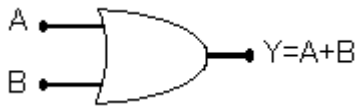
La salida de la compuerta AND es verdadera sólo si ambas entradas son verdaderas. El número de entradas de la compuerta AND es ilimitado, pero sólo tiene una salida. La función AND puede ser implantada de varias maneras, la figura que se muestra a continuación es la representación lógica, electrónica y en diagrama de escalera:



$$Y1 = X1 \bullet X2$$

### La función OR (O ó también suma booleana)

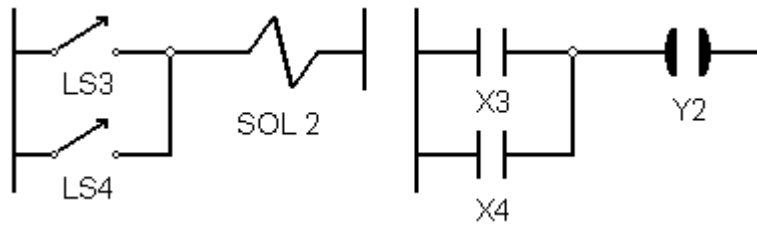
La siguiente figura muestra el símbolo de una compuerta OR así como su tabla de verdad:



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Figura 2-2 Función OR y su tabla de verdad

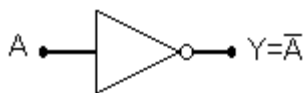
La salida de la compuerta OR es verdadera si una o más de sus entradas es verdadera. El número de entradas de la compuerta OR es ilimitado, pero sólo tiene una salida. La función OR puede ser implantada de varias maneras, la figura siguiente muestra la representación lógica, electrónica y en diagrama de escalera.



$$Y2 = X3 + X4$$

### La función NOT (Negación)

La figura siguiente muestra el símbolo empleado para representar gráficamente la función NOT, así como su tabla de verdad:



A	Y
0	1
1	0

Figura 2-3 Función NOT y su tabla de verdad

La salida de la función NOT es verdadera si la entrada es falsa. El resultado de operación NOT es siempre el inverso de la entrada y por lo tanto algunas veces es llamado inversor. La función NOT a diferencia de las compuertas AND y OR sólo tiene una entrada y raras veces se utiliza en forma aislada. En principio la función NOT no es tan fácil de visualizar como las funciones AND u OR. Sin embargo, al examinarla detalladamente es evidente su utilidad. Enseguida se presentan dos ejemplos que ilustran el uso de la función NOT:

En este momento es interesante retomar las tres características mencionadas, esto es:

- La asignación de "1" o "0" a una condición es arbitraria
- Un "1" es normalmente asociado con Verdadero, Ato, Encendido, etc.
- Un "0" es normalmente asociado con Falso, Apagado, etc.

Al examinar los puntos 2 y 3, es claro que una salida “1” está relacionado con la activación de algún dispositivo, mientras que una salida “0” con la desactivación del mismo. Esta convención puede ser empleada de manera inversa (lógica negada).

Aunque existen otras funciones como:

NAND: hacer una operación AND seguida de una NOT;

XOR: OR exclusiva, en la cual la salida es verdadera cuando sólo una de las entradas es verdadera, (en la OR cuando ambas entradas son verdaderas, la salida es verdadera, mientras que en la XOR es falsa); en realidad estas funciones se construyen a partir de las tres funciones básicas AND, OR y NOT.

Es importante señalar que cualquier función booleana por compleja que sea, puede ser representada únicamente en términos de dichas funciones.

## Principios de lógica y álgebra booleana.

### Operaciones básicas.

#### Compuertas básicas

Estas compuertas llevan a cabo funciones lógicas sencillas. Cada compuerta lógica es presentada a través de un símbolo, tabla de verdad y su expresión booleana.

#### Compuertas combinadas

Cualquier combinación de funciones de control se puede expresar en

$$NOT \Rightarrow Y = \bar{A}$$

$$AND \Rightarrow Y = A \bullet B$$

$$OR \Rightarrow Y = A + B$$

$$NAND \Rightarrow Y = \overline{A \bullet B}$$

$$NOR \Rightarrow Y = \overline{A + B}$$

Términos booleanos usando los tres operadores básicos.

$$Y = (\overline{A + B}) \bullet C$$

$$Y = A \bullet B + C$$

$$Y = (\overline{A \bullet B}) + C$$

$$Y = A + B \bullet \bar{C}$$

## Reglas del álgebra booleana

Funciones de control lógico pueden ser combinadas muy simples o extremadamente complicadas de las variables de entrada. Sin embargo, no importando su simplicidad deben satisfacer estas reglas básicas:

Ley de conmutatividad

$$A + B = B + A$$

Ley de asociatividad

$$A + (B + C) = (A + B) + C$$

$$A \bullet (B \bullet C) = (A \bullet B) \bullet C$$

Ley de distributividad

$$A \bullet (B + C) = A \bullet B + A \bullet C$$

$$(A + B) \bullet C = (A \bullet C) + (B \bullet C)$$

Ley de absorción

$$A \bullet (A + B) = A + A \bullet B = A(1 + B) = A$$

Leyes de DeMorgan

$$\overline{(A + B)} = \bar{A} \bullet \bar{B}$$

$$\overline{(A \bullet B)} = \bar{A} + \bar{B}$$

## Procedencia de los operadores y agrupación de signos.

El orden de prioridad en expresiones booleanas es:

$$NOT(\bar{x})$$

$$AND(\bullet)$$

$$OR(+)$$

A menos que se haya indicado la agrupación de signos mediante el uso de paréntesis, corchetes o llaves.

Cuando se usa una agrupación de signos para asegurar el orden apropiado de evaluación de una expresión primero se evalúan las expresiones entre paréntesis (), después las expresiones entre corchetes [ ] y finalmente las expresiones entre llaves {}.



### El concepto de señales binarias

El concepto de las señales binarias no es una idea nueva, de hecho es una concepción que se ha presentado desde hace mucho tiempo. Básicamente se refiere a la idea de muchas situaciones en las que señales sólo presentan dos estados, encendido-apagado, abierto-cerrado, activado-no activado, alto-bajo, etc. sólo por mencionar algunos ejemplos.

Estos dos estados pueden ser la base para la toma de decisiones, y puesto que se pueden relacionar fácilmente con el sistema de numeración binaria, constituyen uno de los bloques funcionales fundamentales de los controladores programables. Los dos elementos básicos del sistema de numeración binaria son: "1", el cual representa la presencia de la señal o la ocurrencia de un evento, por el contrario el "0" representa la ausencia de señal o la no ocurrencia de un evento. Enseguida se presentan algunos ejemplos de la utilización del concepto de señales binarias:

"1"	"0"	Ejemplo
Encendido	Apagado	Alarma
Abierto	Cerrado	Válvula
Suena	No suena	Campana
Presencia	Ausencia	Indicador de límite
En marcha	Detenido	Motor
Ilumina	No ilumina	Lámpara

Tabla 2-1 Ejemplos de señales binarias

### Simbología de contactos

Los elementos empleados para formar circuitos lógicos de control tanto en controladores programables como en sistemas de lógica alamburada con relevadores, conceptualmente operan de una manera similar. Entre estos elementos básicamente se tienen contactos de dos tipos, normalmente abiertos (NO) y normalmente cerrados (NC) y bobinas, estas últimas se utilizan para abrir o cerrar contactos (activar y/o desactivar dispositivos conectados a la misma, según sea el caso). La simbología utilizada en ambos casos es exactamente la misma, aún cuando no se cuenta con estándares y/o normas para ésta. La diferencia esencial estriba en que para los relevadores los contactos son físicos y en los PLC's son lógicos. Los símbolos comúnmente empleados se muestran en la siguiente figura:

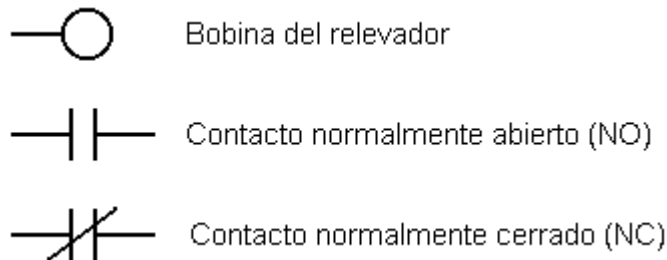


Figura 2-4 Símbolos empleados en los diagramas de escalera

Los contactos y las bobinas representan el conjunto básico de instrucciones para formar los diagramas de escalera, con excepción de las referentes a temporización y conteo.

## Contactos normalmente abiertos

Estos contactos representan cualquier tipo de entrada a la lógica de control, y puede ser el cierre de un interruptor, algún sensor conectado, un contacto conectado a una de las salidas o un contacto de alguna de las salidas internas. Su principio de operación es como sigue: cuando la entrada o salida asociada al contacto es referida se busca una condición de “encendido”, si su estado lógico es “1” el contacto se cerrará permitiendo así el flujo de corriente a través de él. Contrariamente si su estado lógico es “0” el contacto se abrirá con lo cual no habrá flujo de corriente a través del mismo.

## Contactos normalmente cerrados

Similar a lo referido en el párrafo anterior este tipo de contactos se pueden referir tanto a una entrada como a una salida del sistema. Su operación difiere en que estos contactos tienen un estado inicial cerrado; cuando la señal referida a este contacto se presenta, el contacto pasa de ser un corto circuito entre sus terminales a un circuito abierto, lo cual evita el flujo de corriente a través de éste, cuando la señal referida a este contacto no está presente, el contacto permanecerá como un corto circuito.

## Lenguajes de programación

Existen básicamente cuatro tipos de lenguajes de programación, comúnmente utilizados para desarrollar aplicaciones con controladores programables:

- a) Diagrama de escalera.
- b) Bloque de funciones.
- c) Lista de instrucciones.
- d) Alto nivel (BASIC específicamente)

Estos lenguajes pueden ser agrupados en dos categorías. El diagrama de escalera y el de bloques de funciones, forman los lenguajes básicos del PLC, mientras los otros son considerados como lenguajes de alto nivel.

Los diagramas de escalera se pueden considerar como el lenguaje básico (de alto nivel), el cual consiste de un conjunto de símbolos que permiten formar expresiones lógicas para llevar cabo la toma de decisiones. Los bloques de funciones y la lista de instrucciones están en la categoría de lenguajes funcionales (simbólicos), el primero de ellos está formado por un conjunto de bloques que realizan funciones y expresiones lógicas (AND, OR, NOT, etc.) y la lista de instrucciones consiste de un grupo de enunciados (statements), del tipo AND, OR, IF, IF-THEN, IF-THEN-ELSE, SET, RESET. Etc. Con los tres tipos de lenguajes se pueden formar funciones lógicas de control, pudiendo ser desde muy simples hasta altamente complejas según las necesidades y/o requerimientos de la aplicación considerada.

Estos lenguajes varían en extensión y diversidad de un control a otro, así como sus opciones y facilidades para el desarrollo. En este último rubro se cuenta desde los programadores portátiles de mano hasta las sofisticadas interfaces a través de computadoras personales, mediante el uso de compiladores cruzados (cross-compilers) en los que el desarrollo y la programación de aplicaciones se lleva a cabo en lenguajes algorítmicos de alto nivel, como C, C+, PASCAL, etc.

En este momento es necesario hacer mención que los lenguajes simbólicos (tales como diagrama de escalera y bloques de funciones) cuentan con bloques y/o funciones para conteo, temporización y comparación, pudiendo tener una buena variedad de opciones, tales como contadores hacia arriba, hacia abajo, distintas unidades básicas de tiempo, típicamente desde milisegundos hasta segundos dependiendo del tiempo de scan del controlador, así como comparaciones del tipo, mayor que, menor que, mayor o igual a, menor o igual a, etc.

El diagrama de escalera consta de un conjunto de instrucciones simbólicas que son usadas para crear el programa del PLC. Se compone básicamente de 5 tipos de instrucciones que incluyen símbolos tipo relevador, temporizadores, contadores, aritméticos, manipulación de datos,

transferencia de datos y control de programa. La función principal del programa en diagrama de escalera es controlar las salidas basada en condiciones de entrada.

El control es llevado a cabo por el uso de renglones de escalera; un renglón de escalera consiste de un conjunto de condiciones de entradas por símbolos de contacto y una instrucción de salida al final del renglón, representado por símbolos de bobina (terminales). Cuando se programa, cada contacto y bobina se hace una referencia con un número de dirección, el cual identifica cual entrada esta siendo evaluada o que salida esta siendo controlada. Los contactos pueden ser colocados en configuración serie, paralela o una combinación serie y paralelo. Para que una salida sea energizada o activada, al menos un camino de contactos debe ser cerrado, es decir, que las condiciones del renglón son verdaderas. A un camino completamente cerrado se le denomina lógica continua. Cuando existe una lógica continua en por lo menos un camino se dice que la condición del renglón es verdadera y si no existe un camino continuo se dice que la condición del renglón es falsa.

Aunque las instrucciones y símbolos pueden diferir de un controlador a otro, las instrucciones son genéricas y pueden aplicarse prácticamente a todos los PLC.

## Lenguaje STEP 5

La familia Simatic está integrada por los siguientes PLC: S5-90U, S5-95U, S5-100U, S5-115U/H/F, S5-135U, S5-155U/H. Todos estos poseen un mismo lenguaje de programación llamado STEP 5.

El STEP 5 es un interprete que sirve de interfase entre el operador y la CPU, indicándole a esta ultima las acciones que deben ejecutarse, este lenguaje emplea mnemónicos para todas las operaciones. Aún cuando este lenguaje no es de alto nivel, tampoco representa un código de máquina, por lo cual se puede considerar un lenguaje ensamblador.

El lenguaje de programación STEP 5 se encuentra en los aparatos de programación que ofrece la compañía Siemens, los cuales tienen diversas capacidades de visualización, diagnóstico y utilerías. Estos programadores pueden tomar la forma de aparatos manuales (PG 605), de tipo Laptop (PG 710/720), de tipo de computadora portátil (PG 750/740), y también se dispone del programa para ser instalado en una computadora personal del tipo PC-AT. Los programadores de la serie 7xx permiten al usuario elegir la forma de representación del programa, entre diagramas de escalera, bloques de funciones y listas de instrucciones, así como el idioma, entre inglés, español y alemán. Esto le da una mayor flexibilidad, pues el usuario puede utilizar la forma que más le convenga.

La Programación de un sistema Simantic S5 puede efectuarse tanto en modo fuera de línea como en línea, en este último, los cambios se hacen sin necesidad de parar el PLC.

## Tipos de Representación.

Como se expuso anteriormente, en principio los PLC han sido diseñados para facilitar el trabajo de técnicos, entre otras ventajas. Esto da origen a un lenguaje de programación gráfico, orientado a esquemas ó diagramas eléctricos. Para el STEP 5 esto se cumple, y adicionalmente se dispone de otras dos formas de representación de un programa que son el de bloque de funciones y el de lista de instrucciones.

El diagrama de escalera es un lenguaje de programación gráfico, orientado a esquemas de lógica de relevación. Fue el primer lenguaje utilizado en un PLC dada su facilidad de comprensión que daba a sus usuarios. Esta forma de representación es fácil de interpretar y se recomienda para todas las funciones de lógica combinatorial y parte de las aplicaciones lógica secuencial.

La programación en bloques de funciones es muy similar a la elaboración de diagramas lógicos con compuertas. En este tipo de representación es posible negar una entrada, estas compuertas lógicas se aplican en forma de cajas, cada caja contiene un mnemónico para indicar su función. En esta forma se puede programar casi todas las funciones básicas, pero no las avanzadas, ni las complementarias.

La lista de instrucciones (STL), es una forma de representación en la cual un programa será escrito basándose en mnemónicos adquiriendo al final el aspecto de un programa hecho en lenguaje ensamblador. Cada línea de comando está relacionada con un código de operación, un operando, una dirección, todo lo cual instruye a la CPU de lo que debe hacer. Esta forma de representación es la más conveniente para la programación de sistemas, pues permite utilizar todos los recursos de que se dispone en el PLC Simatic.

El programa STEP 5 permite también la visualización de un programa en cualquiera de las tres formas de representación, sin importar cómo se haya creado el programa originalmente. Debe notarse que algunos comandos en STL no tienen equivalentes en LAD o CFS.

### Definiciones

Antes de explicar los comandos del STL definiremos un grupo términos utilizados en un programa.

### Ciclo de Barrido

El PLC Simatic corre permanentemente en una secuencia definida, que se conoce como ciclo de barrido. Esta sucesión de eventos se muestra a continuación:

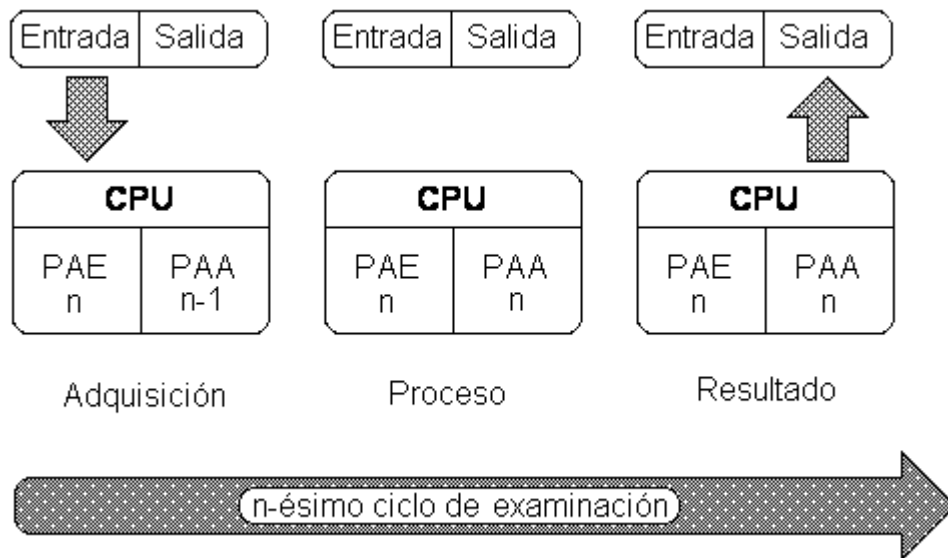


Figura 2-5 Ciclo de barrido del PLC

En el primer paso, la adquisición, se capturan las señales de entrada provenientes de todas las tarjetas conectadas al bus. Esta información se guardará en una área de memoria conocida como *Imagen de los Procesos de Entrada*, o PAE. En el segundo paso, se procesan las informaciones almacenadas en toda la memoria, de acuerdo al programa de aplicación que se ha cargado en el PLC y a medida que se ejecute dicho programa, se irá modificando los valores contenidos en una área de memoria conocida como *Imagen de Proceso de Salidas* o PAA. El tercer paso consiste en utilizar el bus para transferir la PAA hacia las tarjetas de salida que se tienen conectadas.

En resumen este ciclo de barrido se compone de tres pasos:

- Adquisición de la imagen de entrada de PAE
- Proceso de ejecución del programa de aplicación
- Salida o transferencia de la PAA a las interfaces de salida.

## Operación

Indica lo que se debe hacer. Es representada por un mnemónico ( que puede ser una ó hasta tres letras ó símbolos) ó por un símbolo gráfico.

## Operando

Indica la variable sobre la cual se ejecutará la operación. Los tipos de operandos pueden ser entradas, salidas, banderas ó datos.

## Puesto

Es la posición física en que se encuentra ubicado un módulo, a la derecha de la CPU.

## Canal

Es un número que indica la ubicación de un dispositivo externo, dependiendo del borne en el cual este conectado y del tipo de módulo, se determina el canal al que está conectado determinado dispositivo.

## Dirección

La forma en que la CPU reconoce a un operando es por medio de un número que por lo general se compone del puesto y el canal, separados por un punto decimal.

## RLO

Las operaciones lógicas (tipo bit) resultan en una condición cierta o falsa, la cual se almacena en un registro, el *Result of Logic Operation* (RLO). Las operaciones subsecuentes se realizan con este bit.

## Primera Consulta

La idea de primera consulta se refiere a la forma en que el RLO es afectado. La primera consulta siempre será independiente del resultado previo del RLO, por lo que el RLO guardará el resultado del examen descrito por la operación recién efectuada. Se consideran primeras consultas todas aquellas que se ejecutan después de la asignación de un resultado, o al iniciar un nuevo segmento del programa. Cuando no se tiene una primera consulta, el resultado del RLO se combina con el estado de la variable consultada.

## Instrucción

Se tiene una estructura específica para todas las líneas que componen un programa, el cual es: <operación, operando>. El operando se compone a su vez de un identificador de tipo y de la dirección.

A continuación se muestra un ejemplo:

Supongamos que se desea analizar el estado lógico de una entrada digital, que se encuentra en la primera tarjeta junto a la CPU, en el segundo canal. Para esto buscaremos el valor de la imagen de dicho punto en la memoria RAM. La instrucción que hace esto es la siguiente:

### A I 0.1

En donde:

Operación:	“A”	And ó scan por un 1	
Operando:	“I 0.1”	Entrada digital 0.1	
		Identificador: “I”	Entrada Digital
		Dirección: “0”	Puesto de enchufe 0
		Dirección: “0.1”	Segundo canal

### Estructura de Bit, Byte y Word

Algunas veces es necesario manejar simultáneamente más información que un solo bit por lo que el procesador trabaja con un conjunto de bits, al cual llamaremos byte. Un byte está constituido por ocho bits de información. Al conjunto de dos bytes se le denomina Word (palabra).

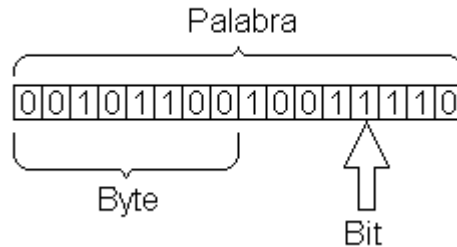


Figura 2-6 Estructura de la información

### Operaciones Básicas

Este tipo de operaciones pueden ser programadas con un PG605 ó con un PG710, permiten el control de las entradas y salidas digitales, así como los bits de banderas, operaciones con temporizadores y con contadores.

Operaciones	Operandos	Descripción
A AN O ON	I, Q, F, T, C	Consulta del estado de una señal digital
=	Q, F	Asigna el resultado de una operación lógica a un operador digital, del tipo salida o bandera
S R	Q, F, C	Coloca a un valor determinado el operador (S=1, R=0)
Si SD SF SS SE R	T	Funciones exclusivas de los temporizadores
CU CD	C	Operaciones con contadores
O( A( )	I, Q, F, T, C	Encadena los resultados de dos operaciones lógicas separadas

Tabla 2-2 Operaciones básicas del PLC

Estas operaciones comprenden la captura de valores digitales tipo bit, operaciones con temporizadores, contadores, la asignación del RLO a un bit, el SET y RESET de un bit, y la combinación AND, OR, NAND y NOR de bits, incluyendo las ecuaciones con paréntesis; estas operaciones binarias comprenden:

- Combinaciones
- Operaciones de memoria
- Operaciones con temporizadores
- Operaciones con contadores

### Combinaciones

Este primer juego de operaciones examina el estado de un operando tipo bit, y produce un valor que se almacena en el RLO. Este resultado se puede combinar con otras variables tipo bit y utilizar posteriormente para encadenación lógica de variables, o para ejecutar operaciones condicionales.

Las operaciones de combinación disponibles y sus equivalencias son las siguientes:

Operación	Descripción
A	AND lógico con el RLO, o scan por un 1
AN	AND negado con el RLO, ó scan por un 0
O	OR lógico con el RLO ó scan por un 1
ON	OR negado con el RLO ó scan por un 0

Tabla 2-3 Operaciones combinacionales

### Operación A (AND)

Esta operación equivale a la expresión lógica AND, debido a que modificará el valor almacenado en el RLO de acuerdo a la siguiente tabla:

RLO (t)	Entrada	RLO (t+1)
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 2-4 Operación A

Como se puede observar esta tabla de verdad es similar a la tabla de verdad de una función lógica AND, con la diferencia de que uno de los operandos se modifica después de la ejecución de la instrucción.

Esta tabla no es válida cuando se trata de una primera consulta. En este caso se interpretará como scan por un 1, de tal forma que el RLO tendrá el mismo valor que la variable consultada.

En un sentido práctico su significado puede tomarse como la investigación de si el valor de la entrada especificada es "1", es decir, si se activo dicha entrada. Equivale también a la combinación en serie de contactos normalmente abiertos.

### Operación AN (NAND)

Equivalente a NAND. En la primera consulta coloca un "1" en el RLO cuando la entrada en cuestión esta desactivada, o tiene un valor "0" y viceversa. En otros casos se aplica la siguiente tabla:

RLO (t)	Entrada	RLO (t+1)
0	0	1
0	1	1
1	0	1
1	1	0

Tabla 2-5 Operación AN (NAND)

La instrucción AN equivale a la conexión en serie de contactos normalmente cerrados. Al contrario de la función "A", examina si la entrada está desactivada, en cuyo caso el valor del RLO será 1.

### Operación O (OR)

Equivalente a OR. En la primera consulta asigna un valor "1" en el RLO cuando la entrada está activada, o tiene un valor de "1", para los demás casos se tiene la siguiente tabla:

RLO (t)	Entrada	RLO (t+1)
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 2-6 Operación O (OR)

Equivale a la conexión en paralelo de contactos normalmente abiertos en un diagrama de escalera.

### Operación ON (NOR)

Equivale a la operación NOR. En la primera consulta coloca un "1" en el RLO cuando la entrada en cuestión está desactivada, o tiene un valor de "0". Su tabla de verdad es la siguiente:

RLO (t)	Entrada	RLO (t+1)
0	0	1
0	1	0
1	0	0
1	1	0

Tabla 2-7 Operación ON (NOR)

En un diagrama de escalera esto equivaldría a la conexión en serie de dos contactos normalmente cerrados.



### Operaciones de Memoria

Existen tres operaciones de memoria que son el *SET*, *RESET* y la *asignación*. Estas tienen como operandos admisibles las entradas, salidas y las banderas. Para el uso de SET y RESET su ejecución dependerá del RLO, mientras que el uso de una asignación será del tipo incondicional, por ejemplo:

- S F 0.0:** El bit 0 de la bandera 0 será un "1", siempre y cuando el RLO sea "1"
- R F 0.1:** El bit 1 de la bandera 0 será un "0", siempre y cuando el RLO sea "1"
- = F 0.2:** El valor contenido en el RLO será asignado al bit 2 de la bandera 0, esta acción se lleva a cabo sin ninguna condición y además tiene prioridad sobre las demás, ya sea un SET o un RESET.

### Operaciones con Temporizadores

Por lo general se utiliza un temporizador para el arranque de diversas partes de un sistema, para lo cual se dispone de un grupo de operaciones, cada una de ellas tiene una finalidad específica.

El tiempo se especifica a través de una operación de carga, lo que se conoce como inicialización, con un número de acuerdo con el siguiente formato:

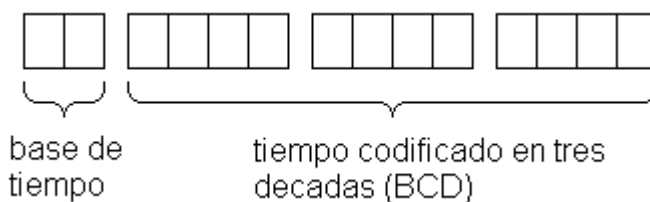


Tabla 2-7 Formato de la constante de tiempo de un temporizador

Una vez cargado el valor válido de tiempo en el ACCU 1, se puede proceder al arranque

Los temporizadores pueden también ser forzados a cero, para lo cual se utiliza la instrucción **R** (Reset). Esta operación provoca que el temporizador se inicialice inmediatamente, entonces el valor del contador empieza a disminuir de acuerdo a lo indicado en la base de tiempo.

Los posibles valores que puede tomar la base de tiempo son:

- 0: este multiplicador es por valores de 0.01 seg.
- 1: este es un multiplicador por 0.1 seg.
- 2: este es un multiplicador por 1 seg.
- 3: este es un multiplicador por 10 seg.

De esta forma la mínima cuenta es de 0.01 segundo, mientras que el mayor disponible es de 9990 segundos (2 horas, 46 minutos y 30 segundos), para mayores lapsos de tiempo se pueden conectar varios temporizadores en cascada, o bien se pueden usar combinaciones de contadores y temporizadores. Cuando se necesite una mayor precisión es recomendable emplear una base de tiempo menor, así por ejemplo:

Para contar 30 segundos se puede emplear esta instrucción:

**LKT 30.2**

Sin embargo una instrucción con más exactitud será la siguiente:

**LKT 300.1**

Para el primer caso se contarán 30 segundos en incrementos de un segundo (que es la base de tiempo), mientras que la segunda opción nos contará 300 décimas de segundo, lo cual nos da una mejor exactitud.

### SP Tipo pulso

En este tipo de operación el temporizador se arranca con un flanco de subida del RLO. Cuando el RLO tiene un valor de 0, la salida del temporizador se fuerza a 0.

**Ejemplo:** Deseamos que el valor máximo de la bandera F3.5 sea de 5 segundos cuando la entrada 3.1 este presente. El programa que realiza esto es el siguiente:

```

A I 3.1      ;Operación A aplicada a la entrada 3.1
LKT 500.0   ;Constante de tiempo 5 segundos (500 centésimas de segundo)
SP T 1      ;Función de pulso único del temporizador 1
A T 1       ;Operación A aplicada al temporizador 1
= F 3.5     ;Asignación a la bandera 3.5
    
```

Gráficamente la función será de la siguiente manera:

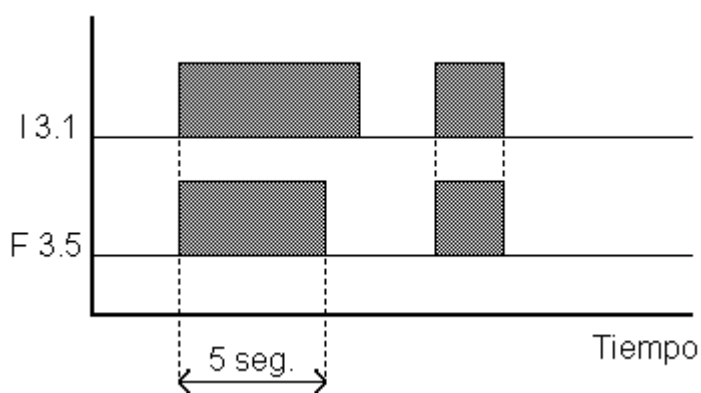


Figura 2-8 Operación de un temporizador tipo SP.

### SE Tipo pulso alargado

El temporizador se activa con un flanco de subida del RLO, una vez que se inicia el pulso el temporizador no será afectado por el contenido del RLO y el pulso durará lo establecido en la base de tiempo.

**Ejemplo:** Se desea activar la salida 1.2 por un lapso de 2.6 segundos, esto se llevará a cabo cuando el valor de la bandera 3.5 sea un cero lógico, el programa es el siguiente:

```

AN F 3.5 ;Operación AN aplicada a la bandera 3.5 (Se busca un 0)
LKT 260.0 ;Constante de tiempo de 2.6 seg. (260 centésimas de segundo)
SE T 2   ;Función de pulso prolongado del temporizador 2
A T 2    ;Operación A aplicada al temporizador 2
= Q 1.2  ;Asignación a la salida 1.2.
    
```

Gráficamente la visualización de la función será lo siguiente:

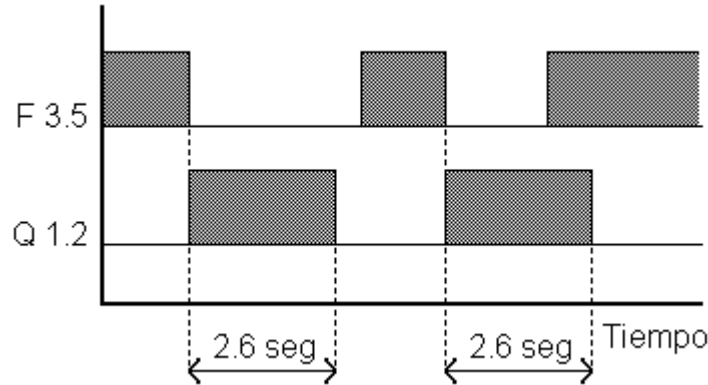


Figura 2-9 Temporizador tipo SE.

Como se puede observar el pulso de la salida 1.2 tiene una duración definida de 2.6 segundos, aún cuando la bandera 3.5 se desactive (cabe recordar que la bandera se activará cuando se tenga un nivel lógico de 0).

### SD Retardo de pulso al encendido

En esta función el temporizador inicia un pulso con un flanco positivo del RLO. Cuando el RLO toma un valor de 0, la salida del temporizador se fuerza a 0. La operación en la cual se examina el estado de la señal nos dará 1 cuando el temporizador haya terminado y el RLO todavía este activo.

**Ejemplo:** Se debe activar la bandera 5.0, 3 segundos después de que se haya presentado la entrada 1.0, utilizando el temporizador 1:

- A I 1.0** ;Operación A a la entrada 1.0 (Determina si esta presente)
- LKT 300.0** ;Constante de tiempo de 3 seg.(300 centésimas de segundo)
- SD T 1** ;Retardo de pulso en el temporizador 1
- A T 1** ;Operación A aplicada al temporizador 1
- = F 5.0** ;Asignación a la bandera 5.0

Gráficamente:

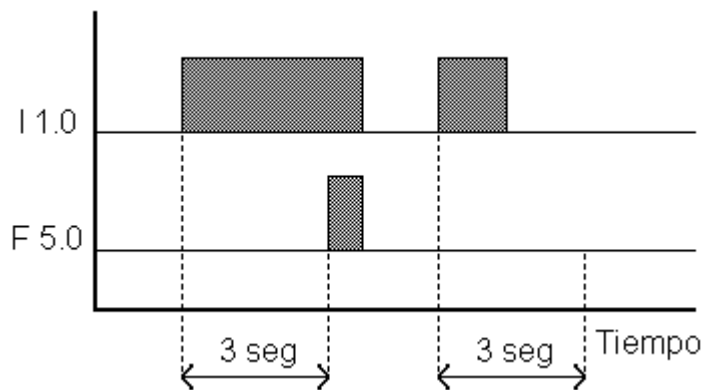


Figura 2-10 Temporizador tipo SD

Como podemos observar la bandera se activa 3 segundos después de que la entrada 1.0 se presenta, y continua activa mientras la entrada 1.0 este presente, sin embargo cuando la entrada 1.0 se presenta y se desactiva antes de que el pulso se active, éste no se presentará.

### SS Tipo retardo con memoria

Su funcionamiento es similar al tipo SD, se diferencia en que los cambios del RLO no reinician al temporizador.

**Ejemplo:** La salida 2.5 se deberá activar 4 segundos después que alguna de estas banderas 3.3, 2.0 tengan un valor de 1, o bien la bandera 3.4 tenga un valor de 0:

- O F 3.3** ;Operación O a la bandera 3.3
- O F 2.0** ;Operación O a la bandera 2.0
- ON F 3.4** ;Operación ON a la bandera 3.4
- LKT 250.0** ;Constante de tiempo de 2.5 seg.(250 centésimas de segundo)
- SS T 1** ;Retardo con memoria en el temporizador 1
- A T 1** ;Operación A aplicada al temporizador 1
- = Q 2.5** ;Asignación a la salida 2.5

*Nota:* El pulso se mantendrá activo, mientras no se aplique un RESET al temporizador 1. Gráficamente la función será:

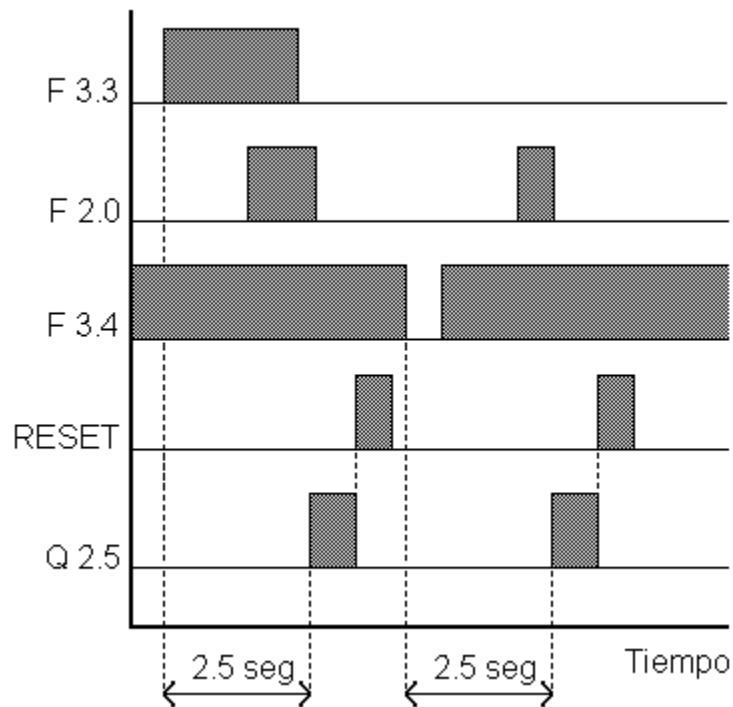


Figura 2-11 Temporizador tipo SS

### SF Tipo desconexión con retardo

En este tipo de pulso el arranque se inicia cuando el RLO manda un flanco de bajada, si el RLO tiene un valor lógico de 1, el temporizador se cargará con el valor de la constante de tiempo. La operación de examinar el estado de la señal resultará 1, mientras que el RLO de entrada sea 1, o bien el temporizador se encuentre corriendo.

**Ejemplo:** La salida 3.1 deberá permanecer energizada desde que la Palabra de entrada 10 (IW 10) sea igual a una constante hexadecimal de 0011, hasta un tiempo de 20 segundos después de esta condición, se utilizará el temporizador 2:

**LKH 0011** ;Carga una constante hexadecimal 0011  
**L I W 10** ;Se carga la palabra de entrada 10  
**! = F** ;Se compara si ambos valores son iguales  
**LKT 200.1** ;Carga la constante de tiempo (20 segundos)  
**SF T 2** ;Se aplica el pulso de retardo de conexión al temporizador 2  
**A T 2** ;Operación A aplicada al temporizador 2  
**= Q 3.1** ;Asignación a la salida 3.1

Gráficamente:

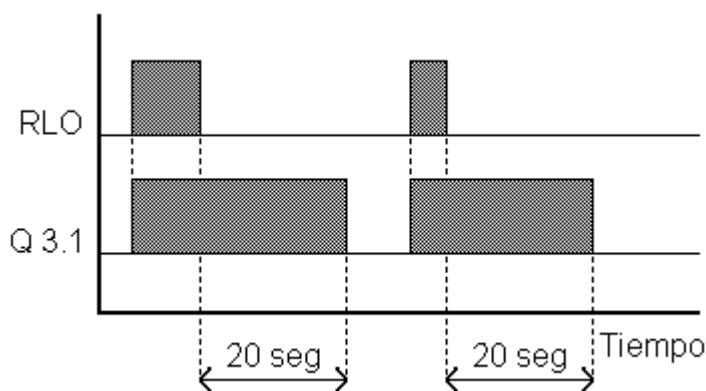


Figura 2-12 Temporizador tipo SF

## Operaciones con contadores

En los PLC se tienen implementados elementos de conteo como parte de la memoria, lo cual posibilita el uso de hasta 32 contadores en una CPU 102. Debido a que todos los contadores son por software, la frecuencia de utilización está limitada a la del ciclo de examinado.

Para el empleo de un contador se dispone de una entrada de conteo hacia arriba y una entrada de conteo hacia abajo. Se puede inicializar el contador con valor arbitrario, dicho valor se codifica en BCD y el rango es de tres décadas, iniciando en 000 y teniendo un valor final de 999. Al llegar al máximo valor el contador no se borra automáticamente, sino que preserva este valor aún cuando siga recibiendo instrucciones de incrementar la cuenta por medio de pulsos de entrada. Se tienen cuatro formas de operar los contadores, que son: *SET*, *RESET*, *incremento* y *decremento de la cuenta*.

La operación SET se ejecuta en el flanco de subida del RLO, en este instante se transfiere el número contenido en el ACCU 1 hacia la localidad de memoria que constituye el contador. Este será el valor inicial del contador, a partir de éste se podrá incrementar o decrementar la cuenta.

La operación RESET borra el valor contenido en el contador y se mantendrá activo mientras el RLO tenga un valor lógico de 1, durante este tiempo los pulsos de entrada no tienen efecto en el contador.

Generalmente el contador actuará con los flancos de subida de una entrada digital, de una bandera ó de una operación de comparación.

Ejemplos de contadores:

- RESET (borrado) :
  - A I 1.0** ;Operación que condiciona el borrado
  - R C 1** ;Reset aplicado al contador 1
  
- Carga de un valor inicial (100) en el contador 2:
  - A I 1.3** ;Operación condicional de la carga
  - LKC 100** ;Carga el valor de BCD 100 en el ACCU 1
  - S C 2** ;Carga el valor al contador 1
  
- Cuenta hacia arriba cada vez que la entrada 1.2 recibe un pulso:
  - A I 1.2** ;Se busca un 1 en la entrada 1.2
  - C U C 1** ;El contador 1 incrementará la cuenta

Cuenta hacia abajo cada vez que se active la salida 1.4

- A Q 1.4** ;Se monitorea si hay un 1 a la salida
- C D C 1** ;El contador 1 decrementará la cuenta

### Operaciones Digitales con Palabras

Estas son un conjunto de instrucciones que operan con palabras (un dato de 16 bits), por lo general las palabras en PLC tienen el siguiente formato:

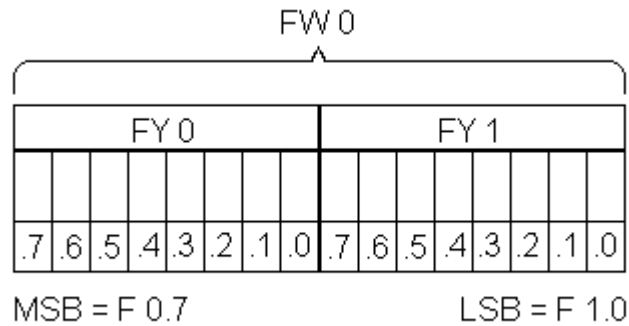


Figura 2-13 Formato de la palabra

Observamos que una palabra consta de dos bytes, y que están ordenados de tal manera que el byte más alto es aquel de numeración par y el byte bajo es el número impar siguiente. Por lo general la numeración de las palabras comienza en 0, en donde FW0 = FY0 + FY1, y así sucesivamente el orden será FW2, FW4, FW6..., esta forma de interpretar a las banderas es la razón por la cual solamente se deben direccionar palabras pares.

En los bloques de datos DB las palabras se organizan de manera distinta, por lo que su uso es más sencillo. En los DB es permitido direccionar las DW como 0, 1, 2, ..., hasta la longitud máxima que permita el bloque de datos.

### Operaciones de carga y transferencia

Estas operaciones utilizan operandos tipo palabra, es decir, de 16 bits, los cuales se manipulan con la ayuda de dos registros en la CPU, conocidos como ACCU 1 y ACCU 2.

La operación de carga tiene el efecto de copiar el valor del operando en el ACCU 1, lo cual provoca el desplazamiento del contenido previo del ACCU 1 hacia el ACCU2.

Existe también la posibilidad de cargar un número codificado en BCD, esto es útil para las operaciones con contadores y temporizadores; esta instrucción debe aplicarse con restricciones, ya

que al comparar un número de BCD contra uno de punto fijo, puede provocar resultados inesperados.

Los acumuladores pueden cargarse con información procedente de las entradas digitales, las banderas, palabras de datos, temporizadores, contadores e incluso salidas; también se permite el empleo de constantes, las cuales pueden ser de punto fijo, configuraciones binarias, números hexadecimales ó caracteres alfanuméricos (ASCII).

La operación de transferencia copia el contenido del ACCU 1 en el operando que se haya especificado, estos operandos pueden ser entradas, salidas, banderas ó datos. Esta operación no modifica el contenido del ACCU 1, ni de los ACCU 2.

### **Ejemplos:**

**L DW 10** ;Carga en el ACCU 1 el valor contenido en la palabra 10.  
**L C C 5** ;Carga un valor codificado en BCD al contador 5  
**L KB 130** ;Carga la constante de punto fijo 130  
**L KH 8AF9** ;Carga la constante hexadecimal 8AF9  
**L KF -24000** ;Carga una constante de punto fijo -24000 en forma de complemento a dos  
**T DW 20** ;Copia el contenido del ACCU 1 a la palabra de datos 20 del módulo de datos activo  
**T QW 10** ;Copia el contenido del acumulador ACCU 1 en la palabra de salida 10

Hay que observar que el operando es interpretado de acuerdo al operador, de manera que las instrucciones:

**L KH 000A**  
**L KF +10**  
**L KM 00000000 00001010**

Tendrán la misma función dentro de una operación de comparación ó en operaciones aritméticas de punto fijo de 16 bits.

### **Combinaciones digitales**

En estas operaciones se utilizan como operandos la información depositada en los acumuladores, los cuales se combinarán bit a bit.

Estas operaciones son:

**AW** ;Combinación AND bit a bit de los acumuladores.  
**OW** ;Combinación OR bit a bit de los acumuladores.  
**XOW** ;Combinación OR exclusiva bit a bit de los acumuladores.

En el siguiente ejemplo se aplican las funciones anteriores:

**L I W 10** ;Se carga el contenido de la palabra 10 en el ACCU 1.  
**L K H 03BF** ;Se carga la constante hexadecimal 03BF al ACCU 2.  
**A W** ;Combinación AND de ambos acumuladores.  
;El resultado se almacenará en el ACCU 1  
**L K H 0A80** ;Se carga la constante hexadecimal 0A80 al ACCU 2.  
**O W** ;Combinación OR de ambos acumuladores.  
**T DW 20** ; El resultado se transfiere a la palabra de datos 20.

Gráficamente:

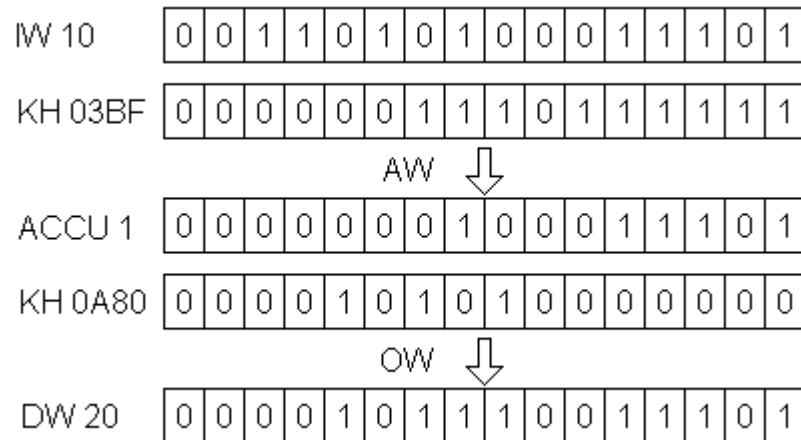


Figura 2-14 Ejemplo de operaciones combinacionales con palabras

Se debe enfatizar que no se dispone de combinaciones NAND y NOR, posteriormente se mostrará la forma de realizar estas combinaciones por otros métodos.

### Operaciones aritméticas

Se tienen solo dos operaciones aritméticas, que son la adición y la sustracción, ambas se realizan con números de punto fijo:

- + F ;Adición, se suman los contenidos de ambos acumuladores.
- F ;Sustracción, se resta el contenido del ACCU 1 al ACCU 2.

Las operaciones de multiplicación y división solo son posibles por medio de bloques de funciones, también como números de punto fijo.

**Ejemplo:**

- L I W 4 ;Cargar el contenido de la palabra de entrada 4
- L I W 6 ;Carga el contenido de la palabra de entrada 6
- + F ;Suma el contenido de ambos acumuladores
- L DW 12 ;Carga la palabra de datos 12
- F ;Resta en contenido de ambos acumuladores
- T DW 14 ;Transfiere el resultado a la palabra de datos 14



El movimiento de los contenidos de los acumuladores se muestra en la siguiente figura:

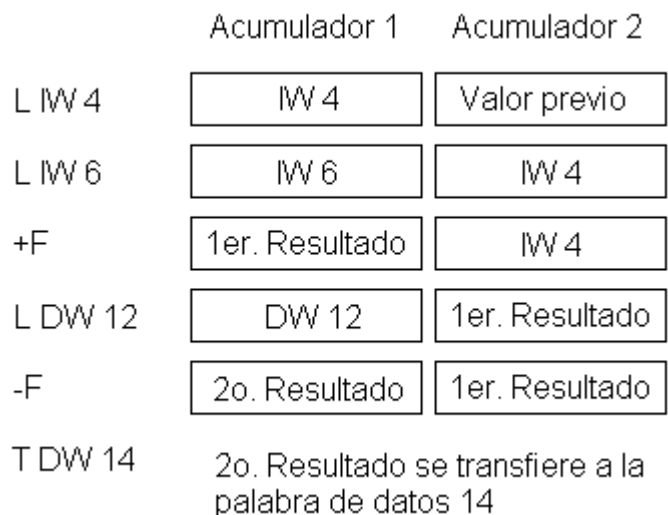


Figura 2-15 Detalle del contenido de los acumuladores en las operaciones de suma y resta

Estas operaciones actúan sobre los acumuladores, en los cuales se han cargado previamente los operandos, las operaciones se llevan a cabo incondicionalmente y generan un resultado tipo bit, que se almacena en el RLO. En caso de que la condición se verifique el RLO contendrá un 1 lógico, en otro caso el valor será de 0.

En este tipo de operaciones se compara el número del acumulador 2 con el valor del acumulador 1, siendo estos números del tipo de punto fijo. Esto nos indica que el valor HF será menor que 0FFF:

Las operaciones de comparación son las siguientes:

- ! = F** ; compara si ambos acumuladores tienen el mismo valor
- >< F** ;compara si ACCU 1 es diferente a ACCU2
- > F** ;compara si ACCU 2 es mayor que ACCU 1.
- > = F** ;compara que ACCU 2 sea mayor o igual a ACCU 1.
- < F** ;Compara si ACCU 2 es menor que ACCU 1.
- < = F** ;compara si ACCU 2 es menor o igual a ACCU!

**Ejemplo:**

**L F W 11** ;Carga la bandera 11 del tipo palabra

**L D W 25**;Se carga la palabra de datos número 25

**! = F** ;Se compara a los acumuladores

**JC PB 1** ;Brinco al PB1, si ambos acumuladores son iguales.

El salto al PB1 estará condicionado a que la palabra FW 15 sea igual a la palabra DW 25 del bloque de datos activo.

**Operaciones complementarias**

En el CPU 102 existe un grupo de operaciones que solo se pueden programadas en bloques de funciones y que permiten un manejo más complejo de la memoria del sistema.

## Operaciones de transformación

**CFW** Complemento a 1. Se invierte bit a bit el contenido del ACCU 1.

**CSW** Complemento a 2. Se invierte bit a bit el contenido del ACCU 1 y se le suma la palabra 0001H.

## Operaciones de desplazamiento

Las operaciones que se indican a continuación actúan sobre el valor contenido en el acumulador, de la siguiente manera:

**SLW n** Desplaza a la izquierda “n” posiciones el contenido del ACCU 1. Los bits que quedan libres se rellenan con ceros. Los bits desplazados se pierden.

### *Ejemplo:*

**LKH 00FF** ;Carga la constante hexadecimal 00FF  
**SLW 4** ;Desplazamiento a la izquierda cuatro posiciones.  
**T QW 6** ;Copia el contenido del ACCU 1 en la palabra de salida 6.

En este caso el resultado de la operación será **FF00**.

**SRW n** Desplaza a la derecha “n” posiciones el contenido del ACCU 1. Al igual que la instrucción **SLW**, los bits de relleno serán ceros, mientras que los bits desplazados se pierden.

## Operaciones de salto

En los bloques de funciones es posible colocar etiquetas, con las cuales se logra un control óptimo del flujo del programa, éste flujo se lleva a cabo con las operaciones de salto:

**JU** ;Salto incondicional a la etiqueta.  
**JC** ;Salto condicionado a la etiqueta.  
**JZ** ;Salto a la etiqueta si el resultado es cero.  
**JN** ;Salto a la etiqueta si el resultado no es cero.  
**JP** ;Salto a la etiqueta si el resultado es positivo.  
**JM** ;Salto a la etiqueta si el resultado es de signo negativo.  
**JO** ;Salto a la etiqueta si ocurre un desbordamiento.

### *Ejemplo:*

**L DW 10** ;Carga el contenido de la palabra de datos 10 al ACCU 1.  
**T FW 2** ;Transfiere el contenido del ACCU 1 a la bandera de datos 2.  
**A F 2.7** ;Operación A aplicada al bit 7 de la bandera de datos 2.  
**JC = STP** ;Salto a la etiqueta STP, si el resultado es 1 en la operación anterior.

## Organización de un programa en STEP 5

Un programa de mando puede ejecutarse de diversas formas:

- Ejecución cíclica
- Ejecución controlada por alarmas, esta función se realiza en los CPU 103.
- Ejecución controlada por tiempo, igualmente esta función se realiza a partir del CPU 103.

El programa de mando se ejecuta generalmente de forma cíclica. Para ello, se lee paso a paso la memoria del programa. Esto se hace a través del OB 1.

Como se explico con anterioridad, en la ultimo parte del ciclo de barrido, tras la ejecución del programa se corre un ciclo de escritura, durante el cual los datos del PAA se transfieren a los módulos de salida, al mismo tiempo se transfiere hacia la PAE los datos actuales de los módulos de entrada. Con esta PAE actualizada vuelve a realizarse la ejecución la ejecución del programa.

Antes de la ejecución de la primera línea del OB 1 se inicializa un tiempo de vigilancia (perro guardián), si este tiempo expira, el autómata pasa forzosamente a "STOP", bloqueando todos los módulos de salida. Esto ocurre cuando se ha programado un lazo sin fin o cuando surge una avería en el CPU. El tiempo de vigilancia sólo puede modificarse en el CPU 103, a través del OB 31.

Esta ejecución cíclica del OB 1 puede ser interrumpida cuando alguna señal de entrada ha sido programada como alarma para ejecutar instrucciones adicionales. Con esto se dice que el programa está controlado por alarmas. En adición a esto es posible hacer interrupciones a intervalos de tiempo definidos, lo que se conoce como programa controlado por interrupciones de tiempo. Una vez ejecutadas las instrucciones de la interrupción, el CPU retorna al punto de interrupción en el programa cíclico y prosigue allí su ejecución.

## Tipos de programación

En la familia S5, el software se ha diseñado tipo modular, con lo cual se facilita la elaboración de programas, a la documentación de los mismos y estimula la solución de problemas en una forma estructurada. Con este fin se dispone de 5 tipos de módulos ó bloques de programas que son:

- **OB** ;Bloque de organización.
- **PB** ;Bloque de programas.
- **FB** ;Bloque de funciones.
- **SB** ;Bloque de pasos.
- **DB** ;Bloque de datos.

Cada bloque ó módulo esta dividido a su vez por segmentos, hasta un máximo de 254 en cada bloque de programa.

La longitud máxima del programa está limitada por el espacio que se dispone en la memoria de programa (La capacidad de CPU 100 es de 1024 instrucciones, en el CPU 102 se tiene 2048 instrucciones y en el CPU 103 10240 instrucciones).

Una instrucción ocupa normalmente una palabra en la memoria de programa, aunque también hay instrucciones de dos palabras, lo que deberá tomarse en cuenta al calcular la longitud del programa. Todos los programas de aplicación se pueden organizarse en dos formas:

1. Programas lineales
2. Programas estructurados.

### Programas lineales

En este caso, el programa de mando está contenido en un solo módulo. La programación lineal solo es posible en el módulo de programa **PB 1** ó en el módulo de organización **OB 1**. El programa se ejecuta cíclicamente. Esta programación es útil solamente en programas pequeños.

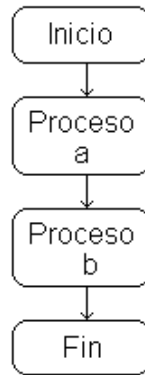


Figura 2-16 Ejemplo de un programa lineal

### Programas estructurados

En esta forma el programa de mando está organizado en diferentes bloques ó módulos. La estructura principal del programa es el bloque de organización **OB 1**, por lo que siempre que se quiera escribir un programa es necesario programar también el OB 1. El OB 1 se procesa cíclicamente.

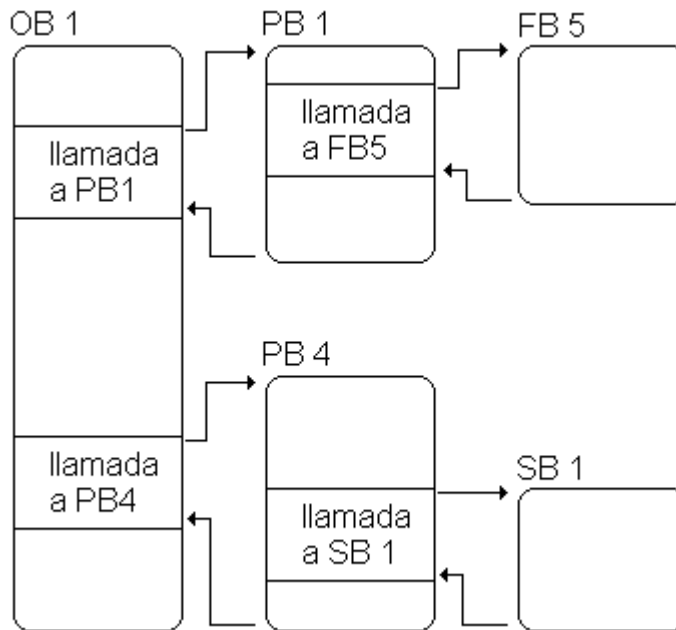


Figura 2-17 Programación estructurada

Este tipo de programación es el más recomendado cuando el programa de control del PLC va a ser desarrollado por un equipo de programadores, pues permite delimitar el área del trabajo de cada uno.

## Tipos de módulos de programa

Es posible programar instrucciones en los siguientes diferentes módulos:

- Módulos de organización (OB)
- Módulos de programa (PB)
- Módulos de paso (SB)
- Módulos de funciones (FB).

En los módulos de datos (DB) solo sirven para depositar datos.

## Módulos de organización (OB)

Constituyen la interfase entre el sistema operativo y el programa de mando, son llamados por el sistema operativo cuando se producen determinados eventos.

Se disponen de OB para las siguientes aplicaciones:

- Ejecución cíclica del programa **OB1**.
- Rutina de arranque: **OB 21** y **OB 22**.
- Fallo de batería: **OB 34**.
- Perro guardián: **OB 31**.

### Ejecución cíclica del programa usando el OB 1.

El sistema operativo llama cíclicamente el OB 1, la primera instrucción del OB 1 es el principio del programa de mando. La instrucción "BE" (*Block End*) del OB1 es la última instrucción del programa de mando.

### Módulos de organización para arranque: OB 21 y OB 22.

Con ambos OB es posible fijar el comportamiento del PLC antes de la primera ejecución del programa.

El OB 21 se ejecuta una sola vez tras el paso de "STOP" a "RUN", es decir, en el arranque del equipo.

El OB 22 se ejecuta una sola vez al conectar, siempre que el PLC estuviera en "RUN" antes de desconectar y no se presente ninguna causa de "STOP" al conectar, es decir el OB 22 organiza el comportamiento del reinicio automático después de una falla de la alimentación.

### Fallo de batería. usando el OB 34

El sistema operativo comprueba permanentemente la tensión de la batería. En este módulo se programa la reacción del PLC al ocurrir el evento de falla de batería. Si se da cuenta que ha fallado, por alguna razón, se procesará el OB 34 antes de cada ejecución cíclica del programa (OB 1).

### Perro guardián empleando el OB 31.

Un perro guardián "Watch dog", es un temporizador interno de la CPU que vigila el tiempo total de la ejecución del programa. Si el programa dura más que el tiempo de vigilancia de ciclo ajustado, el CPU pasa a STOP. Esto puede ocurrir cuando el programa es muy largo o se programa un lazo sin fin.

Llamando el OB 31 es posible reiniciar el conteo del perro guardián en cualquier punto del programa, esto es, volver a arrancar el tiempo de vigilancia

## Módulos de programa (PB).

En estos módulos es posible utilizar el juego básico de operaciones del lenguaje STEP 5 y sus tres representaciones posibles, dependiendo del programador que se empleé. Toda la lógica combinacional se programará en estos bloques.

### *Ejemplo:*

Se debe programar los módulos OB1 y PB1.

Programa del OB1:

```
OB1           ;declaración de bloque de organización 1
JU PB 1      ;salto incondicional al PB 1
BE           ;Fin del OB 1
```

Programa contenido en el PB 1:

```
PB 1         ;Declaración del bloque de programas 1
O I 0.1      ;Operación O aplicada a la entrada 0.1
O Q 3.0      ;Operación O aplicada a la salida 3.0
A I 0.7      ;Operación A aplicada a la salida 0.7
= Q 2.2      ;El resultado se asignará a la salida 2.2
BE           ;Fin del PB 1.
```

## Módulos de paso (SB)

Los mandos secuenciales se dividen en pasos individuales cuya sucesión es gobernada por algún conjunto de condiciones. Esta estructura en cadena debe fijarla el usuario programando módulos de paso. Esto contienen las instrucciones y las condiciones de prosecución.

Es posible programar 256 módulos de paso (SB0, SB1, ..., SB 255) con lo cual se dispone una cantidad doble de "módulos de programa".

## Módulos funcionales (FB)

En estos módulos se programan funciones de mando complejas o que se repiten con frecuencia. Para esto es posible utilizar operaciones avanzadas, además del juego básico de operaciones de combinación y comparación, con las siguientes características:

- Tienen un encabezado más largo
- Es posible utilizar las operaciones complementarias y de sistema.
- Solo son programables en STL.
- A partir del CPU 103 es posible asignar parámetros a los FB.
- Para ser llamados desde otro módulo es imprescindible que existan previamente, si no se emitirá un aviso de error en el programador.

## Módulos funcionales integrados

A partir de los CPU 102 se tienen grabados en la ROM 6 bloques de funciones, los cuales ejecutan diversas tareas. Estos bloques son: FB 240, 241, 242, 243, 250 y 251, realizan las tareas que se describen a continuación:

- Conversor de código B4 FB 240.
- Conversor de código 16 FB 241.
- Multiplicador 16: FB 242.
- Divisor 16 FB 243.

- Lectura y normalización del valor analógico FB 250.
- Salida de un valor analógico FB 251.

### Convertor de código B4

FB 240: Este módulo de programa permite transformar un número de 4 tétradas en BCD con signo en un número de punto fijo de 16 bits.

Aquellos números de dos tétradas deben transformarse previamente en cuatro tétradas antes de realizar la conversión. En el caso de que una tétrada no se encuentre dentro del margen BCD, se toma como valor cero y no se señala con bit de error.

Parámetro	Significado	Tipo	Asignación
BCD	Número BCD	I W	0... 9999
SBCD	Signo del número BCD	I (bit)	"1" para "-" "0" para "+"
DUAL	Número de punto fijo	Q W	16 bits

Tabla 2-8 Convertor de código B4

**Ejemplo:**

Se tomará el valor del contador 1 (C 1) y se transformará en un número de punto fijo que se escribirá en la palabra de datos 21 (DW 21). El signo del contador será almacenado en el bit 1 de la bandera 2. La bandera 0 se utiliza como registro intermedio.

```

LC C 1           ;Carga del valor del contador 1.
T FW 0          ;Transferencia a la Bandera 0 del ACCU 1.
JU FB 240       ;Salto al módulo de conversión.
NAME: COD :B4
BCD : FW 0      ;Convierte FW 0 a BCD.
SBCD : F 2.1    ;Comprueba el signo de la cantidad en el bit F 2.1
DUAL : DW 21    ;Convierte la cantidad a punto fijo y escribirla en la palabra de
                  datos 21
    
```

### Convertidor de código 16

FB 241: Este módulo funcional permite convertir un número binario de punto fijo (16 bits) en uno de BCD de cinco tétradas, considerando también el signo.

Parámetros	Significado	Tipo	Asignación
DUAL	Número binario	IW	-32768... +32767
SBCD	Signo del número BCD	O B I	"1" para "-" "0" para "+"
BCD1	Número BCD tétradas 0, 1, 2 y 3	Q B Y	4 tétradas
BCD2	Número BCD tétradas 4 y 5	Q W	2 tétradas

Tabla 2-9 Convertor de código 16

**Ejemplo:**

Se deberá mostrar en una pantalla de 5 dígitos el valor de una variable de punto fijo que se obtiene como resultado de una conversión A/D. Para esto es necesario que cada dígito sea codificado en BCD y que cada tétrada se transfiera a la salida correspondiente. La variable de

punto fijo está depositada en la bandera 17 y se dispone de 3 módulos de salida digital de 8 canales conectados en los puestos de enchufe 1, 2 y 3. En el puesto 0 se tiene conectada la entrada analógica.

Para este programa se utiliza el PB1 como se muestra a continuación:

```

PB1
JU FB 241 ;Salto a la subrutina de convertidor de código 16
NAME : COD 16 ;Identificador
DUAL: FW 17 ;Convertir el contenido de FW 17 a punto fijo.
SBCD: F 0.0 ;El bit del signo está en el bit F 0.0.
BCD1: QW 2
BCD2: QY 1
:BE
    
```

### Multiplicador 16

FB 242: Este módulo funcional permite la multiplicación de dos números de punto fijo de 16 bits, el resultado, también en formato de punto fijo, se almacena en dos palabras, en 32 bits.

Parámetro	Significado	Tipo	Ocupación
Z1	Multiplicador	I W	-32768... +32767
Z2	Multiplicando	I W	-32768 ... +32767
Z3 = 0	Consulta de cero	I B1	"0" cuando el producto es cero
Z32	Producto palabra alta	Q W	16 bits
Z31	Producto palabra baja	Q W	16 bits

Tabla 2-10 Multiplicador 16

#### Ejemplo:

Se desea la multiplicación de dos variables analógicas, provenientes de sensores de peso y velocidad, con estos datos se calculará el gasto másico de una banda transportadora. Estos datos se almacenan en los registros de bandera 1 y 2. El resultado debe almacenarse en las palabras de datos (DW) 26 y 27 del bloque de datos. Se tiene un piloto que se debe encender en el caso de que el gasto másico sea cero, está en el puesto 4 del canal 3.

Para esta implementación se utilizará el FB 242 en el PB1:

```

PB 1
:C DB 7
:JU FB 242
NAME: MUL: 16
Z1 :FW1
Z2 :FW2
Z3 = 0 ;Q 4.3
Z32 :DW 26
Z 31 :DW 27
:BE
    
```



## Divisor 16

FB 243: Este módulo permite la división de dos números de punto fijo. El cociente se expresa como un número de punto fijo al igual que el residuo.

Parámetro	Significado	Tipo	Ocupación
Z1	Dividendo	I W	-32768 ... +32767
Z2	Divisor	I W	-32768 ... +32767
OV	Sobre flujo	Q B I	"1" para desbordamiento
FEH	Indicador	Q B I	"1" para divisiones por cero
Z3 = 0	Consulta de cero	Q B I	"0" para cociente cero
Z4 = 0	Consulta de cero	Q B I	"0" residuo es cero
Z3	Cociente	Q W	16 bits
Z4	Residuo	Q W	16 bits

Tabla 2-11 Divisor 16

## Lectura y normalización de un valor analógico

FB 250. Este módulo lee un valor proveniente de un módulo de entrada analógica y lo transforma de modo que corresponda a un valor dentro de un margen designado por el usuario.

Parámetros	Significado	Tipo	Asignación
BG	Puesto de enchufe	DKF	0.. ..7
KNKT	Numero y tipo de canal	DKY	KY=x,y
OGR	Limite superior	DKF	-32768.. ..+32767
UGR	Límite inferior	DKF	-32768.. ..+32767
EINZ	-	-	-
XA	Valor de salida	QW	Valor analógico normalizado
FB	Bit de error	QB1	"0" cuando no hay error
BU	Margen desbordado	QB1	"0" cuando no hay error

Tabla 2-12 Lectura y normalización de un valor analógico

El parámetro KNKT indica el canal (x, puede tomar valores 0, 1, 2, 3), y el tipo (3: indica entrada de valor absoluto, como las tarjetas de entrada de corriente de 4 a 20mA; tipo 4, es una entrada unipolar; tipo 5 es una entrada de valor absoluto, bipolar de tipo 6, un número de punto fijo, bipolar).

## Salida de un valor analógico

FB 251: Este programa permite al usuario hacer una conversión D/A por medio de un módulo de salida conectado en algún puesto. Cualquier número de 16 bits puede utilizarse.

Parámetro	Significado	Tipo	Asignación
XE	Valor digital a transformar	IW	Cualquier número entre OGR UGR
BG	Puesto de enchufe	DKF	0.. ..7
KNKT	Número y tipo de canal	DKY	KY=x,y
OGR	Límite superior	DKF	-32768.. ..+32767
UGR	Límite inferior	DKF	-32768.. ..+32767
FEH	Bit de error de límite	QBI	"0" cuando no hay error
BU	Margen desbordado	Qbi	"0" cuando no hay error

Tabla 2-13 Salida de un valor analógico

El parámetro XE se interpreta como un número de complemento a dos.

Los datos en el parámetro KNKT solo pueden tomar dos estados cada uno, de modo que solo hay dos canales (canal: x = 0 ó 1): tipo representación unipolar (y = 0) ó representación en coma fija bipolar (y = 1)

El bit de error FEH se enciende en el caso de que UGR = OGR, también con número de canal o de puestos no válidos, o con tipo de canal no válido.

El bit de error BU indica que la palabra de entrada está fuera de los límites definidos por OGR y UGR, en cuyo caso XE tomará el valor del límite.

### Ejemplo:

Una tarjeta analógica con rango de salida de 1 a 5 Volts, se conecta en el enchufe 1 y se requiere una pendiente lineal de tal manera que para 5 Volts corresponda a una salida de 30000, mientras que para 0 se tenga 1 Volt a la salida (representación de punto fijo de 16 bits). Los errores se mostrarán en una tarjeta de salida conectada en el puesto de enchufe 0.

Programa:

**OB 1**

**JU FB 251**

**NAME :RLG :AA**

**XE :FW 4**

**BG :KF + 1**

**KNKT :KY 0.0**

**OGR :KF + 30000**

**UGR :KF + 0**

**FEH :Q 0.0**

**BU :Q 0.1**

**:BE**

## Módulos de datos

En estos módulos se depositan los datos, en palabras, que deben procesarse en el programa.

En cada DB es posible almacenar hasta 256 palabras de datos. Antes de crear un DB es preciso depositar en el ACCU 1 su longitud. La longitud máxima del DB a crear depende del espacio disponible en la memoria RAM. En caso de no haber suficiente memoria, el PLC ira a STOP.

Los datos contenidos en palabras de datos pueden tener diferentes formatos: configuración binaria, números hexadecimales, binarios, decimales ó bien caracteres alfanuméricos.

En el CPU 102 se dispone de 64 bloques de datos, los cuales solo pueden ser direccionados directamente. Los bloques DB0 y DB1 son reservados para el sistema y no deben ser utilizados.

Para acceder una palabra en un bloque de datos, se necesita la apertura del mismo. Solo un módulo puede estar abierto a la vez y para evitar conflictos el sistema operativo cierra el módulo inactivo cada vez que se abre otro. Un bloque puede abrirse automáticamente sólo cuando este regresa al módulo desde el que hizo la llamada. No existe ninguna instrucción capaz de cerrar los módulos.

### *Ejemplo:*

Se utilizará el bloque de datos 10 para guardar 90 valores que han sido capturados secuencialmente por la entrada analógica.

Utilizando el PB 0:

**L KF 90**

**G DB 10**

**C DB 10**

Para borrar toda la información de un bloque de datos se le da tamaño de cero:

**L KF 0**

**G DB 10**

Antes de llamar un bloque de datos es necesario haberlo creado previamente, en caso contrario el PLC se pasa a STOP.

## Documentación de programas

La documentación de un programa consiste en agregar información que ayuda a un mejor entendimiento de las funciones del programa.

Se puede incluir una línea de texto al lado de cada instrucción, en un campo máximo de 28 caracteres, para esto se presiona la tecla de tabulador para que el cursor se coloque al lado derecho de la pantalla, ó también con la tecla shift y la flecha derecha.

Se puede incluir un título en cada bloque, con la tecla shift y el número 5 del bloque de números en el teclado.

Una de las funciones de documentación de un programa es la que nos permite la programación con símbolos, es decir para cada punto de entrada o salida se le puede asignar un nombre alfanumérico. De esta forma, la entrada I 0.0 puede reconocerse dentro del programa como “detector de botella” ó “E1”. De igual modo, se pueden agregar etiquetas a las banderas, salidas, palabras, temporizadores y contadores.

Otra forma de documentación importante es la tabla de variables utilizadas. Esta es generada automáticamente por el software y nos da una visión del espacio libre en memorias y otros objetos de interés.

Adicionalmente a estas dos formas de documentación, se tiene la generación de la tabla de referencias cruzadas, que nos indica donde se utiliza un operando cualquiera. La referencia cruzada puede ser por operador, por bloque y puede ser llamada en línea con las funciones de prueba.

## Capítulo 3 Presentación de la aplicación

### Planteamiento del problema

De los temas vistos anteriormente, sobre programación del PLC Simatic S5 100U, se pensó en como se podría aplicar en la industria; por lo que se decidió diseñar y construir una maqueta que representara un proceso industrial. Se desea realizar de manera práctica y real un sistema dinámico que tenga como finalidad el llenado de envases, el sistema debe ser capaz de realizar esta tarea mediante una forma automatizada, para esto es necesario seleccionar primero las secciones que conformarían el sistema, considerando únicamente a las más relevantes. Se requiere en primera instancia pensar en una trayectoria o un recorrido que dirija a los envases a través de dichas secciones siguiendo una secuencia establecida y lograr el llenado de estos.

La primera sección contempla la necesidad de colocar los envases sobre la trayectoria del proceso de llenado con un determinado orden o secuencia.

En la siguiente sección se debe determinar si los envases se encuentran en la posición y en las condiciones necesarias para que continúe adelante en el proceso, en caso contrario, se debe realizar las medidas necesarias para corregir el desperfecto.

Una vez preparado el envase tendrá que ser colocado a una sección de llenado y tomar en cuenta el tiempo necesario que tarda en llenar la sustancia a cierto nivel y liberarlo a la siguiente sección.

Por último los envases llenos se deben retirar de la trayectoria e ir almacenando cada uno de estos sobre un lugar o depósito y observar que todos se encuentren al final del proceso para que automáticamente se detenga.

La realización del sistema implica el diseño de diferentes procesos ligados (mecánicos y electrónicos), para realizar un proceso automatizado.

## Desarrollo del Prototipo

Con el fin de llevar a cabo el control de un proceso en miniatura mediante el PLC, se analizaron varias opciones y finalmente se decidió implementar un proceso de llenado de botellas. Posteriormente se llevó un análisis para su implementación basándose en los siguientes criterios en el cual se incluyen los siguientes pasos:

### Bosquejo del prototipo

Se analizó cuales pudieran ser las secciones que intervendrían dentro del proceso, es obvio que para un proceso real de llenado se necesitan mas secciones, sin embargo el fin de la tesis es presentar una aplicación demostrativa de control de un proceso, después de varias opciones se llevo a la conclusión de que el proceso presentaría las secciones que se muestran a continuación en un bosquejo inicial del prototipo.

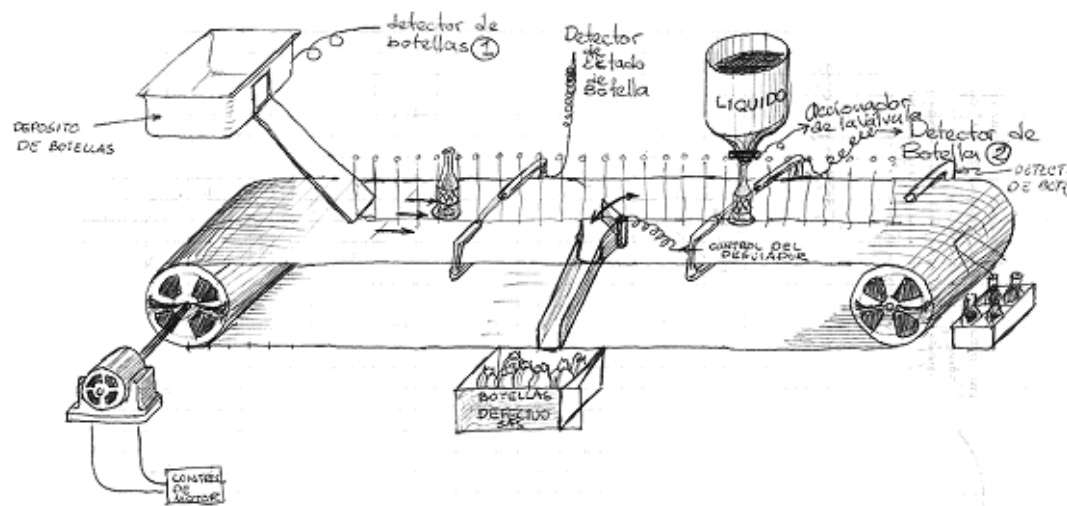


Figura 3-1 Bosquejo inicial del prototipo

Como podemos observar el proceso consta de:

1. La banda
2. El despachador de botellas
3. El desviador de botellas
4. Llenado de botellas

Posteriormente se diseñó el retirador de las botellas para completar el proceso.

Como se puede observar desde el inicio se consideró el uso de detectores de botella ópticos

## Partes involucradas en su funcionamiento

### Dispositivos mecánicos

Una vez definido el sistema y sus secciones, se diseñó el funcionamiento de cada sección, desde un principio se decidió que los movimientos fueran realizados por motores de pasos, dado que la velocidad y el desplazamiento de los mismos pueden ser controlados de manera electrónica, además de poseer un buen par de trabajo.

Otra parte importante fue el desarrollo de los mecanismos de cada sección, en si cada parte del proceso realiza un movimiento específico, el cual involucra además de los motores a las cremalleras, arreglo de engranes, bandas y otros componentes; todos estos conjuntos mecánicos se fueron desarrollando de acuerdo a las necesidades de cada sección.

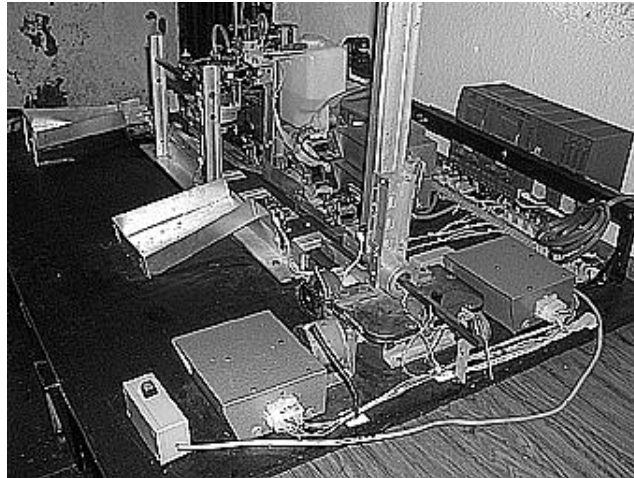


Figura 3-2 Vista del proceso.

### Módulos de control

A medida que se desarrollaba el prototipo, se vio la necesidad de que el control para los motores de pasos fuese realizado de cierta manera. El PLC tiene la virtud de gobernar todo un sistema, sin embargo el control de un motor de pasos resulta muy complejo, de hecho si nuestro sistema tuviera únicamente un motor, el PLC podría proporcionar la secuencia de control para generar los giros y aun así el programa resultante es muy complejo.

Dado que nuestro sistema consta de 6 motores de pasos, el total de las fases a controlar es de 26, si a cada fase se le asignara una salida del PLC, se requerirá de 3 módulos de salidas digitales (considerando módulos de 8 salidas digitales) además de 2 de otro módulo y 2 salidas extras para la bomba y la electro válvula. El programa del PLC para gobernar la secuencia de los 6 motores sería muy grande y bastante complejo.

Para solucionar este problema se optó por desarrollar módulos de control, los cuales van dedicados a cada uno de los motores; en si los módulos proporcionan la secuencia de pulsos a las fases de cada motor para que éste gire, además son capaces de invertir el sentido del giro, con lo cual el motor puede realizar movimientos automáticos, dependiendo de la posición del mecanismo en cualquier etapa del proceso que se trate.

El accionamiento de cada módulo de control se lleva a cabo mediante una salida digital del PLC, esto reduce el número de salidas de 26 a únicamente 6 y el programa del PLC resulta más simple, ya que solo debe mandar un pulso de arranque al módulo de control, en lugar de una secuencia de pulsos en 4 ó 5 salidas digitales dependiendo del motor.

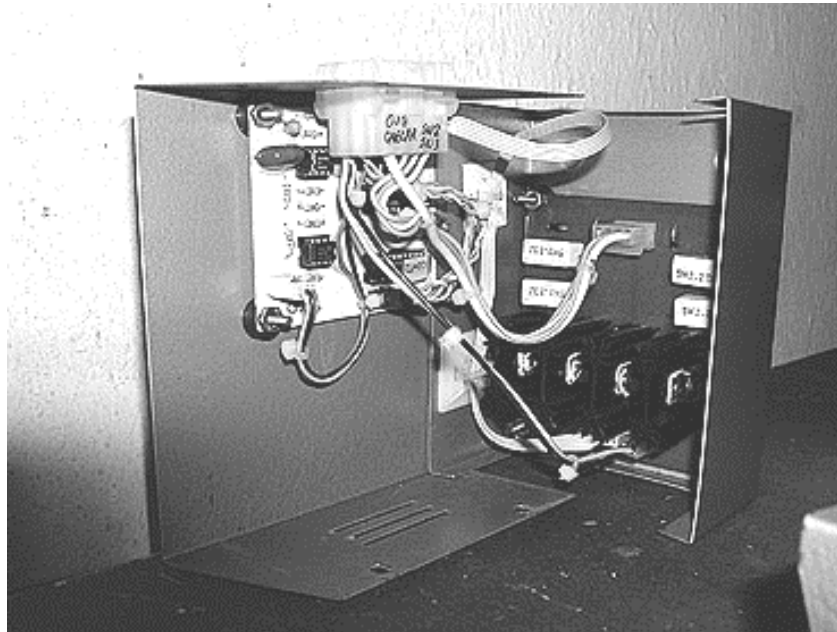


Figura 3-3 Módulo de control

Tanto la secuencia de los pulsos de los motores como el control automático, se desarrolla en base a la lógica digital. Con el fin de minimizar los componentes que conformarían a la tarjeta, se optó por el uso de las GAL<sup>1</sup>, las cuales se pueden programar con la lógica digital necesaria para el control de los motores, de esta forma una GAL puede sustituir a 10 ó más circuitos integrados TTL, lo cual nos ofrece un menor costo y primordialmente disminuye el tamaño de las tarjetas.

De tal forma que se puede afirmar que el control del proceso se lleva a cabo mediante el PLC y los módulos de control.

Cabe mencionar que un módulo de control consta de dos tarjetas una que nos proporciona la secuencia de pulsos y otra que nos ofrece la potencia necesaria para hacer girar al motor. Contamos con dos tipos de motores (motor de 4 y 5 fases) y tarjetas de potencia diseñadas para cada caso.

### Dispositivos de monitoreo

Para que se lleve a cabo un proceso de control automatizado es necesario monitorear lo que acontece en todo momento para cada sección; esta tarea es pensada para que sea realizada por sensores, de tipo óptico y por interruptores. Por lo versátil, práctico y económico para nuestro propósito se ha pensado en utilizar elementos infrarrojo encargados de monitorear la posición y determinar en que sección del proceso se encuentra la botella, además de determinar si se encuentra en las condiciones correctas para poder continuar en el proceso. También se emplearan interruptores, que son los encargados de control la posición de los mecanismos; esto es, nos determinan el desplazamiento de ida y vuelta de cada motor.

Para los sensores del tipo infrarrojo es necesario el diseño de una tarjeta que acople la señal de monitoreo (salida digital TTL) con el sistema de control del PLC. Los interruptores se conectan a las tarjetas de los módulos de control de los motores, mencionado anteriormente.

---

<sup>1</sup> Las GAL (Arreglo Genérico Lógico) son dispositivos lógicos programables, que pueden realizar funciones lógicas mediante un arreglo matricial de compuertas.

### Depósitos de entrada y salida

Hubo diversas formas de pensar en la forma de cómo introducir las botellas al proceso. Se pensó en rampas, espirales, charolas, etc. Por cuestiones de espacio y por ser la forma más sencilla se determinó el uso de entrada en escalera de forma vertical; se dependería únicamente de la fuerza de gravedad para cumplir la función deseada sin la utilización de algún mecanismo complejo que pudiera ocasionar errores considerables.

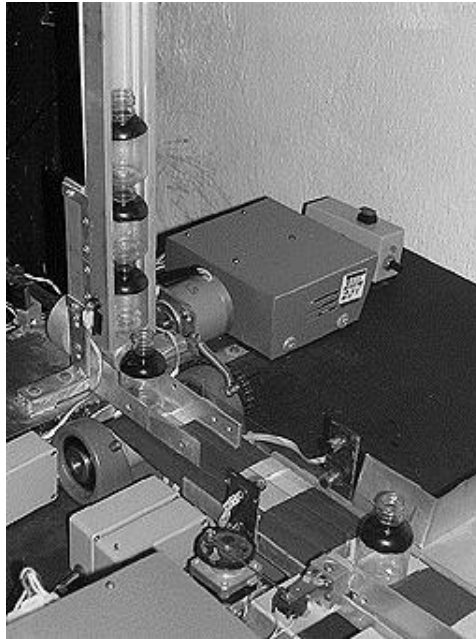


Figura 3-4 Sección de despacho de botellas.

El depósito de salida es la sección final, que a decir verdad fue la sección que se pensó en última instancia, resultó ser relativamente una de las más sencillas; ya que sólo se utilizó un motor y un pistón, los cuales al actuar desviarían la botella de la trayectoria de la banda, hacia un depósito adecuado donde se recibiera un número determinado de ellas calculando que coincidieran tanto en la entrada como en la salida.

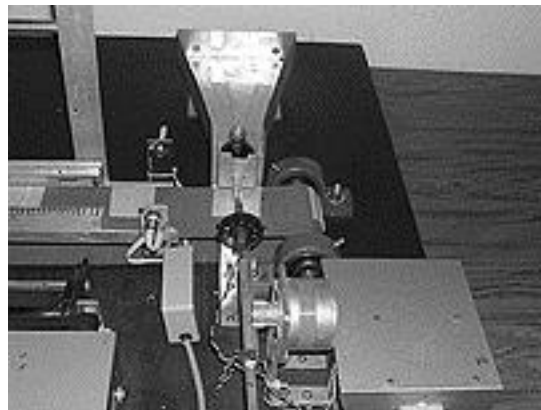


Figura 3-5 Sección de retiro de botellas.



### Depósito de desvío

Cuando alguna botella se encuentre en mal estado, ya sea en su estructura ó en la posición normal del proceso de llenado, el sistema deberá retirarla de la banda desviándola a un depósito tipo rampa, que se encuentra entre el proceso de despacho de botellas y el proceso de llenado.

### Proceso de llenado

El proceso de llenado, consta de un deposito de liquido, un detector de posición y un mecanismo que permite el correcto llenado de las botellas cuando llegan a dicha sección del proceso, con la facilidad de que se repita la misma acción hasta completar el llenado de todas las que se presenten.

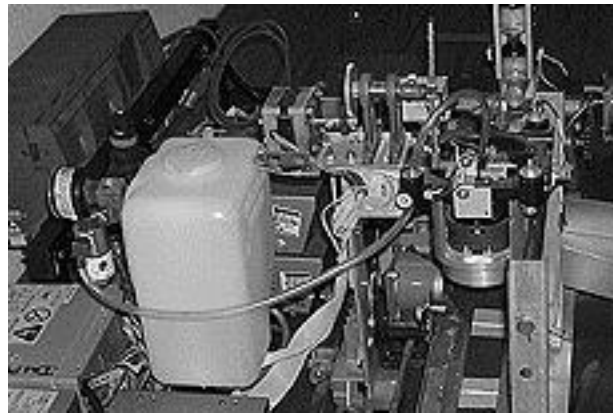


Figura 3-6 Sección de llenado de botellas.

### Descripción del proceso.

La función principal de la planta es el proceso de llenado de botellas, el proceso consta de cinco secciones las cuales se enumeran a continuación:

1. La banda transportadora
2. El despachador
3. El desviador
4. El llenado de botellas
5. El retirador

El diagrama ilustrativo de la interacción de cada una de las secciones será el siguiente:

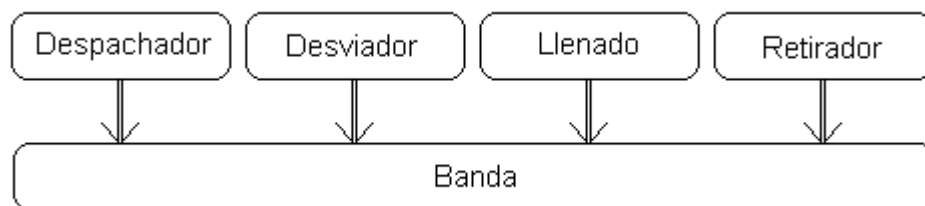
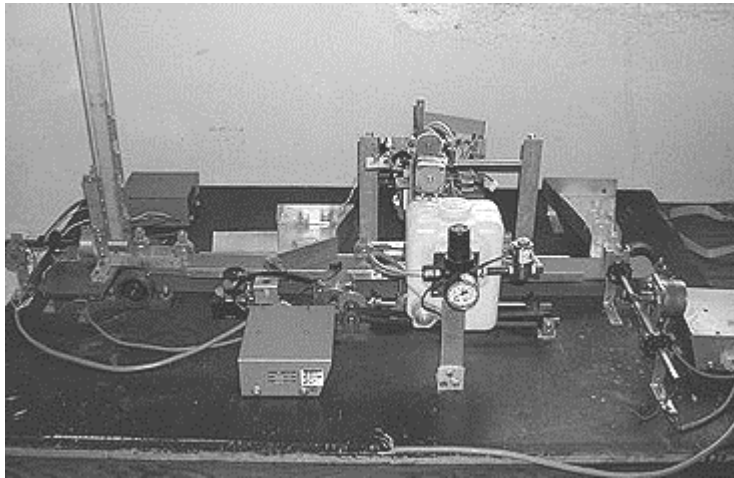


Figura 3-7 Diagrama del proceso.

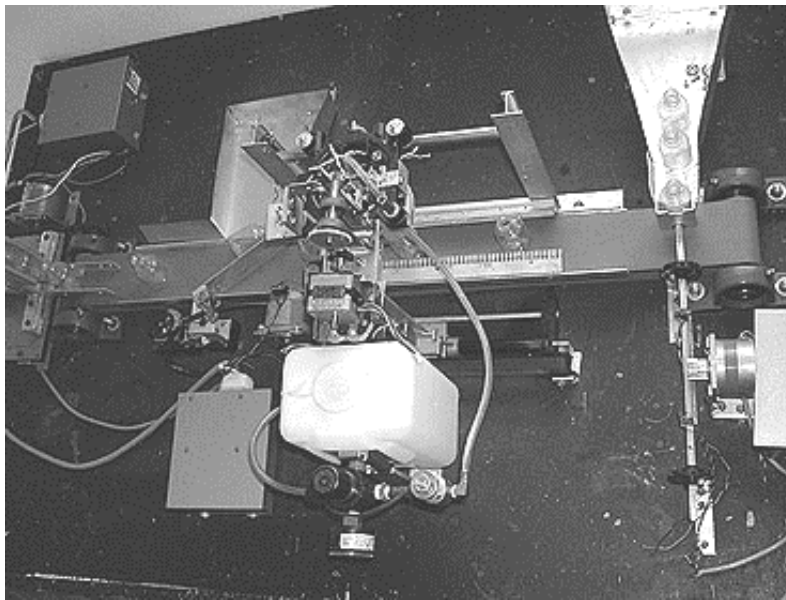


**Figura 3-8** Aspecto real del proceso.

A excepción del llenado de botellas, cada sección consta de un motor de pasos, un mecanismo y un módulo de control.

En la sección para el llenado de botellas se cuenta los siguientes elementos:

- Motor para el desplazamiento horizontal
- Motor para el desplazamiento vertical
- Depósito del líquido
- Bomba
- Electro válvula



**Figura 3-9** Vista superior del proceso.

En esta ilustración se observa mejor la trayectoria de las botellas sobre la banda, el recorrido se realiza de izquierda a derecha, la primera sección es la del despachador, posteriormente se encuentra el desviador, el llenado de botellas se observa como un complejo mecánico donde resalta el depósito del líquido y finalmente a la derecha se observa la sección del retirado de botellas; a continuación se dará una breve explicación de cada una de las secciones.

## La banda transportadora

La banda en sí es el centro de operación de todo el sistema, la banda cuenta con un ancho de 3 pulgadas y la distancia que recorren las botellas es de 83 centímetros, la banda gira sobre una pieza de aluminio que le sirve de soporte y guía en la mayor parte del proceso, las partes móviles son dos chumaceras una de las cuales es impulsada por un motor de pasos mediante una transmisión de cadena, este motor realiza un movimiento unidireccional, y su única entrada de control será un bit de paro, con éste la banda se detendrá bajo alguna condición de control, que en nuestro caso será cuando se terminen las botellas en el proceso.

El diagrama de esta sección será el siguiente:

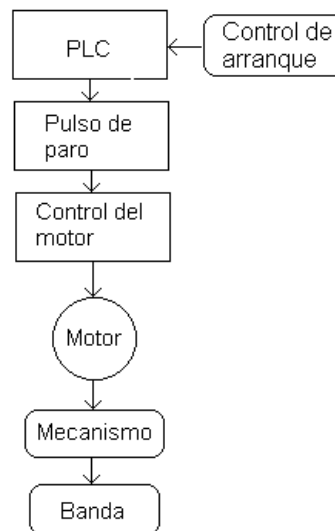


Figura 3-10 Diagrama para el control de los desplazamientos unidireccionales.



Figura 3-11 Motor para el desplazamiento de la banda.

En el caso de los siguientes motores de pasos que completan la planta, estos realizan movimientos automáticos, cada motor lleva un módulo de control distinto al igual que su mecanismo, sin embargo todos tienen características similares:

- Poseen interruptores de límite
- Tienen un punto de inicio
- Controlan el giro del motor

Por lo cual podemos afirmar que un diagrama representativo para estos motores será el siguiente:

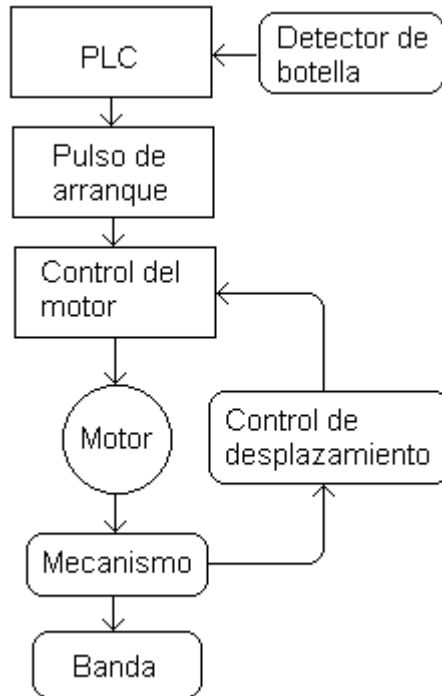


Figura 3-12 Diagrama para un control bidireccional.

Lo único que varía en cada motor es el control del motor y el mecanismo, todo esto dependiendo del proceso en cuestión.

## Despachador de botellas

En esta sección se alimentan las botellas hacia la banda transportadora; sobre un riel de aluminio se colocan las botellas y con un embolo se desplazan las botellas, cuando el embolo se desplaza hacia la banda realiza dos acciones: la primera es empujar las botellas sobre la banda en movimiento y la segunda es bloquear la botella siguiente. Una vez que el embolo regresa a su posición inicial, la siguiente botella se desbloquea y por acción de la gravedad se pone en el punto de despacho. Este embolo se impulsa por un motor que empuja una cremallera, su control corre a cargo de un módulo con controles tipo i y ii<sup>2</sup>.

<sup>2</sup> Una descripción de este tipo de controles se expone en la sección “Control de motores de desplazamiento”.

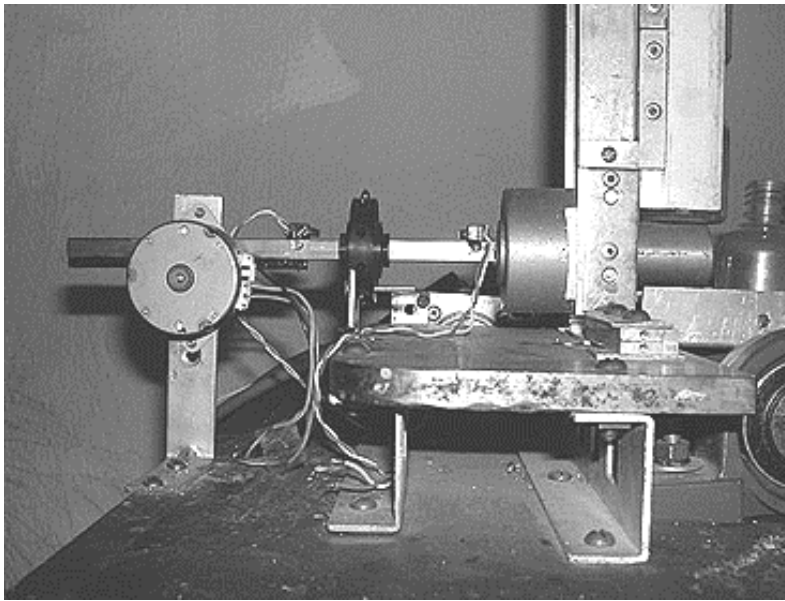


Figura 3-13 Mecanismo para el despachador de las botellas.

### Desviador de botellas

Esta sección tiene la función de detectar cuando una botella se ha caído y debe ser retirada del curso de la banda, en si la función es muy sencilla, dado que el motor realiza un giro de  $360^\circ$  para este fin, como se puede ver en la figura la pala entra dentro de la trayectoria de la banda y desvía la botella hacia la primera charola, el motor emplea un módulo con controladores tipo i y iii.

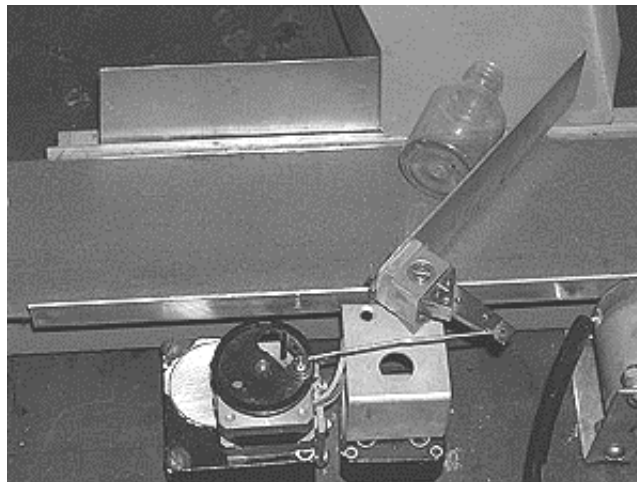


Figura 3-14 Mecanismo del desviador de botellas.

### El llenado

El llenado es una sección que consta de 2 motores y una bomba; en esta sección se llena la botella mediante dos movimientos del conjunto mecánico: un desplazamiento horizontal que seguirá a la botella mientras se llena y otro movimiento horizontal que acercará una boquilla a la botella en cuestión. Cada uno de estos movimientos se realiza por un motor. El primer movimiento

se lleva a cabo por un motor instalado en la base el cual posee un arreglo mecánico de engranes, con el fin de aumentar el par requerido para mover la sección de llenado, el traslado del mismo se realiza mediante una banda, como se muestra en la figura. Este motor es controlado por un módulo con controladores tipo i y iv.

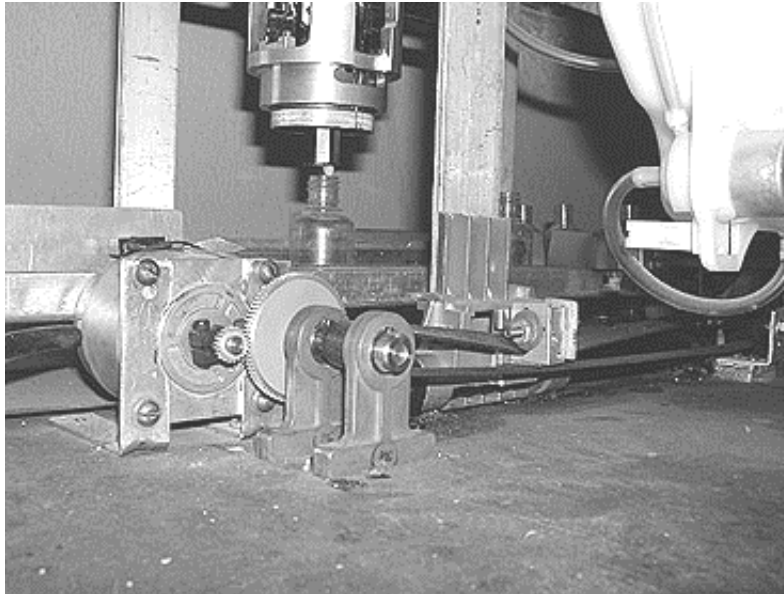


Figura 3-15 Mecanismo para el desplazamiento de la sección de llenado.

El segundo movimiento lo realiza otro conjunto mecánico situado sobre la sección de llenado, en este caso el motor se acopla a un conjunto de engranes que a su vez impulsan a una cremallera la cual moverá la sección de la boquilla hacia arriba o abajo. Este motor se controla por un módulo con controladores tipo i y v.

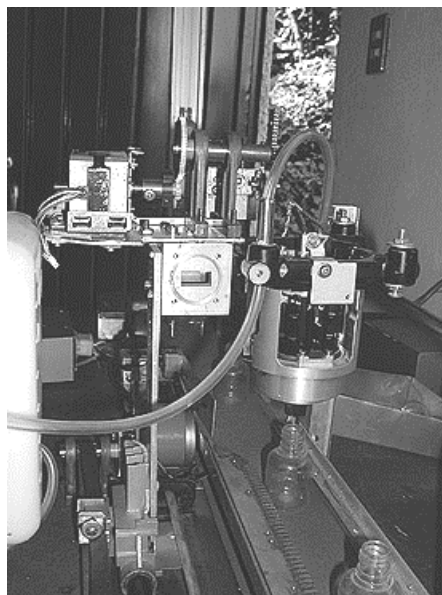


Figura 3-16 Mecanismo para el desplazamiento de la boquilla del llenado.

Finalmente esta sección es complementada por un depósito de líquido, que a su vez incluye una bomba que se acciona con 12 VCD, y que además dentro del recorrido del líquido se ha agregado una electro válvula de 12 VCD, con lo cual se puede manejar eficientemente el líquido y el ducto llega hasta la boquilla del llenador.

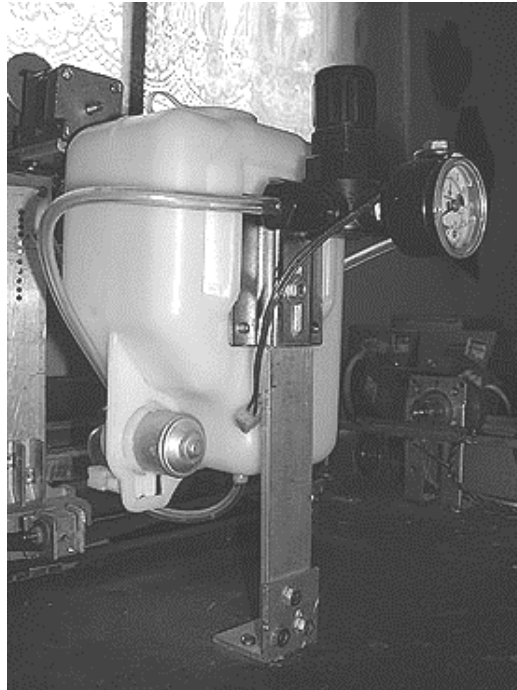
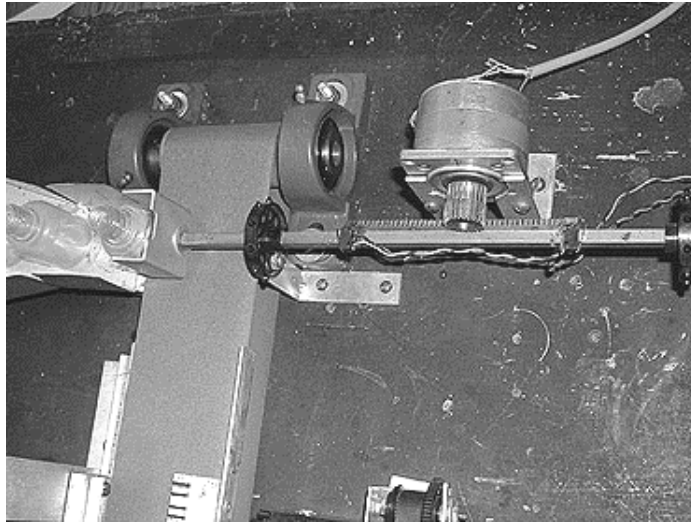


Figura 3-17 Depósito del líquido en la sección de llenado

### **El retirador**

Finalmente cuando se ha llenado la botella y esta llega al final del recorrido, se activará el retirador, cuya función primordial es mover las botellas hacia la charola 2. Al igual que el despachador, su movimiento es de 2 límites de desplazamiento, también se realiza mediante el uso de una cremallera y un engrane. En este caso el módulo del motor contiene controladores tipo i y ii.

**Figura 3-18 Mecanismo de la sección del retirador de botellas.**





### Módulo de Control

El módulo de control esta integrado por dos etapas: una etapa de control de giro del motor y una etapa de potencia. La función principal del módulo de control es determinar el giro del motor bajo ciertas condiciones y estas son: el bit de arranque o de paro y los interruptores de límite. Se cuentan con dos tipos de tarjetas de control de giro de motor y dos tarjetas de potencia, las cuales se podrán tener algunas combinaciones de ellas entre sí, dependiendo de la sección del proceso y del tipo de motor utilizado.

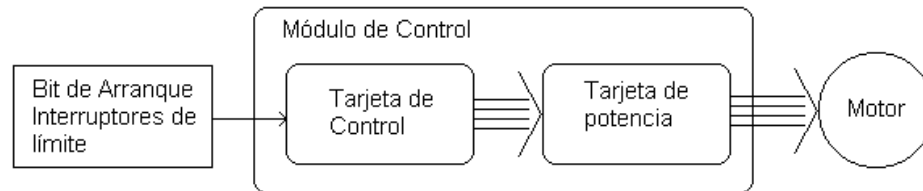


Figura 3-19 Interacción de un módulo de control.

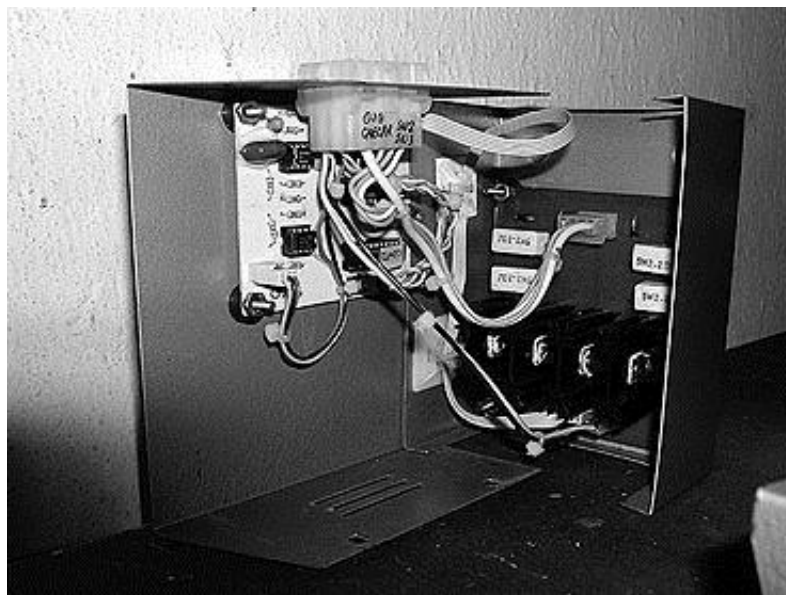


Figura 3-20 Aspecto físico del módulo de control para los motores de pasos.

## Tarjetas de control

Las tarjetas de control de giro del motor son de dos tipos:

1. **Tarjeta de control dedicada a la banda.**-Esta tarjeta controla únicamente el desplazamiento de la banda, dado que su movimiento es unidireccional no se necesita un control de desplazamiento, por lo cual se emplea una GAL16V8 para el control de giro del motor (tipo i), su control de giro se define con la utilización de un jumper, la única entrada que se emplea es el bit de paro; además cuenta con una sección de reloj, esta manejada por un Oscilador Controlado por Voltaje (VCO), este nos permite tener un ajuste fino en la velocidad de giro del motor de la banda.

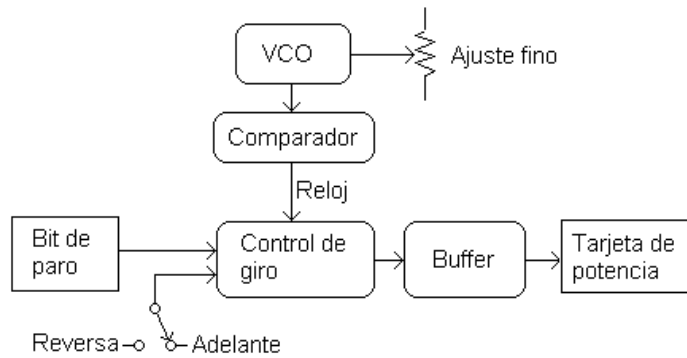


Figura 3-21 Diagrama de la tarjeta unidireccional (banda).

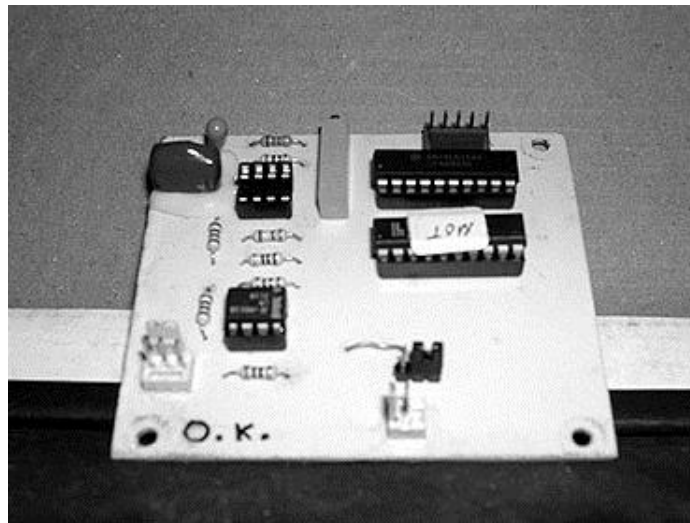
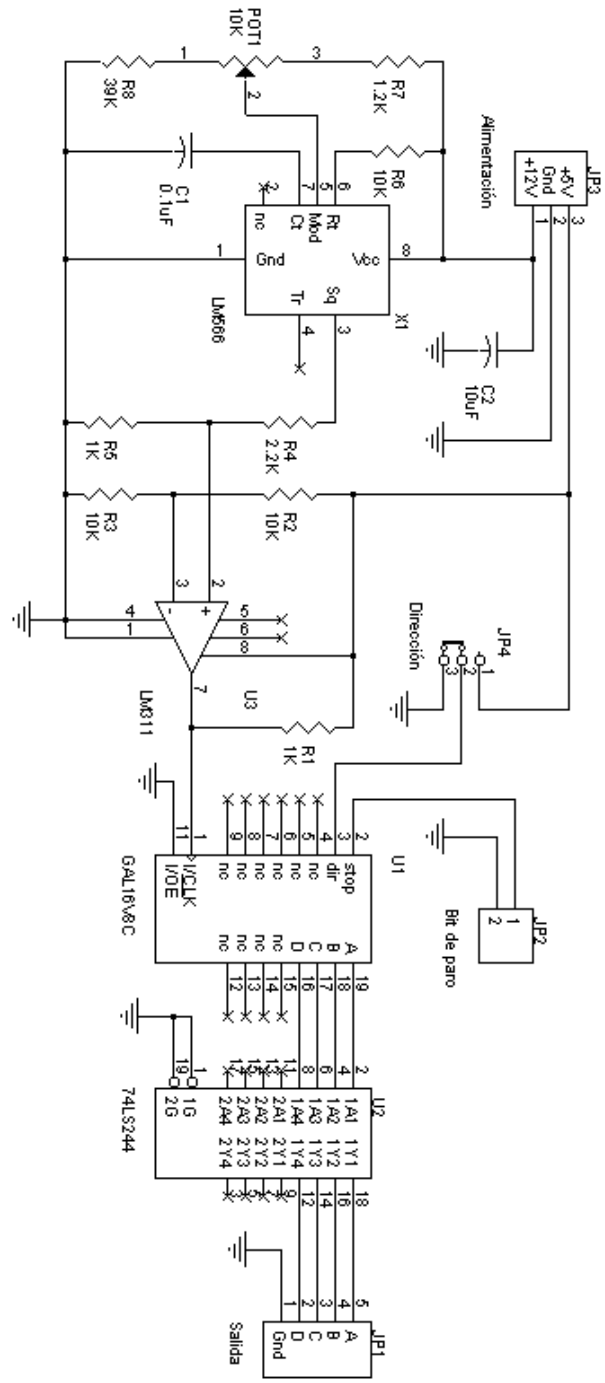


Figura 3-22 Tarjeta unidireccional.



Título		<b>Tarjeta de control 1</b>	
Size	Document Number		Rev
A			1
Date:	Saturday, August 10, 2002	Sheet	1 of 1

Figura 3-23 Circuito electrónico de la tarjeta unidireccional.

2. **Tarjeta de control automático.**- Esta tarjeta es la responsable de controlar los motores que se utilizaran en las secciones de desplazamiento automático, que conforman la planta. Está consta de dos GAL16V8; la primera determina el control de desplazamiento, la cual tomará decisiones a partir de los interruptores de límite (uno ó dos dependiendo de la sección) y del bit de arranque, se tendrá diferente tipo de control (ii, iii, iv ó v) dependiendo de la sección del proceso. La segunda determinara el control de giro del motor, similar a la que se emplea en la tarjeta de control de la banda, en este caso el control de la dirección de giro y el paro serán determinados por la GAL del control de desplazamiento; ésta tarjeta también cuenta con un VCO, que nos permite tener un control más fino del motor.

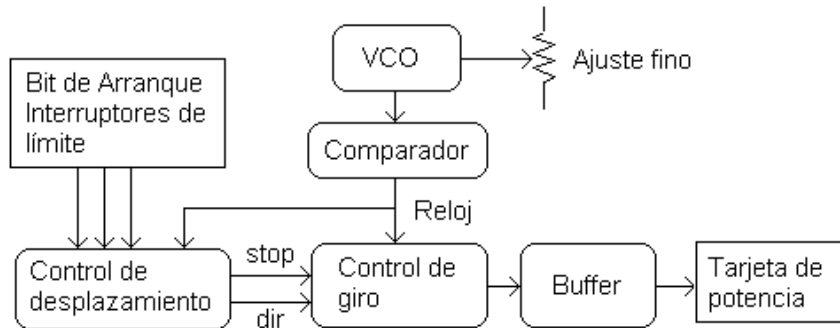


Figura 3-24 Diagrama de la tarjeta bidireccional.

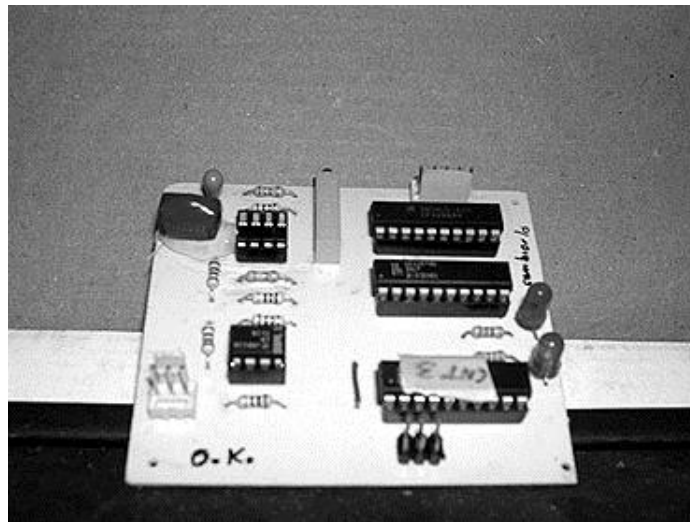


Figura 3-25 Tarjeta bidireccional.

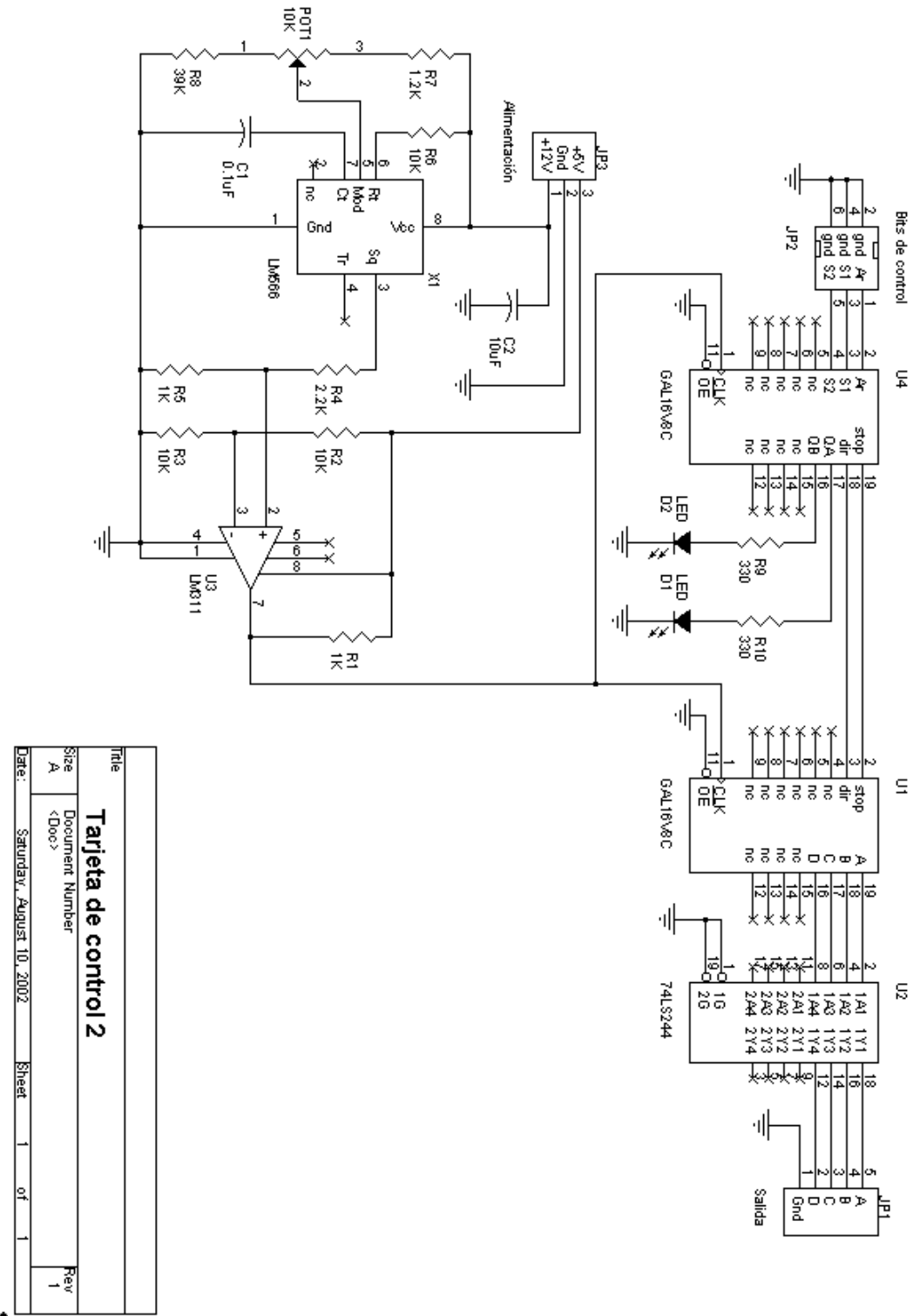


Figura 3-26 Circuito electrónico de la tarjeta bidireccional.

Title		<b>Tarjeta de control 2</b>	
Size	Document Number		
A	<Doc>		
Date:	Saturday, August 10, 2002	Sheet	1 of 1
			Rev
			1

## El VCO

Una parte importante en las tarjetas de control es la de contar con un reloj en el que puedas tener una buena regulación, mediante uso de un VCO; a diferencia de los temporizadores, los VCO pueden producir un rango de frecuencias a partir de un voltaje de modulación, un capacitor y una resistencia externos. Se determino emplear el CI LM566 de National Semiconductors. Para nuestros fines solamente se empleará la salida de pulsos cuadrados (terminal 3) y tendremos el siguiente arreglo para su operación:

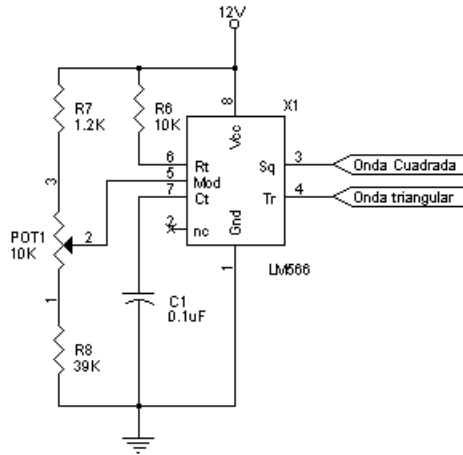


Figura 3-27 Circuito VCO.

Para determinar la operación del VCO, nuestro primer paso es encontrar cuales son los límites del voltaje de modulación (terminal 5), para esto se analizan 2 posiciones: cuando el potenciómetro POT1 está en su máximo y para cuando está en su mínimo:

$$V_{\text{mod}_{max}} = \frac{R8 + POT1}{R8 + POT1 + R7} \times V_{cc} = \frac{39K + 10K}{39K + 10K + 1.2K} \times 12V = 11.7V$$

$$V_{\text{mod}_{min}} = \frac{R8}{R8 + POT1 + R7} \times V_{cc} = \frac{39K}{50.2K} \times 12V = 9.3V$$

Con estos datos podremos ahora calcular sus frecuencias límites de operación empleando la siguiente fórmula:

$$f_o = \frac{2}{R_t C_t} \left[ \frac{V_{cc} - V_{MOD}}{V_{cc}} \right]$$

Sabemos que  $R_t = 10\text{ K}\Omega$ ,  $C_t = 0.1\ \mu\text{F}$ ,  $V_{cc} = 12\text{ V}$ , de tal forma que sustituimos únicamente  $V_{MOD}$  en la fórmula y obtendremos los siguientes resultados:

$$f_{min} = \frac{2}{(1 \times 10^4 \Omega)(1 \times 10^{-7} F)} \left[ \frac{12V - 11.7V}{12V} \right] = 50Hz$$

$$f_{max} = \frac{2}{(1 \times 10^4 \Omega)(1 \times 10^{-7} F)} \left[ \frac{12V - 9.3V}{12V} \right] = 450Hz$$

Se observa que tenemos un rango de operación entre 50 y 450 Hz, cabe señalar que otra característica del VCO es que siempre maneja un ciclo de trabajo del 50%. Sin embargo el problema existente en las ondas de salida es que tienen un offset agregado, para eliminarlo y además acondicionar la señal a los niveles de TTL, es necesario emplear un comparador el cual nos entregará a la salida una señal de reloj compatible con TTL (esta señal manejará el reloj de las GAL); el comparador utilizado para nuestro caso es el LM311 y el circuito será como el que se muestra a continuación:

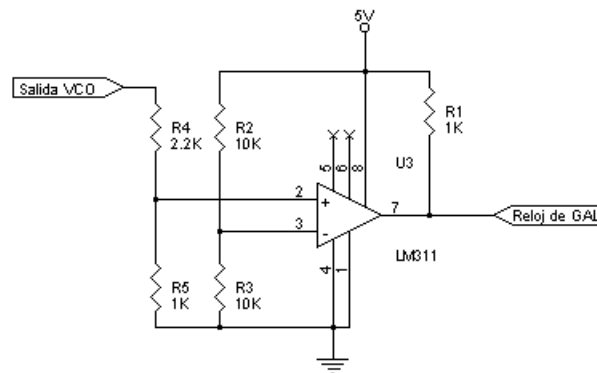


Figura 3-28 Acoplador de nivel empleando un comparador LM311.

## Tarjetas de potencia

Las tarjetas de potencia proveen de la corriente y el voltaje necesarios para hacer funcionar nuestro motor de pasos; dadas las características de nuestros motores contamos con dos tipos de tarjetas de potencia:

1. **Tarjetas de potencia para motores unipolares.**- Esta tarjeta maneja a los motores unipolares como el motor de la banda y el motor de desplazamiento vertical del llenador. Físicamente un motor unipolar posee 5 ó 6 hilos externos, esta tarjeta puede manejar ambas configuraciones como se muestra en las siguientes ilustraciones:

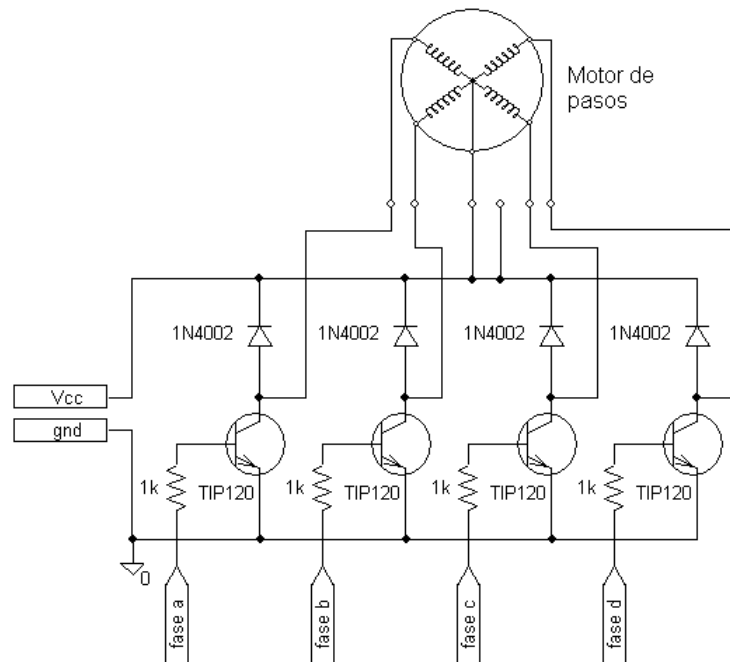


Figura 3-29 Diagrama para un motor unipolar de 5 hilos.

En este caso la tarjeta se conecta a un motor de 5 hilos, como se puede observar la configuración interna del motor es con un común y 4 terminales polares.

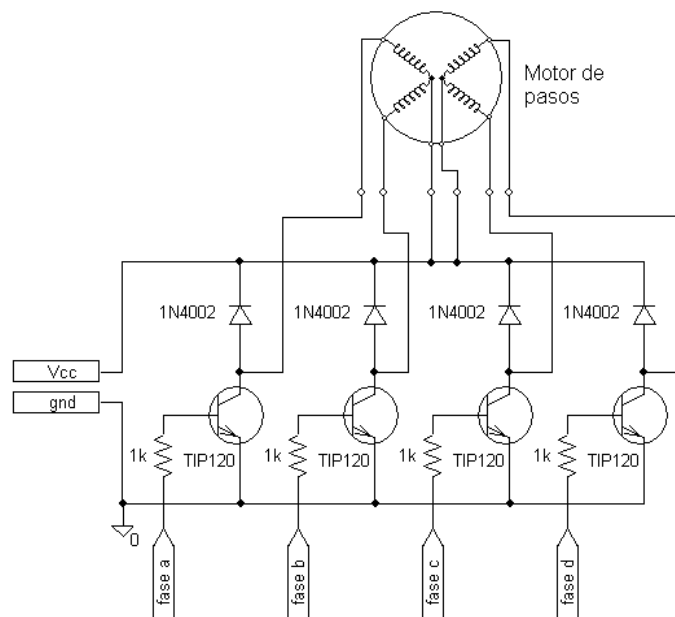


Figura 3-30 Diagrama de conexión para un motor unipolar de 6 hilos.

Esta es la conexión de un motor unipolar de 6 hilos, internamente el motor tiene dos arreglos de embobinados, cada uno con un común y dos terminales polares.



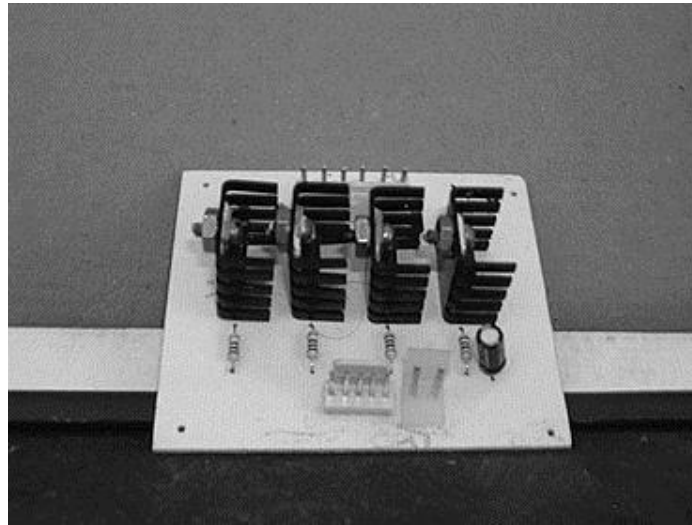


Figura 3-31 Tarjeta de potencia para motores unipolares.

2. **Tarjeta de potencia para motores bipolares<sup>3</sup>**. - Esta tarjeta impulsa a los motores de las secciones restantes del proceso, se emplea una configuración “media H”, que nos permite cambiar la polaridad en cada bobina del motor. El circuito varía con respecto a la tarjeta de potencia para motores unipolares como se muestra en la siguiente ilustración:

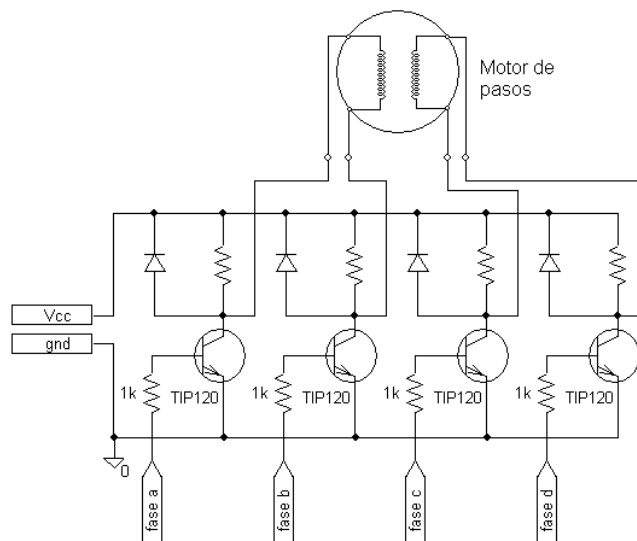


Figura 3-32 Diagrama de conexión para motores bipolares.

<sup>3</sup> Una descripción más detallado a cerca de los motores unipolares y bipolares se muestra en la siguiente sección “Motores de pasos”.

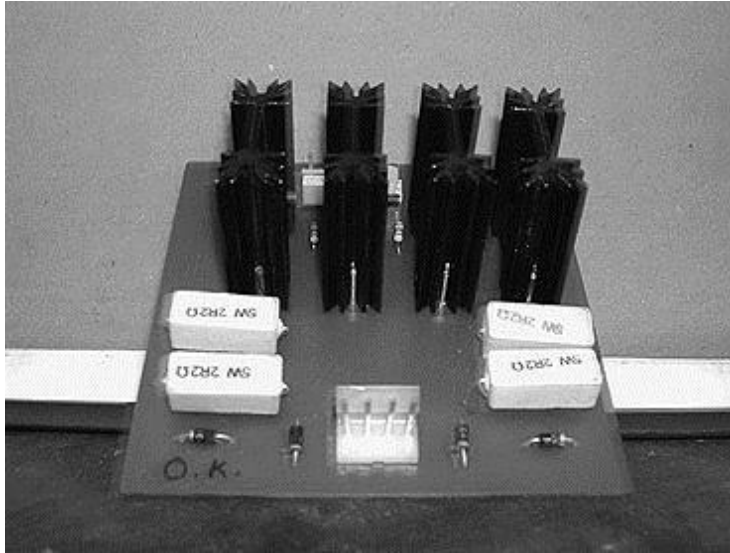


Figura 3-33 Tarjeta de potencia para motores bipolares.

## Motores de pasos

Los motores de pasos difieren de los motores de corriente directa en su forma de producir su giro, mientras que los otros motores simplemente girarán libremente al aplicárseles una diferencia de tensión en las terminales de la bobina del estator, en los motores de pasos se le tendrá que aplicar una secuencia de pulsos a las terminales de las bobinas del estator para generar un movimiento con cierto ángulo de desplazamiento, a esto se le llama "paso", y tiene valores que van desde  $0.9^\circ$  hasta  $90^\circ$  de desplazamiento angular, con estas características podemos controlar el giro en este motor de una manera más precisa; además es posible cambiar el sentido del giro del motor, simplemente cambiando la secuencia de los pulsos en las bobinas.

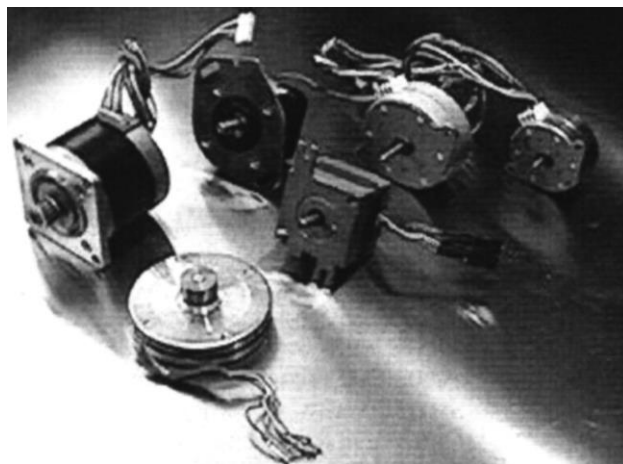


Figura 3-34 Motores de pasos.

La velocidad del giro va relacionada con la frecuencia de los pulsos aplicados, debemos tomar en cuenta que este motor girará a partir de incrementos en el desplazamiento angular, y por lo mismo se necesitarán  $n$  pasos para completar  $360^\circ$ , sin embargo la frecuencia va relacionada con dos características del motor:

- A bajas frecuencias el motor ofrece un par alto, que irá disminuyendo a medida que se incremente la frecuencia de los pulsos.
- Existe una frecuencia límite de operación para estos motores y que varía en cada modelo, arriba de esta el motor perderá la sincronía y comenzará a vibrar sin realizar ningún desplazamiento angular.

La necesidad de poder maniobrar un movimiento con exactitud nos ha llevado a seleccionar a los motores de pasos sobre otro tipo de motores, ya que ofrecen algunas ventajas como:

1. Control preciso de posición y velocidad.
2. Alto par a bajas velocidades, sin el empleo de un juego de engranes.
3. Operación bidireccional, que se genera cambiando la secuencia de pulsos aplicados a las bobinas del motor.
4. Tienen un error de posición no acumulativo ( $\pm 5\%$  del ángulo de paso).
5. Par inherente debido a su construcción.
6. Pueden ser forzados a parar sin que el motor sufra daños.
7. No tienen escobillas, lo cual nos da una vida operativa más larga.
8. Pueden manejarse en un sistema de lazo abierto.
9. La velocidad de rotación es proporcional a la frecuencia de los pulsos de entrada.
10. Una respuesta excelente a comandos de aceleración, desaceleración y avance por pasos.

Aunque claro también ofrece algunas desventajas que se pueden compensar con sus prestaciones:

1. Mayor números de bobinas en el estator.
2. Control más complejo.
3. Tamaño mayor y por lo mismo mayor peso.
4. La frecuencia de operación es menor que los motores convencionales de corriente directa.
5. Tiene una frecuencia límite de operación arriba de ésta el motor comienza a vibrar y pierde par.
6. Costo.

En lo que se refiere a su funcionamiento, un motor de pasos se ha basado en el principio de los motores síncronos: un campo magnético rotativo, controlado por circuitos electrónicos que ponen en funcionamiento al rotor, que es un imán permanente.

Como en todos los motores, las partes principales son el estator y el rotor. El estator puede contener varias fases con sus respectivas bobinas de excitación, en todo momento el campo magnético producido por una fase dependerá de la intensidad de la corriente que circula por el embobinado de dicha fase. Cuando esta corriente sea grande producirá un campo magnético intenso.

El rotor por su parte, al ser un imán se alineará al campo producido por el estator; cuando el rotor esté bajo la influencia de un campo magnético intenso, se producirá un par, con el cual se puede mover una carga.

### Construcción

La construcción de un motor de pasos típico se muestra a continuación:

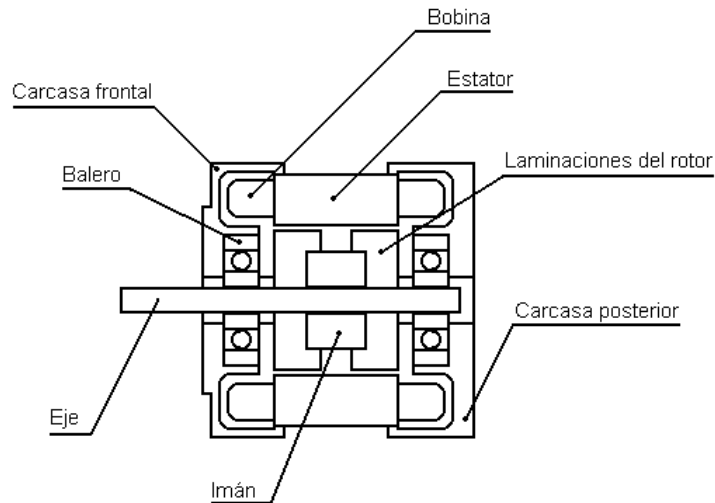


Figura 3-35 Construcción de un motor de pasos.

Para producir un paso más pequeño, el rotor contiene algunos arreglos en cuanto a sus polos magnéticos como se muestra a continuación:

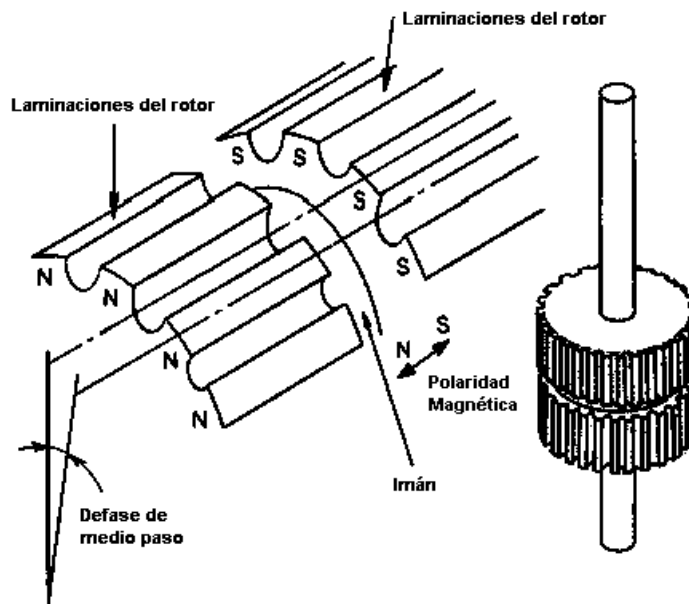


Figura 3-36 Detalle de la disposición de los polos del rotor de un motor de pasos.

### Funcionamiento de un motor de pasos

Para explicar el funcionamiento de estos motores, tomaremos como ejemplo un motor de pasos de  $1.8^\circ$  y de 4 fases. Su estator tiene 8 polos distribuidos cada  $45^\circ$ , en cada polo tenemos 5 piezas dentadas en intervalos de  $7.2^\circ$ , de tal manera que el embobinado del estator estará distribuido como se muestra a continuación:

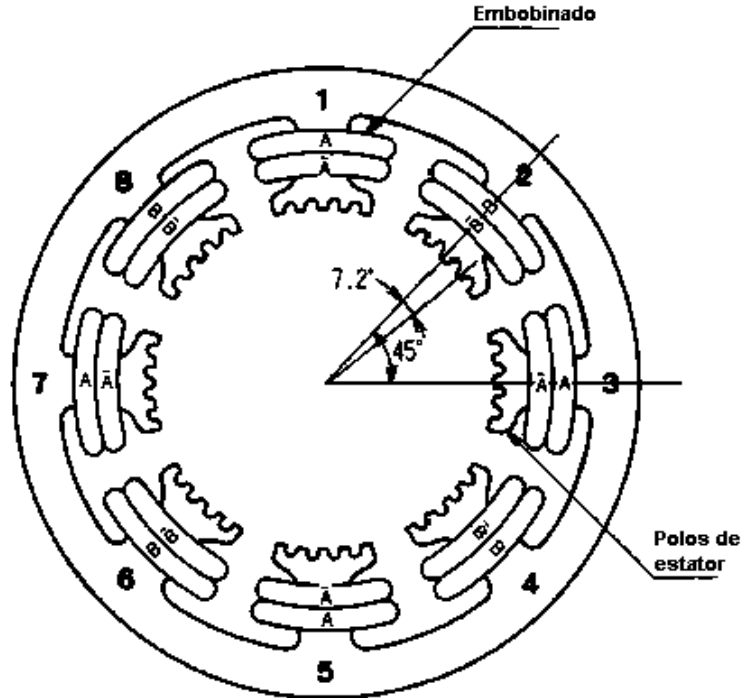


Figura 3-37 Disposición de los embobinados en el estator de un motor de pasos.

Para producir un campo magnético rotatorio se debe aplicar corriente a cada embobinado con la siguiente secuencia:

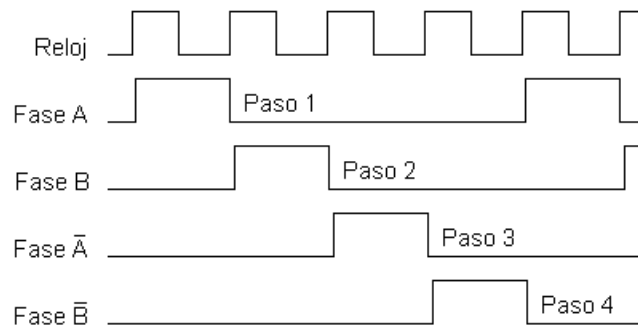


Figura 3-38 Secuencia de la excitación de las bobinas de un motor de pasos.

El movimiento alternativo del campo magnético rotatorio generado en los polos del estator se puede observar en las siguientes figuras:

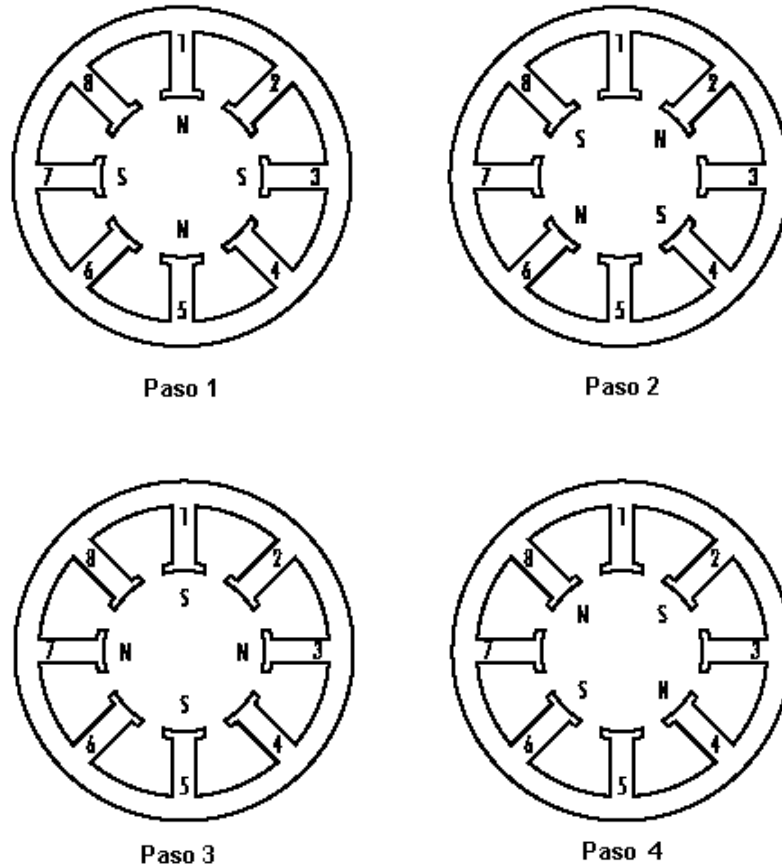


Figura 3-39 Movimiento del campo magnético en los polos del estator.

El rotor tiene 2 juegos de laminaciones separadas por un imán permanente, cada conjunto de laminaciones contiene 50 piezas dentadas, las cuales están desfasadas medio paso, por lo cual el rotor tiene 50 polos norte y 50 polos sur.

En el paso 1, la fase A será alimentada con lo cual el polo S del rotor será atraído hacia los polos 1 y 5 del estator que tendrán polaridad N, de la misma forma los polos N del rotor serán atraídos a los polos 3 y 7 del estator que tienen polaridad S. En este punto hay una diferencia de ángulo entre las piezas dentadas del rotor y del estator de  $\frac{1}{4}$  de paso ( $1.8^\circ$ ). Por esta razón las piezas dentadas del estator de los polos 2, 6, 4 y 8, tienen un desfase de  $1.8^\circ$ , con respecto a las del rotor.

En el paso 2, la posición se estabilizará cuando los polos S del rotor se alineen con los polos 2 y 6 del estator, que ahora tendrán una polaridad N, de igual forma los polos N del rotor se alinearán con los polos 4 y 8 del estator que ahora tienen polaridad S. Con este paso se produce un desplazamiento  $1.8^\circ$ , con respecto a la posición presente en el paso 1.

Por consiguiente los pasos 3 y 4, generarán un paso de  $1.8^\circ$  de rotación.

De manera gráfica, este desplazamiento se puede observar en la siguiente ilustración:

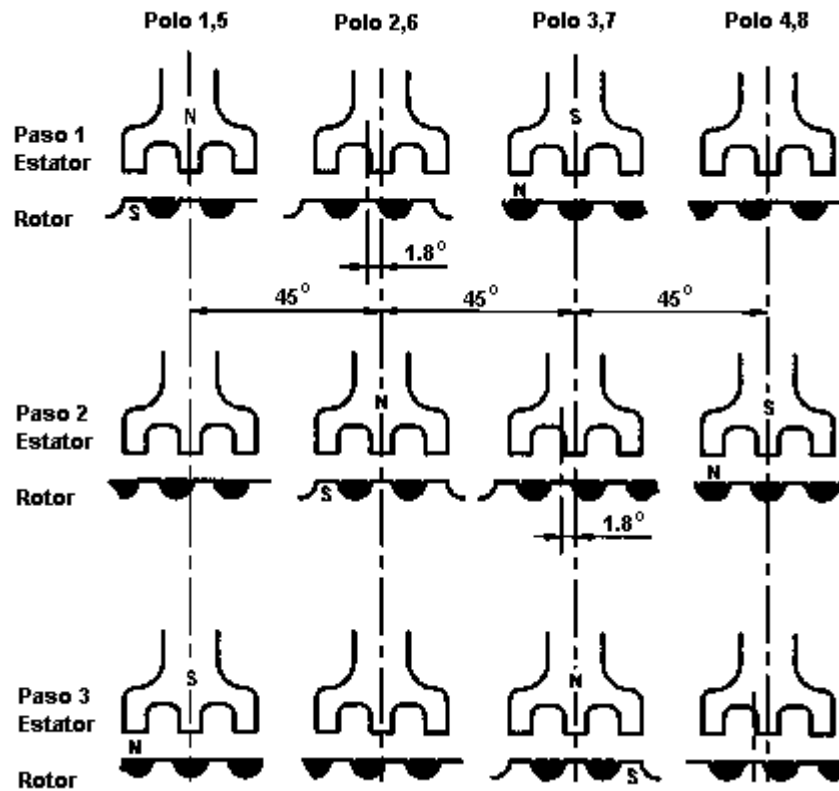


Figura 3-40 Movimiento mecánico resultante.

De lo anterior se puede determinar que para que el motor realice una revolución se necesitan  $n$  pasos.

Para calcular la velocidad de giro, necesitaremos conocer el paso del motor, la frecuencia de los pulsos de excitación de las bobinas. La siguiente fórmula nos sirve para calcular su velocidad de giro:

$$V_{giro} = \frac{(f_{pulsos})(\phi_{paso})}{360^\circ} \left[ \frac{Rev}{seg} \right]$$

Cabe mencionar que la frecuencia de operación tiene un límite superior y que esta variará de un modelo a otro, y que por lo general todos los motores de paso ofrecen un mejor desempeño a bajas frecuencias.

### Disposición de las bobinas

Los motores de pasos, dependen de la distribución de las bobinas dentro del estator del motor, por lo que se clasifican en:

1. Motores unipolares.
2. Motores bipolares.

### Motores unipolares

La distribución de las bobinas de los motores unipolares se muestra en la siguiente figura:

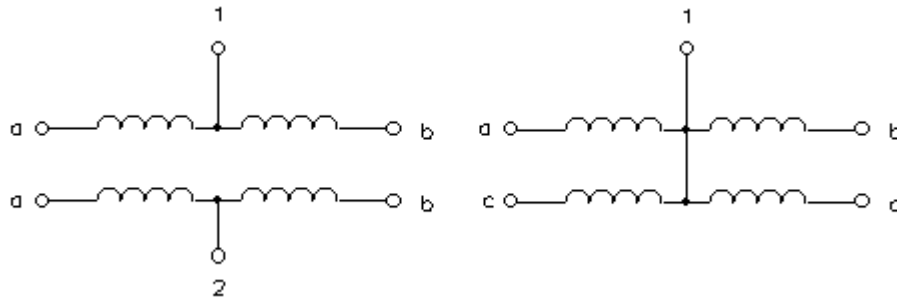


Figura 3-41 Arreglo de las bobinas de un motor unipolar.

Como se puede observar cada bobina del estator se encuentra dividida en dos mediante una derivación central que por lo general se conecta a la alimentación, el sentido de la corriente que circula a través de la bobina se determina mediante las otras terminales de la bobina que quedan libres; así, si una terminal se conecta a tierra la corriente fluirá por esa bobina y será la que determine el campo magnético del estator. El caso opuesto es cuando una terminal se conecta a alto, no fluirá corriente y por lo tanto no producirá campo magnético alguno, de esta forma los polos magnéticos del estator se invierten conmutando las terminales de las medias bobinas de excitación.

Estos motores son fáciles de distinguir, ya que cuentan entre 5 y 6 cables en el conector del mismo.

### Motores bipolares

Estos motores solo cuentan con un par de bobinas de excitación, como se muestra en la siguiente figura:

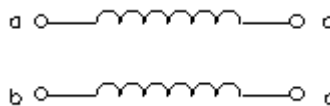


Figura 3-42 Disposición de las bobinas de un motor bipolar.

Al igual que en el caso de los motores unipolares, estas bobinas deberán ser las responsables del campo magnético generado, la corriente fluirá solamente si existe una diferencia de potencial en ambas terminales de la bobina, así por ejemplo, para que circule corriente en la primera bobina,  $V_a$  deberá tener más potencial que  $V_c$ , por lo que la intensidad del campo magnético generado se relacionará con la corriente que circula en la bobina; algo similar ocurrirá en la segunda bobina, sin embargo, para que el rotor gire deberá existir un campo magnético rotatorio, para lograr esto se deberá estar conmutando el sentido de la corriente en la bobina y así, en el caso de la primera bobina la corriente fluirá de  $a$  hacia  $c$ , para cierto sentido en el campo magnético, para invertir el campo magnético, la corriente deberá fluir desde  $c$  hacia  $a$ .

Como se puede observar el control del campo magnético ahora involucra el manejo de las terminales de cada bobina, para producir un cambio en el sentido de la corriente. Este control de la corriente resulta más complejo, y por lo general se emplea un puente H para cada bobina, en



nuestro caso se empleo un medio puente H, como se puede ver en la sección de las tarjetas de potencia.

Una de las pocas ventajas que ofrecen los motores bipolares sobre los motores unipolares, es que tienen mejor par que los motores unipolares.

### Diferentes tipo de excitación de las bobinas.

Como se dijo anteriormente, estos motores necesitarán la aplicación de una secuencia determinada en sus bobinas que provocará un desplazamiento angular del eje llamado paso, el sentido del giro está determinado por la secuencia seguida.

En el caso de los motores unipolares, se le pueden aplicar 3 tipos distintos de excitación, las cuales se detallaran más adelante; cabe señalar que todas comienzan en el paso 1 y al llegar al paso 4 se repite el ciclo. Para revertir el sentido del giro, se deberán ejecutar las secuencias en orden inverso.

### Secuencia normal

Esta es la secuencia más usada y es la que más recomiendan los fabricantes. Con esta secuencia el motor avanza un paso a la vez y debido a que siempre hay dos bobinas activadas, se obtiene un par alto de paso y de retención.

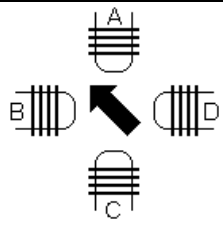
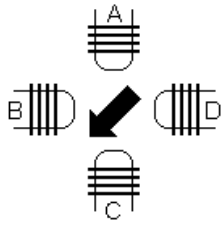
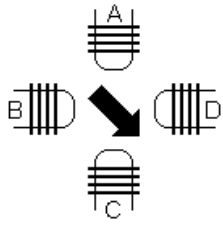
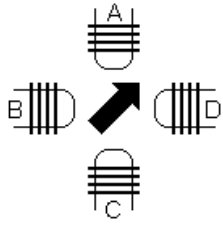
Paso	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	ON	OFF	OFF	
2	OFF	ON	ON	OFF	
3	OFF	OFF	ON	ON	
4	ON	OFF	OFF	ON	

Tabla 3-1 Secuencia normal.

**Secuencia tipo wave drive.**

En esta secuencia también se produce un desplazamiento de un paso, la diferencia es que solo se activará una bobina a la vez. En algunos motores esto produce un desplazamiento más suave, sin embargo al estar activada solo una bobina, el par de paso y retención es menor.

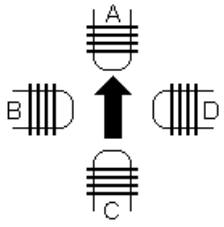
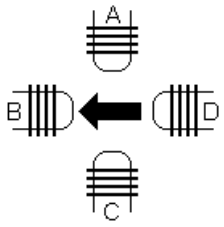
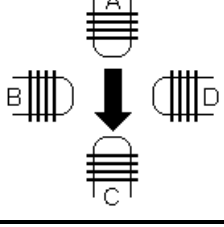
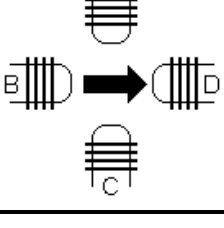
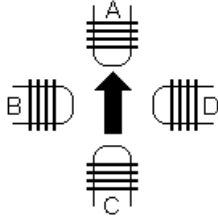
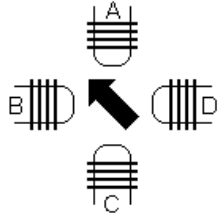
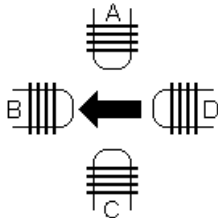
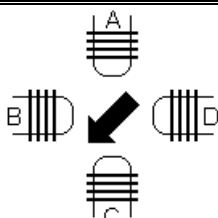
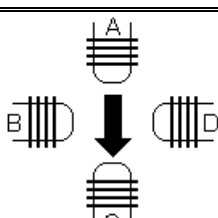
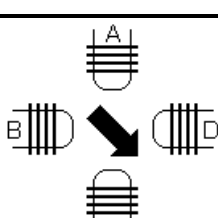
Paso	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	OFF	OFF	OFF	
2	OFF	ON	OFF	OFF	
3	OFF	OFF	ON	OFF	
4	OFF	OFF	OFF	ON	

Tabla 3-2 Secuencia tipo wave drive.

**Secuencia de medio paso**

Esta secuencia es una combinación de las anteriores, las bobinas se activan de tal forma que el movimiento resultante es un desplazamiento igual a la mitad del paso real. Para ello se activan dos bobinas y después una de forma sucesiva. El movimiento resultante se puede observar en la siguiente tabla:

Paso	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	OFF	OFF	OFF	
2	ON	ON	OFF	OFF	
3	OFF	ON	OFF	OFF	
4	OFF	ON	ON	OFF	
5	OFF	OFF	ON	OFF	
6	OFF	OFF	ON	ON	

7	OFF	OFF	OFF	ON	
8	ON	OFF	OFF	ON	

Tabla 3-3 Secuencia de medio paso.

### Secuencia para motores bipolares

Como se dijo anteriormente, estos motores necesitan variar el campo magnético en una secuencia determinada para generar el movimiento del eje del motor en un paso, cuyo sentido de giro está determinado por la secuencia seguida.

La secuencia necesaria para controlar los motores bipolares se muestra a continuación:

Paso	Terminales			
	A	C	B	D
1	+V	-V	+V	-V
2	+V	-V	-V	+V
3	-V	+V	-V	+V
4	-V	+V	+V	-V

Tabla 3-4 Secuencia para motores bipolares.

### Control de los motores de desplazamiento

Con el fin de poder obtener un mejor control de una manera más sencilla el sistema global se ha decidido crear una serie de controladores que manejen el movimiento automático de los componentes mecánicos del sistema.

### Clasificación de los controles

El comportamiento de los mecanismos de los motores de cada sección tiene características que nos permiten realizar la siguiente clasificación de controladores:

- Tipo I - Motor de desplazamiento unidireccional con control de paro
- Tipo II.- Motor de desplazamiento bidireccional con dos puntos de límite de desplazamiento con control de dirección y paro automáticos.
- Tipo iii – motor de desplazamiento unidireccional con giro de 360°, con paro automático.

- Tipo iv – motor de desplazamiento bidireccional con accionamiento mediante pulso de entrada, con control de dirección y paro automático.
- Tipo v – motor con desplazamiento entre 2 puntos con estado de espera accionado por el pulso de entrada, con control de dirección y paro automáticos.

Cabe recordar que cada motor corresponde a un movimiento automático de cada proceso en cuestión, los cuales se describirán más adelante en este capítulo. Dada la complejidad de la lógica de control y con el fin de minimizar el número de componentes en las tabletas, elegimos utilizar arreglos lógicos genéricos (GAL), y en especial el GAL 16V8 y para su programación se empleó el programa OPAL de National Semiconductors.

### Tipo i

Este control corresponde a la banda, en si cada control posee un controlador tipo i, ya que este es el responsable de la generación de los pulsos de excitación que se deberán proporcionar para que gire un motor de pasos, se observó que la mejor manera de generar un buen par, después de varias pruebas es la secuencia del wave drive, es decir en cada momento existirá solo un pulso para una de las cuatros fases, la secuencia del pulso variara dependiendo del valor de la entrada *dir*, así para un valor lógico de cero el giro se hará en forma ascendente, mientras que para un valor lógico de uno el giro se hará en forma descendente, además se tiene un bit de control de paro, de tal forma que cuando su valor sea 1 se realizará el giro conforme lo marque la dirección, mientras que con un 0, no se tendrá nada en la salida:

<i>dir</i>	<i>d</i>	<i>c</i>	<i>b</i>	<i>a</i>
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
1	0	0	0	1
1	1	0	0	0
1	0	1	0	0
1	0	0	1	0

Tabla 3-5 Determinación del giro para un control tipo I.

Como se puede observar el giro depende del bit de *dir*, y si incluimos un bit de *stop* (paro) tendremos la siguiente tabla de funciones<sup>4</sup>:

<i>stop</i>	<i>dir</i>	<i>d</i>	<i>c</i>	<i>b</i>	<i>a</i>	<i>d<sub>+1</sub></i>	<i>c<sub>+1</sub></i>	<i>b<sub>+1</sub></i>	<i>a<sub>+1</sub></i>
0	*	*	*	*	*	0	0	0	0
1	*	0	0	0	0	0	0	0	1
1	0	0	0	0	1	0	0	1	0
1	1	0	0	0	1	1	0	0	0
1	0	0	0	1	*	0	1	0	0
1	1	0	0	1	*	0	0	0	1
1	0	0	1	*	*	1	0	0	0
1	1	0	1	*	*	0	0	1	0
1	0	1	*	*	*	0	0	0	1
1	1	1	*	*	*	0	1	0	0

Tabla 3-6 Movimiento completo para el control tipo i.

<sup>4</sup> Los asteriscos en la tabla 3-6, indican que son valores intrascendentes, ya que tomamos el 1 lógico mas significativo, para esto todos los ceros a izquierda determinan que tan significativo será el 1 lógico.

Cabe recordar que este es un circuito secuencial basado en la lógica de operación de flip-flops tipo D, para poder llevar a cabo nuestras ecuaciones características del circuito tenemos que  $d_{+1}$ ,  $c_{+1}$ ,  $b_{+1}$ ,  $a_{+1}$ , los representaremos en las siguientes ecuaciones como d, c, b y a, esto es con el fin de poder declarar los valores de las terminales de retroalimentación (feedback) dentro del GAL, con lo anterior encontramos que la solución para nuestra tabla de funciones será:

$$\begin{aligned}d &= stop \cdot dir \cdot \bar{d} \cdot \bar{c} \cdot \bar{b} \cdot a + stop \cdot \bar{dir} \cdot \bar{d} \cdot c \\c &= stop \cdot \bar{dir} \cdot \bar{d} \cdot \bar{c} \cdot b + stop \cdot dir \cdot d \\b &= stop \cdot \bar{dir} \cdot \bar{d} \cdot \bar{c} \cdot \bar{b} \cdot a + stop \cdot dir \cdot \bar{d} \cdot c \\a &= stop \cdot \bar{d} \cdot \bar{c} \cdot \bar{b} \cdot \bar{a} + stop \cdot dir \cdot \bar{d} \cdot \bar{c} \cdot b + stop \cdot \bar{dir} \cdot d\end{aligned}$$

Con estas ecuaciones utilizaremos el programa compilador OPAL. En primer término se crea un archivo ASCII con las ecuaciones solución, mismo que se grabará con una extensión eqn "**mi\_archivo.eqn**", dado que estamos empleando una lógica secuencial, se deberán agregar dos puntos antes del signo igual cuando ingresemos las ecuaciones al archivo, también se deberán declarar las entradas y las salidas y su correspondencia para cada pin del GAL, para nuestro caso el bit de **stop** se encuentra en el pin número 2, el bit de **dir** será el pin 3, en ambos son entradas. En el caso de las salidas **a** corresponderá al pin 19, **b** al pin 18, **c** al pin 17 y **d** al pin 16, de forma que nuestro programa será como sigue:

```
;translated from nsc formatted pla file.
chip tipoi 16v8
stop=2 dir=3 d=16 c=17 b=18 a=19
equations
d := stop * dir * /d * /c * /b * a
    + stop * /dir * /d * c
c := stop * /dir * /d * /c * b
    + stop * dir * d
b := stop * /dir * /d * /c * /b * a
    + stop * dir * /d * c
a := stop * /d * /c * /b * /a
    + stop * dir * /d * /c * b
    + stop * /dir * d
```

Como observamos en la línea 2 definimos la palabra chip, aparte aparece el nombre del archivo tipo i, con este nombre se deberá guardar nuestro archivo y para este caso, nuestro archivo deberá llamarse: "**tipoi.eqn**"; asimismo en esta misma línea definimos que estamos empleando una GAL 16V8; en la siguiente línea definimos los pines o patas de la GAL, las únicas patas reservadas son la 1, que corresponde al reloj de operación, la 10 que corresponde a nuestra tierra, la 11 que corresponde a la función output enable (OE) y que se activa con un nivel bajo y finalmente la pata 20 que corresponde al Vcc, fuera de estos pines reservados podemos asignar los pines del 2 al 9 para entradas y los pines del 12 al 19 para entradas o salidas.

Con el archivo EQN ya corregido, se debe transformar a un archivo tipo JEDEC, para esto el compilador OPAL ofrece una opción para transformación directa EQN2JED, una vez generado el archivo es posible grabar la GAL, mediante un grabador de memorias, el archivo "**tipoi.jed**" será el siguiente:



salidas **stop** y **dir**, mientras que sus entradas serán **Ar**, **s1** y **s2**. Hay tres estados definidos que operan en este controlador:

1. Inicio (00): en este estado el controlador esta en espera del pulso de arranque, mientras tanto las salidas serán: **dir** = 0 y **stop** = 0. Cuando se active el bit **Ar** se pasará al estado 2. También es importante mencionar que en este estado el registro tanto en el bit de **s1** como en el de **s2**, resultan intrascendentes.
2. Avance (01): bajo este estado el controlador deberá indicar al motor un avance hasta el punto b, para lograrlo las salidas serán: **dir** = 0 y **stop** = 1, el avance continuará mientras el bit **s1** sea 0, tanto el bit **Ar**, como el bit **s2** serán intrascendentes, cuando el bit **s1** sea 1, se pasará al estado 3.
3. Reversa (10): ahora el motor deberá girar en sentido inverso hasta que se llegue al punto a, en este caso las salidas serán: **dir** = 1 y **stop** = 1, este estado permanecerá mientras el bit de **s2** sea 0, el bit **Ar** y **s1**, resultan intrascendentes en este estado. Al activarse el bit **s2**, se ira al estado 1.

Cuando estamos diseñando también se debe considerar la presencia de un estado no definido que puede presentarse a la hora de activar el dispositivo, para este caso se elige al estado 11 como no definido, para evitar problemas con el control, se envía al estado de reversa, para poder asegurar que el recorrido podrá empezar en el punto a.

El diagrama de estados siguiente muestra el funcionamiento del controlador tipo ii

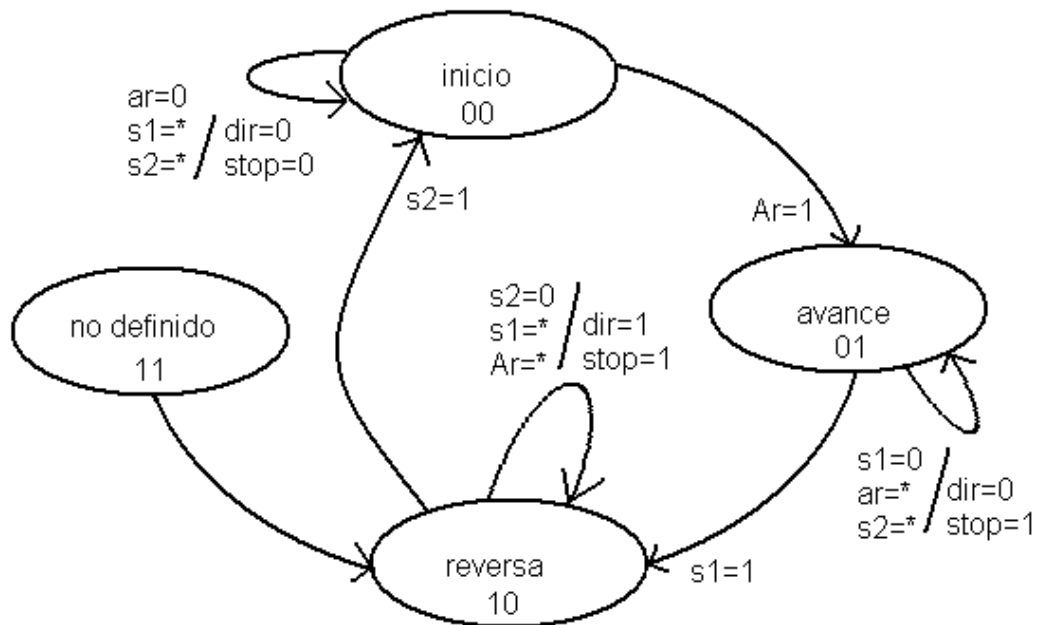


Figura 3-44 Diagrama de estados para el control tipo II.



En este caso, la tabla de funciones será la siguiente:

<i>Ar</i>	<i>s1</i>	<i>s2</i>	<i>b</i>	<i>a</i>	<i>b<sub>+1</sub></i>	<i>a<sub>+1</sub></i>	<i>dir</i>	<i>stop</i>
0	*	*	0	0	0	0	0	0
1	*	*	0	0	0	1	0	1
*	0	*	0	1	0	1	0	1
*	1	*	0	1	1	0	1	1
*	*	0	1	0	1	0	1	1
*	*	1	1	0	0	0	0	0
*	*	*	1	1	1	0	1	1

Tabla 3-7 Tabla de funciones para el control tipo II.

Basándonos en el mapa anterior se obtienen las siguientes ecuaciones:

$$b = s1 \cdot \bar{b} \cdot a + \bar{s2} \cdot b \cdot \bar{a} + b \cdot a$$

$$a = Ar \cdot \bar{b} \cdot \bar{a} + \bar{s1} \cdot \bar{b} \cdot a$$

$$dir = s1 \cdot \bar{b} \cdot a + \bar{s2} \cdot b \cdot \bar{a} + b \cdot a$$

$$stop = Ar \cdot \bar{b} \cdot \bar{a} + \bar{s2} \cdot b \cdot \bar{a} + a$$

Finalmente, el archivo “*tipoi.eqn*”, con el cual se inicia el proceso para programar la GAL será:

```

;controlador tipo ii
chip tipoi 16v8
ar=2 s1=3 s2=4 stop=19 dir=18 a=17 b=16
equations
  b := s1 * /b * a
      + /s2 * b / * a
      + b * a
  a := ar * /b * /a
      + /s1 * /b * a
  dir := s1 * /b * a
        + /s2 * b * /a
        + b * a
  stop := ar * /b * /a
         + /s2 * b * /a
         + a
    
```

### Tipo iii

Este control se diseño para el motor dedicado a retirar las botellas de la banda, el cual realiza un giro de 360°, para que la pala avance y retroceda en la banda, una descripción gráfica de su accionamiento, lo podemos observar en la siguiente figura:

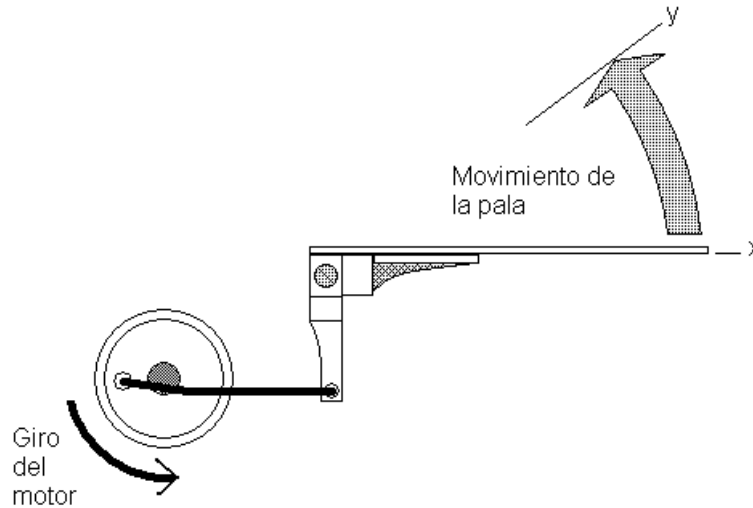


Figura 3-45 Descripción del movimiento físico controlado por un control tipo III.

Como se puede observar, el movimiento la pala comienza en el punto x y tiene un desplazamiento máximo en y, dado que el motor continua su giro la pala ahora continuará su recorrido en forma inversa es decir del punto y al punto x.

El control para este motor deberá contemplar un punto de inicio en x, a partir de este el motor deberá realizar una rotación hasta que la pala vuelva otra vez al punto x.

Los criterios de diseño contemplan un bit de arranque **Ar**, dos interruptores de posición **s1** y **s2**, teniendo en cuenta que **s2** deberá estar en el punto x, mientras que **s1**, deberá estar en otra posición de preferencia en el punto y, ya sea en la pala, o bien, en las otras componentes mecánicas del arreglo.

Al igual que en el control tipo ii, este control funciona junto con un control tipo i, por lo cual nos deberá entregar a la salida los bits de **stop** y **dir**.

El funcionamiento del arreglo sería el siguiente:

1. Inicio (00) : En este estado el motor esperará al bit de arranque **Ar**, en este estado los bits de **s1** y **s2**, son intrascendentes, las salidas serán **stop** = 0 y **dir** = 0, cuando **Ar** sea un 1 lógico se pasará al estado 2.
2. Fase 1 (01): En realidad es un estado intermedio, que es necesario para que el desplazamiento de la pala continúe y se libere **s2**, en este caso tanto **Ar** y **s2** son intrascendentes y las salidas serán **dir** = 0 y **stop** = 1, el control se mantendrá en este estado mientras que **s1** sea 0, al ocurrir un 1 en **s1** se deberá pasar el estado 3.
3. Fase 2 (10): En este estado el motor continuará su giro hasta que la pala vuelva al punto x, al llegar a este punto el motor deberá parar, para esto la pala accionará a **s2**. Tanto **Ar** como **s1** resultan intrascendentes y las salidas serán **dir** = 0 y **stop** = 1, este estado se mantendrá mientras que **s2** sea 0, cuando **s2** sea 1 se pasará al estado 1.
4. No definido (11): Es un estado que se puede presentar y que se deberá tomar decisión sobre las acciones a tomar, para evitar problemas se hace que el motor gire en reversa hasta llegar al punto x. Para esto las salidas serán **dir** = 1 y **stop** = 1, en este caso tanto **Ar** como **s1** resultan intrascendentes, el estado se mantendrá mientras **s2** sea 0, cuando **s2** sea 1 se pasará al estado 1.

La siguiente figura muestra el diagrama de estados del controlador tipo iii:

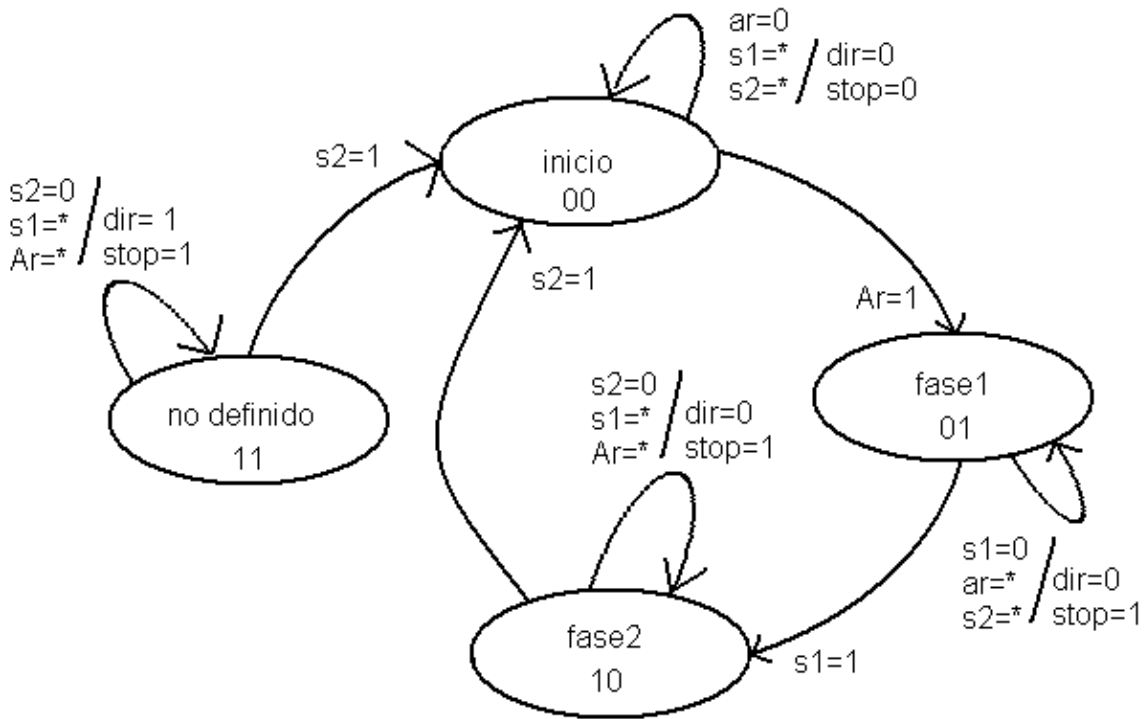


Figura 3-46 Diagrama de estados para el control tipo III.

De lo anterior tendremos la siguiente tabla de funciones:

<i>Ar</i>	<i>s1</i>	<i>s2</i>	<i>b</i>	<i>a</i>	<i>b<sub>+1</sub></i>	<i>a<sub>+1</sub></i>	<i>dir</i>	<i>Stop</i>
0	*	*	0	0	0	0	0	0
1	*	*	0	0	0	1	0	1
*	0	*	0	1	0	1	0	1
*	1	*	0	1	1	0	0	1
*	*	0	1	0	1	0	0	1
*	*	1	1	0	0	0	0	0
*	*	0	1	1	1	1	1	1
*	*	1	1	1	0	0	0	0

Tabla 3-8 Tabla de funciones para el control tipo III.

Las ecuaciones que describen a este controlador son:

$$\begin{aligned}
 b &= s1 \cdot \bar{b} \cdot a + \bar{s2} \cdot b \cdot \bar{a} + \bar{s2} \cdot b \cdot a \\
 a &= Ar \cdot \bar{b} \cdot \bar{a} + \bar{s1} \cdot \bar{b} \cdot a + \bar{s2} \cdot b \cdot a \\
 dir &= \bar{s2} \cdot b \cdot a \\
 stop &= Ar \cdot \bar{b} \cdot \bar{a} + \bar{b} \cdot a + \bar{s2} \cdot b
 \end{aligned}$$

El archivo “*tipooiii.eqn*” será:

```

;controlador tipo iii
chip tipooiii 16v8
ar=2 s1=3 s2=4 stop=19 dir=18 a=17 b=16
equations
  b := s1 * /b * a
      + /s2 * b * /a
      + /s2 * b * a
  a := ar * /b * /a
      + /s1 * /b * a
      + /s2 * b * a
  dir := /s2 * b * a
  stop := ar * /b * /a
      + /b * a
      + /s2 * b

```

Cabe mencionar que al igual que el controlador tipo ii, el tipo iii comparte el mismo patigrama de la GAL, este tipo de construcción modular nos permitirá utilizar un solo tipo de tableta (circuito impreso), la cual contendrá controladores tipo i junto con los del tipo ii, iii, iv y v.

### Tipo iv

Este controlador maneja el movimiento del motor de desplazamiento horizontal del llenado de las botellas. El movimiento empieza al presentarse el pulso de arranque (**Ar**) y mantendrá el movimiento siguiendo la botella sobre la banda hasta que se termine el pulso de arranque, en este instante el motor girará en reversa hasta que el motor regrese a la posición inicial, en la cual se accionará al interruptor 1 (**s1**), mandando al motor a un estado de espera del siguiente pulso **Ar**. El tiempo del desplazamiento hacia adelante se relaciona con la duración del llenado, por parte de la bomba del sistema.

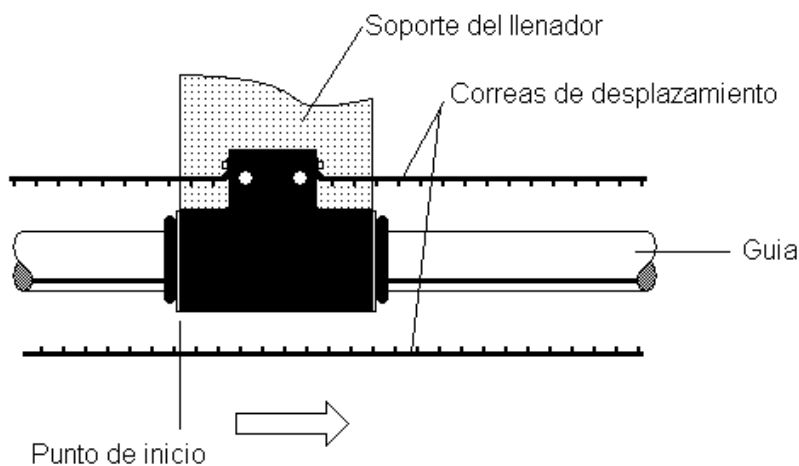


Figura 3-47. Dispositivo mecánico de desplazamiento para el controlador tipo IV.

El motor impulsa la banda dentada de desplazamiento, con el cual hay un movimiento del dispositivo de llenado durante el proceso de llenado de la botella.

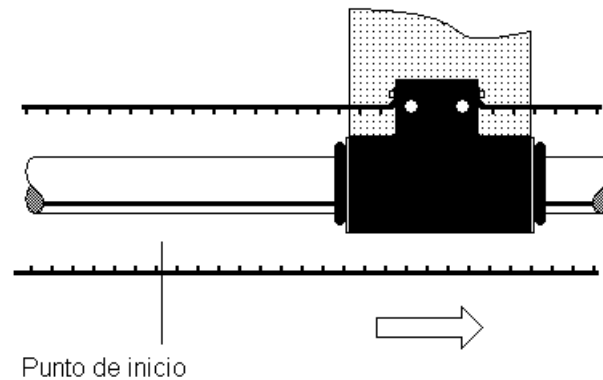


Figura 3-48 Primer movimiento para desplazar la sección de llenado.

Cuando el llenado ha finalizado, el motor girará en sentido contrario para que el soporte regrese a su punto inicial.

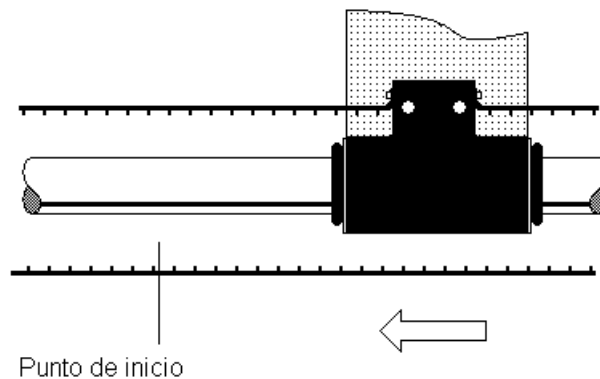


Figura 3-49 Segundo movimiento para regresar al punto de inicio de la sección de llenado.

Para este control se empleará un bit de Arranque **Ar** y un interruptor **s1**, el cual se activará en el punto de inicio, además las salidas deberán ser **dir** y **stop**. Los estados de operación serán los siguientes:

1. Inicio (00): En este estado el controlador esperará que **Ar** se active en nivel alto, **s1** es intrascendente y las salidas serán **dir** = 0 y **stop** = 0. Cuando **Ar** = 1, se pasará al estado 2.
2. Avance (01): En este estado el controlador hace que el motor gire mientras **Ar** sea 1, en este estado **s1** es intrascendente y las salidas serán **dir** = 0 y **stop** = 1, cuando **Ar** = 0 el controlador pasará el estado 3.
3. Reversa (10): Cuando se presenta este estado el motor deberá invertir su giro, con el fin de alcanzar el punto de inicio, esto se mantendrá mientras **s1** sea 0, para este estado **Ar** es intrascendente y las salidas serán **dir** = 1 y **stop** = 1, cuando **s1** = 1 el controlador deberá pasar al estado 1.

4. No definido (11): Al presentarse este estado se mandará al control al estado 3 (reversa), en este estado tanto **Ar** como **s1** son intrascendentes y las salidas serán **dir = 1** y **stop = 1**.

Gráficamente el diagrama de estados será el siguiente:

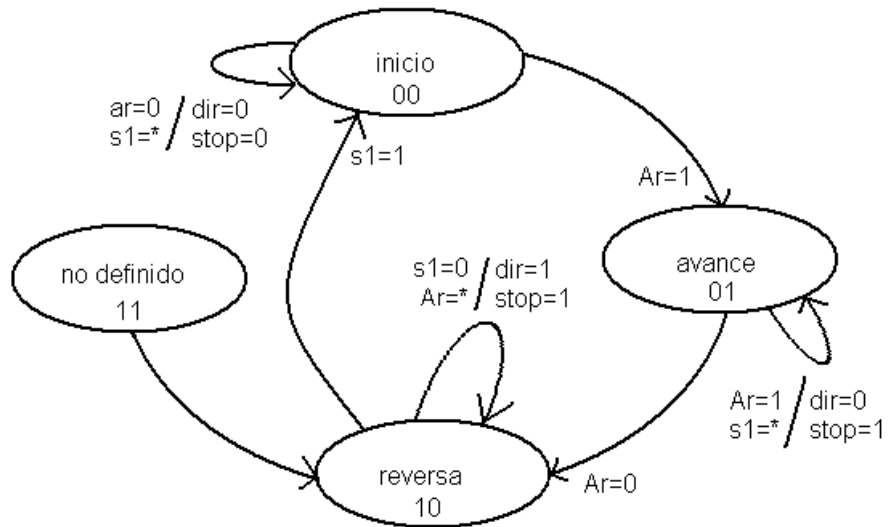


Figura 3-50 Diagrama de estados para el control tipo IV.

Su tabla de funciones para el diagrama anterior se muestra a continuación:

<b>Ar</b>	<b>s1</b>	<b>b</b>	<b>a</b>	<b>b<sub>+1</sub></b>	<b>a<sub>+1</sub></b>	<b>dir</b>	<b>stop</b>
0	*	0	0	0	0	0	0
1	*	0	0	0	1	0	1
0	*	0	1	1	0	1	1
1	*	0	1	0	1	0	1
*	0	1	0	1	0	1	1
*	1	1	0	0	0	0	0
*	*	1	1	1	0	1	1

Tabla 3-9 Tabla de funciones para el control tipo IV.

Las ecuaciones que definen este sistema secuencial serán:

$$\begin{aligned}
 b &= \overline{Ar} \cdot \overline{b} \cdot a + \overline{s1} \cdot b \cdot \overline{a} + b \cdot a \\
 a &= Ar \cdot \overline{b} \cdot \overline{a} + Ar \cdot \overline{b} \cdot a \\
 dir &= \overline{Ar} \cdot \overline{b} \cdot a + \overline{s1} \cdot b \cdot \overline{a} + b \cdot a \\
 stop &= Ar \cdot \overline{b} \cdot \overline{a} + \overline{s1} \cdot b \cdot \overline{a} + a
 \end{aligned}$$

Finalmente el archivo "*tipoiv.eqn*" será:

```

;controlador tipo iv
chip tipoiv 16v8
ar=2 s1=3 stop=19 dir=18 a=17 b=16
equations
b := /ar * /b * a
  + /s1 * b * /a + b * a
a := ar * /b * /a
  + ar * /b * a
dir := / ar * /b * a
  + /s1 * b * /a + b * a
stop := ar * /b * /a
  + /s1 * b * /a + a

```

### Tipo v

Este control maneja el motor de desplazamiento vertical del llenador de botellas, dado que en este elemento hay una boquilla mediante la cual fluye líquido hacia las botellas, se ha pensado en este movimiento que es capaz de acercarla lo suficiente para evitar que el líquido pudiese derramarse fuera de las botellas, como en controles anteriores el movimiento comienza cuando se manda un pulso de arranque, el movimiento continuará hasta que se llegue al límite inferior de desplazamiento, dado que el proceso de bombeo del líquido comenzará después de que la boquilla se acerque lo suficiente a las botellas se requiere un pulso de arranque sostenido hasta que se complete el llenado, una vez que ocurra esto el pulso **Ar** deberá desactivarse, con lo cual el motor girará en sentido inverso hasta que se llegue a su posición inicial, esperando el próximo pulso **Ar**.

En la siguiente gráfica se muestran algunos componentes del dispositivo de desplazamiento vertical:

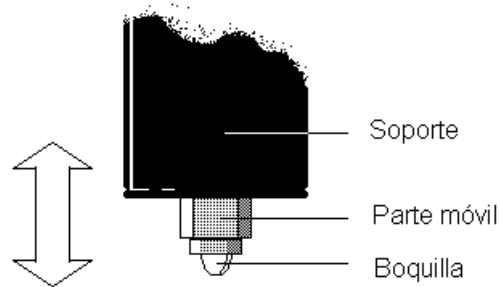


Figura 3-51 Sistema de la boquilla para el sistema del llenado.

Las siguientes gráficas muestran los movimientos involucrados con este dispositivo:

- Inicio : el motor esta en reposo en espera del bit de arranque **Ar**.

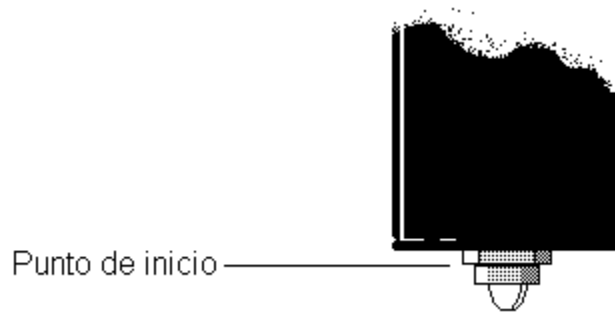


Figura 3-52 Punto de inicio del desplazamiento de la boquilla.

- Descenso : El motor gira hasta que se alcanza el límite inferior de desplazamiento, en el cual se acciona el interruptor 1 (**s1**), lo cual hace que el motor pare. Aunque se ha diseñado para que **Ar** sea intrascendente, en este estado este bit se mantiene en alto.

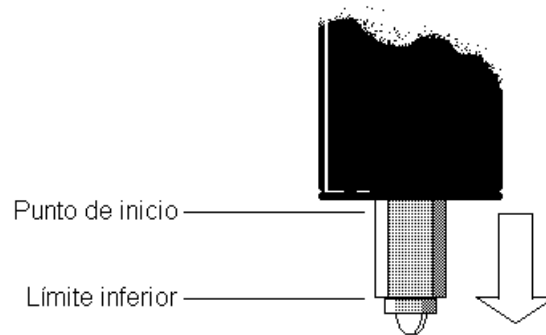


Figura 3-53 Movimiento de descenso de la boquilla.



- Llenado : En este estado se accionará la bomba y fluirá líquido hacia las botellas por la boquilla, cabe recordar que mientras se realiza el llenado **Ar** se deberá mantener en alto, esto es muy similar con lo que ocurre con el control tipo iv.

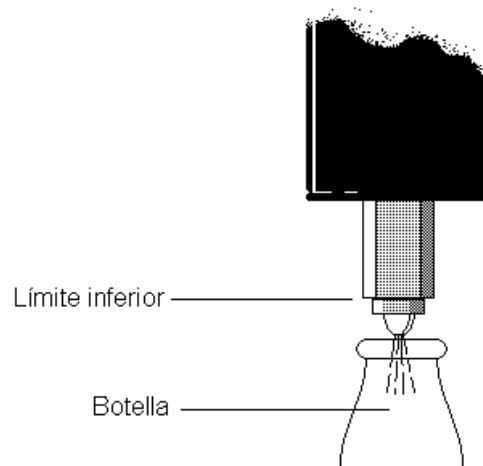


Figura 3-54 Llenado de botella

- Ascenso : Una vez que se ha terminado el llenado **Ar** deberá desactivarse, lo cual hace que el motor comience a girar en sentido contrario, llevando a la pieza móvil hasta el punto de inicio donde se activará el interruptor 2 (**s2**), el cual llevará al controlador al estado de inicio y se esperará el siguiente pulso de arranque.

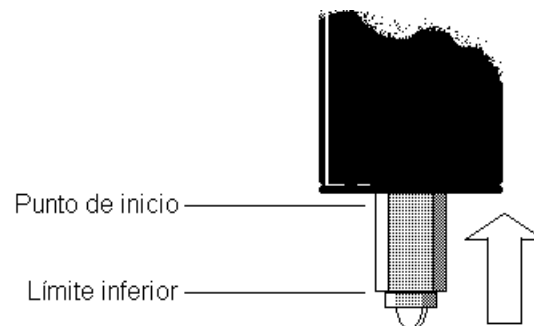


Figura 3-55 Ascenso de la boquilla.

Como se ha visto en este diseño volvemos a involucrar a **Ar**, **s1** y **s2**, al igual que en los controles anteriores las salidas serán **dir** y **stop**.

Los estados presentarán las siguientes combinaciones de entradas y salidas:

1. Inicio (00): Se mantendrá si **Ar** = 0, tanto **s1** como **s2** son intrascendentes, las salidas serán **dir** = 0 y **stop** = 0, solamente cuando **Ar** sea 1, se cambiará al estado 2.
2. Descenso (01): Se mantiene si **s1** = 0, tanto **Ar** como **s2** son intrascendentes, las salidas serán **dir** = 0 y **stop** = 1, si **s1** = 1 se cambiará al estado 3. Cabe mencionar que aunque **Ar** es intrascendente en este estado se recomienda mantenerlo como 1.
3. Llenado (10): En este estado **s1** y **s2** son intrascendentes, las salidas serán **dir** = 1 y **stop** = 0, se mantendrá este estado mientras **Ar** = 1, en caso contrario (**Ar** = 0), se cambia al estado 4.
4. Ascenso (11): Tanto **Ar** como **s1** no importan en este estado, las salidas serán **dir** = 1 y **stop** = 1, el estado se mantendrá mientras **s2**=0, si **s2** = 1 se cambiará al estado 1.

El diagrama de estados para este controlador es el siguiente:

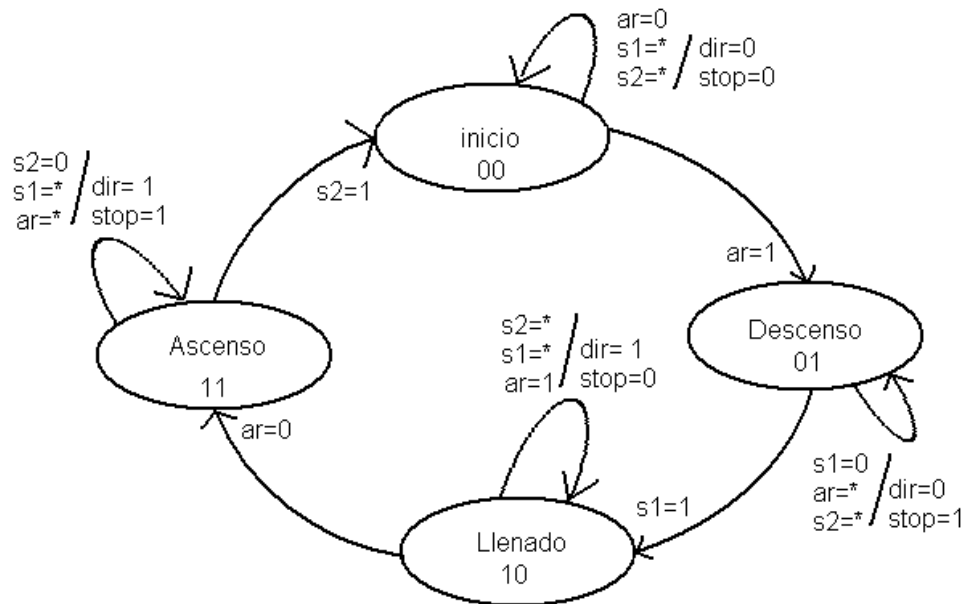


Figura 3-56 Diagrama de estados para el control tipo V.

La tabla de funciones se muestra a continuación:

Ar	s1	s2	b	a	b <sub>+1</sub>	a <sub>+1</sub>	dir	stop
0	*	*	0	0	0	0	0	0
1	*	*	0	0	0	1	0	1
*	0	*	0	1	0	1	0	1
*	1	*	0	1	1	0	1	0
0	*	*	1	0	1	1	1	1
1	*	*	1	0	1	0	1	0
*	*	0	1	1	1	1	1	1
*	*	1	1	1	0	0	0	0

Tabla 3-10 Tabla de funciones para el control tipo V.

De la tabla de funciones anterior obtenemos las siguientes ecuaciones:

$$\begin{aligned}
 b &= s1 \cdot \bar{b} \cdot a + b \cdot \bar{a} + \bar{s2} \cdot b \cdot a \\
 a &= Ar \cdot \bar{b} \cdot \bar{a} + \bar{s1} \cdot \bar{b} \cdot a + \bar{Ar} \cdot b \cdot \bar{a} + \bar{s2} \cdot b \cdot a \\
 dir &= s1 \cdot \bar{b} \cdot a + b \cdot \bar{a} + \bar{s2} \cdot b \cdot a \\
 stop &= Ar \cdot \bar{b} \cdot \bar{a} + \bar{s1} \cdot \bar{b} \cdot a + \bar{Ar} \cdot b \cdot \bar{a} + \bar{s2} \cdot b \cdot a
 \end{aligned}$$

Finalmente el programa “*tipov.eqn*” se muestra a continuación:

```

;controlador tipo v
chip tipov 16v8
ar=2 s1=3 s2=4 stop=19 dir=18 a=17 b=16
equations
  b := s1 * /b * a
      + b * /a
      + /s2 * b * a
  a := ar * /b * /a
      + /s1 * /b * a
      + /ar * b * /a
      + /s2 * b * a
  dir :=s1 * /b * a
      + b * /a
      + /s2 * b * a
  stop := ar * /b * /a
      + /s1 * /b * a
      + /ar * b * /a
      + /s2 * b * a

```

## Detectores de botellas

Los detectores de botellas tienen como finalidad, enviar la información de la presencia de una botella en la cercanía de una sección del sistema, para esto los detectores son de tipo óptico, se emplea un foto diodo y un foto transistor como sensores, al no haber botella de por medio, el haz de luz infrarroja tendrá al fototransistor en saturación, lo cual no dará un nivel bajo a la salida del arreglo. Al interponerse una botella, dicho haz se interrumpe, lo cual provoca que el foto transistor se vaya a corte y la salida ahora será un nivel alto.

Para poder asegurar un nivel óptimo de operación se le agrega un comparador de nivel y con esto se asegura que el nivel a la salida sea el adecuado para activar al MOC3011, el cual es una interfase con el PLC.

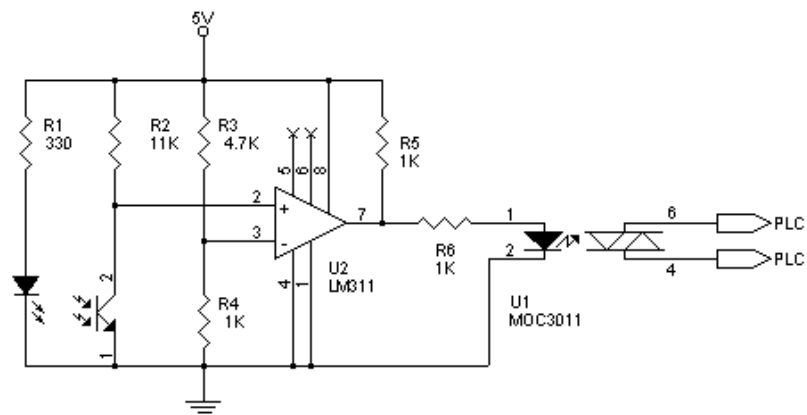


Figura 3-57 Circuito electrónico de los detectores de botella.

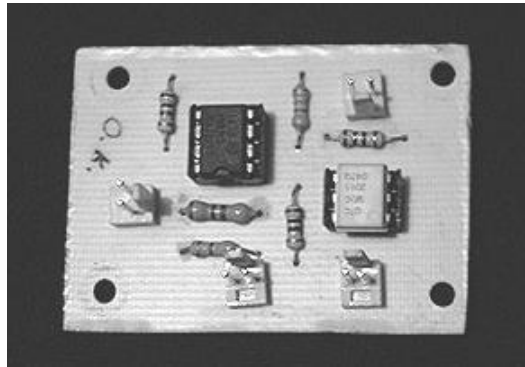


Figura 3-58 Tarjeta para el funcionamiento de los detectores de botellas.

El circuito anterior va montado dentro de una caja, como se muestra a continuación:

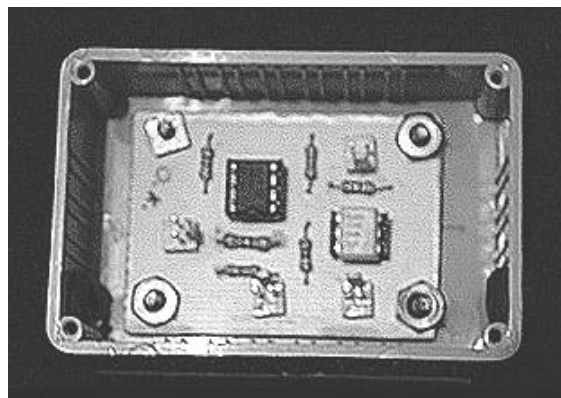


Figura 3-59 Montaje de la tarjeta de los detectores de botellas.

A esta caja llega la alimentación y el estado del par óptico, la salida va hacia el PLC. Externamente se montan tanto el foto diodo, como el foto transistor a la orilla de la banda como se muestra a continuación:

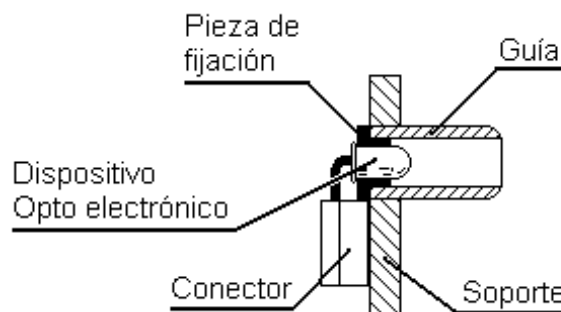
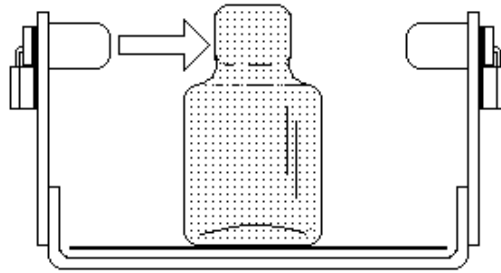


Figura 3-60 Montaje de los opto sensores en las estructuras del proceso.

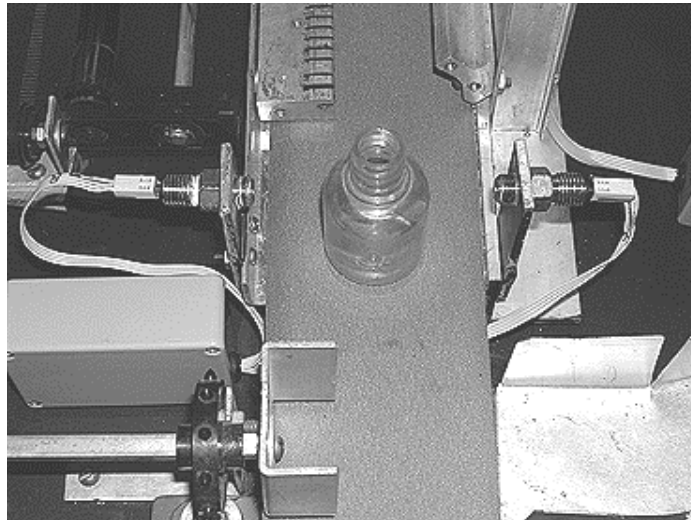
La ubicación de los detectores de botella está antes de las siguientes secciones: retirador, llenado y desviador; para el caso del despachador, los sensores van incluidos dentro de la misma estructura de éste.

Para las secciones de llenado y del retirador, se maneja el siguiente arreglo de sensores



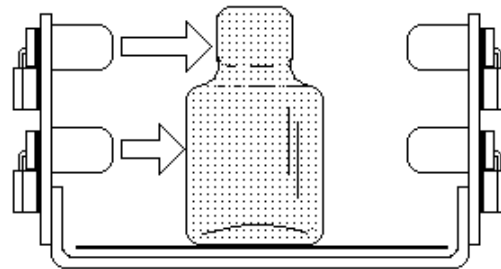
**Figura 3-61 Funcionamiento del opto sensor sencillo.**

Como se puede observar la botella simplemente corta el haz de luz, con lo cual se detecta su presencia y se activará la salida.

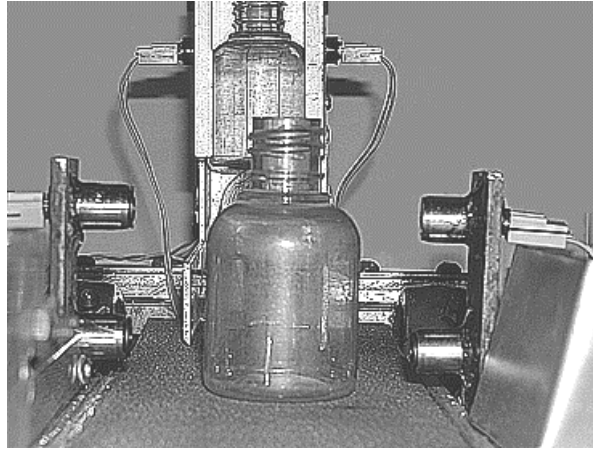


**Figura 3-62 Aspecto físico del opto sensor sencillo.**

En el caso del desviador se tiene un doble arreglo óptico, mediante el cual se podrá detectar cuando la botella se haya caído como se muestra en la siguiente figura:



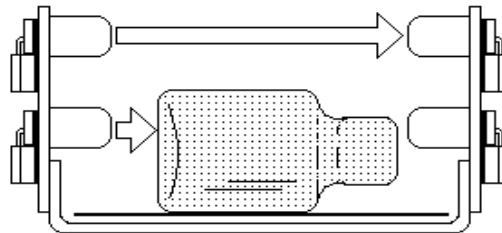
**Figura 3-63 Funcionamiento del opto sensor doble.**



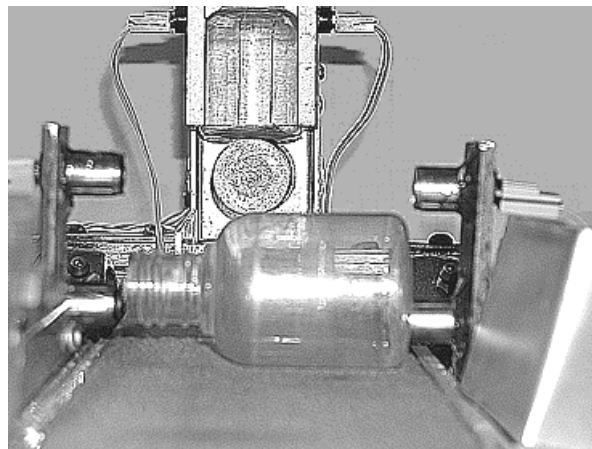
**Figura 3-64** Aspecto físico del opto sensor doble.

En la gráfica anterior se muestra como la botella interrumpe los dos haces de luz infrarroja, este es el indicativo que la botella se encuentra en posición correcta, esto hace que el desviador no se active.

Sin embargo, cuando la botella se cae, solo un haz llega hasta el receptor, como se ve a continuación:



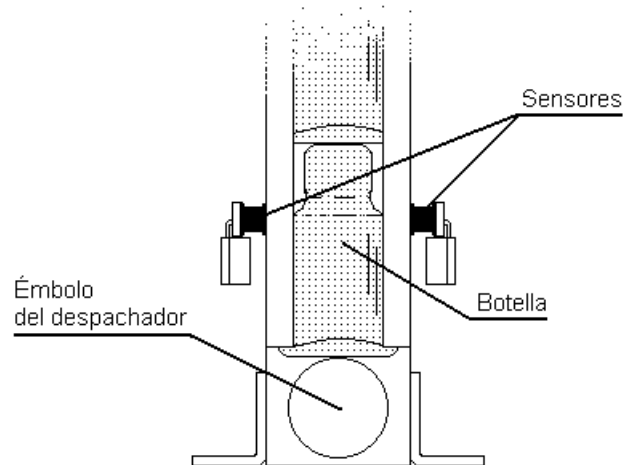
**Figura 3-65** Funcionamiento de un opto sensor doble con la botella caída.



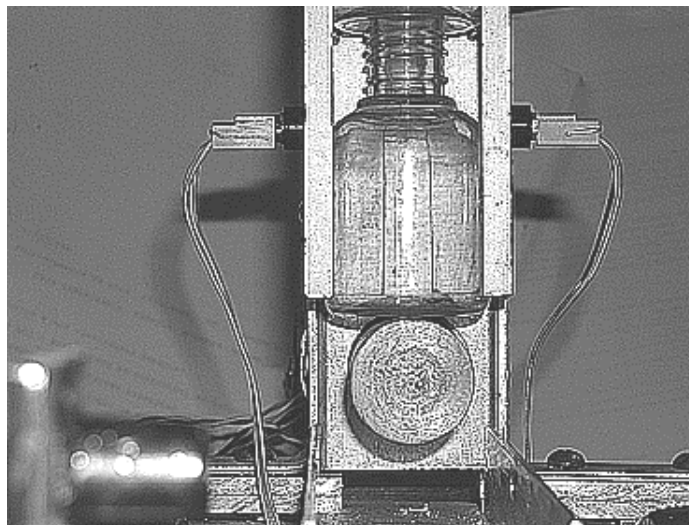
**Figura 3-66** Visualización de la botella caída.

Bajo esa condición, se determina que la botella se cayó, y se pone en operación al desviador.

Finalmente en la sección del despachador los opto sensores van montados sobre el riel de despacho, así se detecta una botella antes de que se despache como se muestra en la siguiente figura:



**Figura 3-67** Opto sensores montados sobre el riel de despacho de botellas.



**Figura 3-68** Aspecto físico del riel de despacho.

### Control maestro de arranque y paro

Con la finalidad de evitar arranques intempestivos cuando se energiza el sistema, al igual que parar el mismo cuando ocurre una emergencia, se debe agregar un control de arranque y paro que gobierne todo el sistema, la técnica más empleada incluye relevadores, nuestro circuito incluye el sistema de enclavamiento, lo cual nos permite con solo oprimir una vez el botón de arranque que la bobina del relevador siga conectada a la alimentación, por consiguiente la salida activará al PLC, hasta que el botón de paro sea pulsado, cuando esto ocurre, la bobina se desconectará de la alimentación y se termina el enclavamiento.

El siguiente circuito realiza la función antes mencionada:

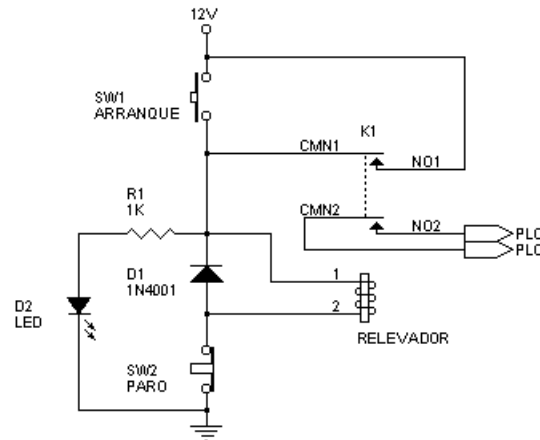


Figura 3-69 Circuito para el control maestro de arranque y paro.

Esta protección se emplea en procesos industriales reales aun de gran tamaño, que aunque estén totalmente automatizados con uno o varios PLC siempre tendrán un control de este tipo.

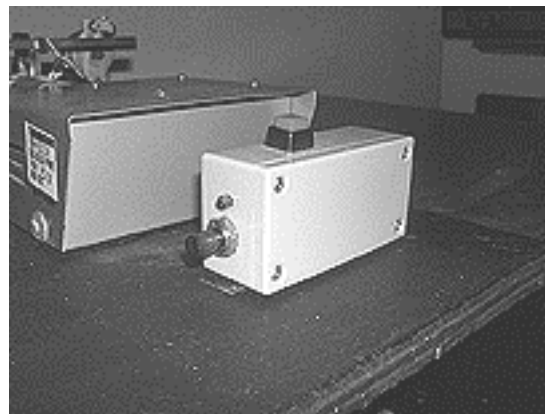


Figura 3-70 Control maestro de arranque y paro.



## Capítulo 4 Programación del PLC

### Entradas y salidas del PLC

Para programar el PLC, se debieron definir cuales son nuestras entradas del proceso hacia el PLC y con esto se generarán ciertas salidas para gobernar el proceso en forma global.

Cada entrada corresponderá a nuestros sensores de botellas instalados a lo largo del proceso, así como al control maestro de arranque y paro.

Dada su jerarquía, se definirán como se muestra en la siguiente tabla:

Elemento	Entrada del PLC
Control maestro de arranque y paro	I 0.0
Detector de botellas 1	I 0.1
Detector de botellas 2A	I 0.2
Detector de botellas 2B	I 0.3
Detector de botellas 3	I 2.0
Detector de botellas 4	I 2.1

Tabla 4-1 Entradas del PLC

Las salidas corresponden siempre a los actuadores del sistema, en este caso la planta tendrá como actuadores a los motores de pasos, a la bomba y a la electro válvula, el orden comienza desde el primer proceso y lo definiremos como se muestra en la tabla a continuación:

Elemento	Salida del PLC
Motor de la banda	Q 1.0
Motor del despachador	Q 1.1
Motor vertical del llenado	Q 1.2
Motor del desviador	Q 1.3
Motor horizontal del llenado	Q 3.0
Bomba	Q 3.1
Electro válvula	Q 3.2
Retirador	Q 3.3

Tabla 4-2 Salidas del PLC

Se consideraron otros elementos, como los temporizadores, que se definirán más adelante en esta misma sección, ya que estos elementos están vinculados en los diferentes procesos de la planta y así entender mejor la programación, la cual se ira mostrando para cada sección del proceso.

### Banda transportadora

La banda transportadora funcionará cuando el control maestro de arranque y paro se active.

El siguiente diagrama de flujo muestra este proceso:

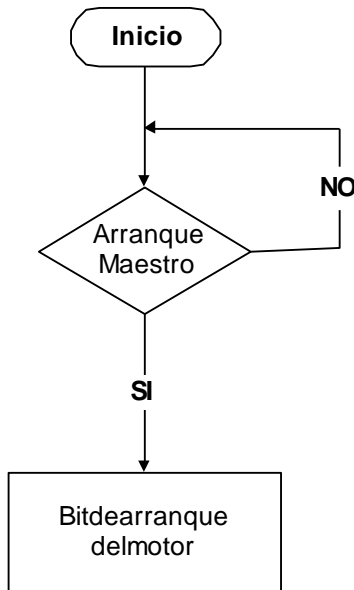
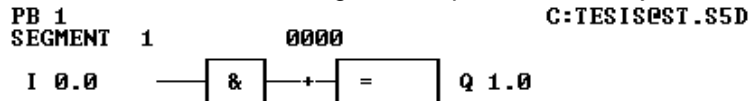
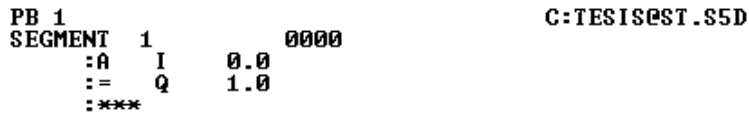


Figura 4-1 Diagrama de flujo del control de la banda

La programación en el PLC tendrá la siguiente representación bajo el formato CSF:



Mientras que para el formato STL el programa será la siguiente representación:



## Despachador de botellas

El despachador de botellas, dependerá del control maestro y de la presencia de una(s) botella(s) en su riel para ser despachada, esto lo realiza el detector de botellas 1, en este instante se activa al arranque del motor del despachador y se realiza un retardo de 20 segundos, con el fin de que no se junten las botellas en los procesos. La salida no se activará cuando el control maestro se desactive (OFF) y cuando ya no haya botellas en espera (detector de botellas 1).

El diagrama de flujo para este proceso se muestra a continuación:

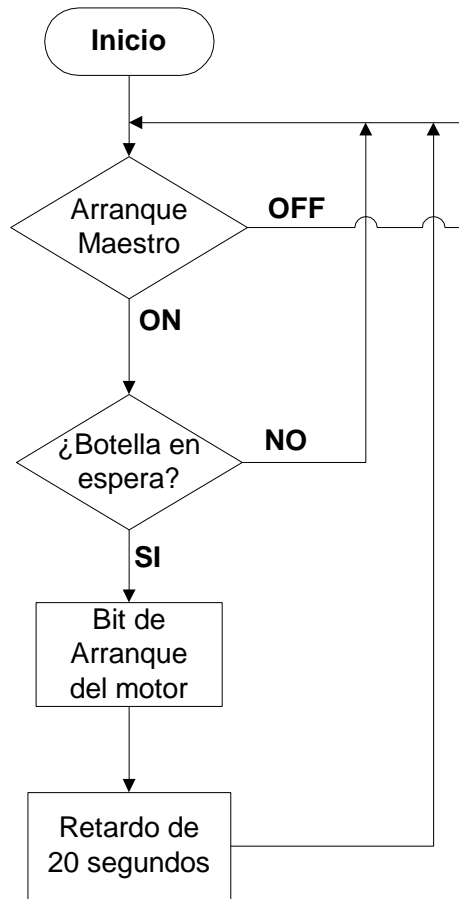


Figura 4-2 Diagrama de flujo del motor del despachador

Considerando el siguiente diagrama de tiempos para esta sección, tendremos:

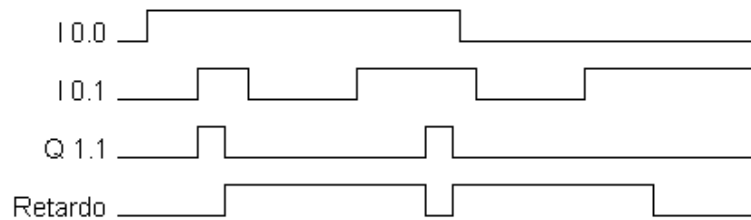
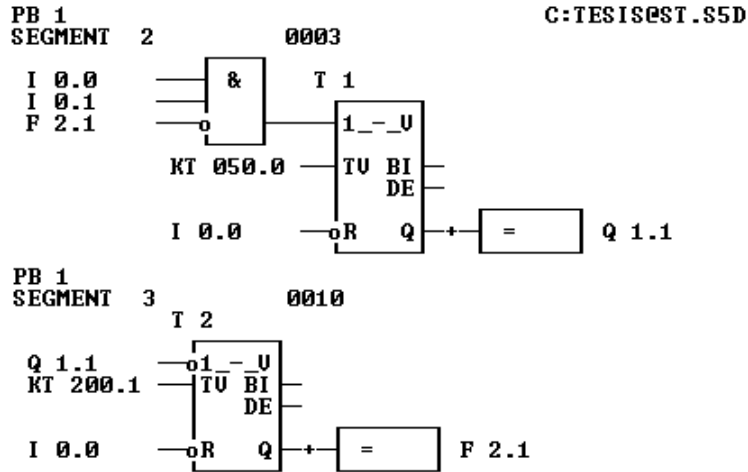


Figura 4-3 Diagrama de los bits de control para el despachador de botellas.

Para nuestro caso el tiempo de retardo será de 20 segundos mientras que Q1.1 se activará 0.5 segundos.

Esta sección del programa tendrá la siguiente representación en el modo CSF, como se puede observar T2 nos dará el retardo para despachar la siguiente botella, si se desea aumentar o disminuir este retardo simplemente se deberá variar la constante de tiempo que en este caso es de 20 segundos (KT 200.1).



El mismo programa tendrá la siguiente visualización en modo STL.

```

PB 1
SEGMENT 2          0003          C:TESISEST.S5D
:AND I 0.0
:AND I 0.1
:AND F 2.1
:L   KT 050.0
:SE  T 1
:AN  I 0.0
:R   T 1
:NOP 0
:NOP 0
:A   T 1
:=   Q 1.1
:***

PB 1
SEGMENT 3          0010
:AN  Q 1.1
:L   KT 200.1
:SE  T 2
:AN  I 0.0
:R   T 2
:NOP 0
:NOP 0
:A   T 2
:=   F 2.1
:***
    
```

### Desviador de botellas

El desviador de botellas se activará, cuando se detecte la presencia de una botella que se halla caído en el inicio del proceso.

Dado que hay una distancia entre el detector y la pala de desvío, se requiere de un retardo para activar el motor.

El diagrama de flujo se muestra a continuación:

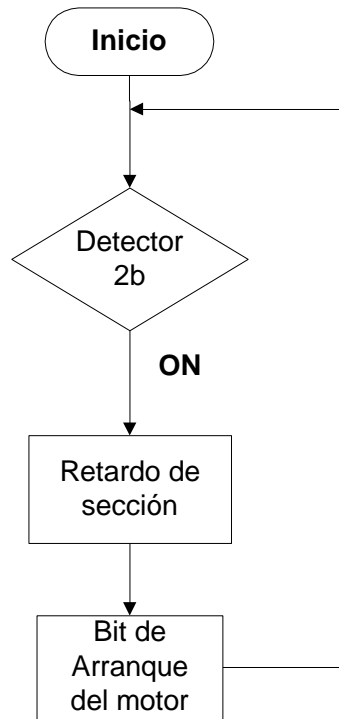


Figura 4-4 Diagrama de flujo del desviador de botellas

Para esta sección el diagrama de tiempos será:

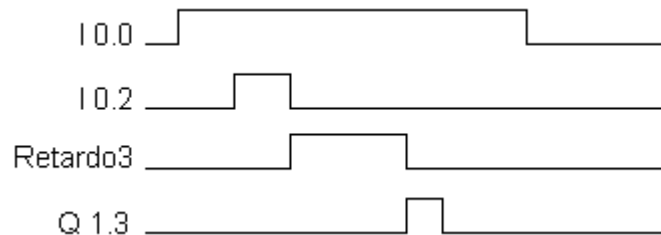
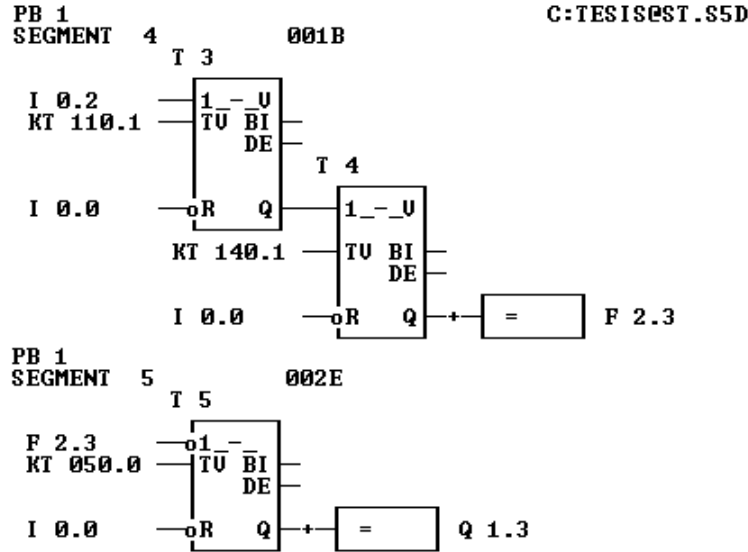


Figura 4-5 Diagramas de los bits de control para una botella caída

Como expliqué en el caso de que una botella se halla caída, solo se detectará la base de está, lo que provoca que la entrada I 0.2 se active, y se genere un pulso con retardo, necesario para que la botella llegue a la posición esperada y finalmente se activará el bit de arranque del motor del desviador a cargo de la salida Q 1.3. y sea retirada del proceso.

Esta sección del programa que se muestra a continuación en modo CSF, utiliza 2 temporizadores ( T3 y T4) y se debe a razones prácticas, ya que la botella nos generaba dos pulsos al estar caída, para evitar este desajuste se utilizó un segundo temporizador, para ajustar dicho problema de tal manera que el retardo resultante es de 14 segundos, después de este tiempo se activa la salida Q 1.3, para activar al desviador de botellas.



El programa de esta sección visualizado en STL se muestra a continuación:

```

PB 1                                C:TESISEST.S5D
SEGMENT 4                            001B
:A  I  0.2
:L  KT 110.1
:SE T  3
:AN I  0.0
:R  T  3
:NOP 0
:NOP 0
:A  T  3
:L  KT 140.1
:SE T  4
:AN I  0.0
:R  T  4
:NOP 0
:NOP 0
:A  T  4
:=  F  2.3
:***

SEGMENT 5                            002E
:AN F  2.3
:L  KT 050.0
:SP T  5
:AN I  0.0
:R  T  5
:NOP 0
:NOP 0
:A  T  5
:=  Q  1.3
:***
    
```

### Llenado de botellas

Este proceso involucra a dos motores de pasos, los cuales se activarán al mismo tiempo, esta salida se tiene que mantener activada hasta que la botella se haya llenado, por su parte la bomba y la electro válvula también se activan, después de un retardo, dado que la boquilla debe estar en posición, con el fin de evitar que el líquido se derrame fuera de las botellas.

Otra consideración que se debe tomar en cuenta es que existe una distancia entre el detector de botellas 3 y el punto de inicio del motor, así que debemos considerar otro retardo.

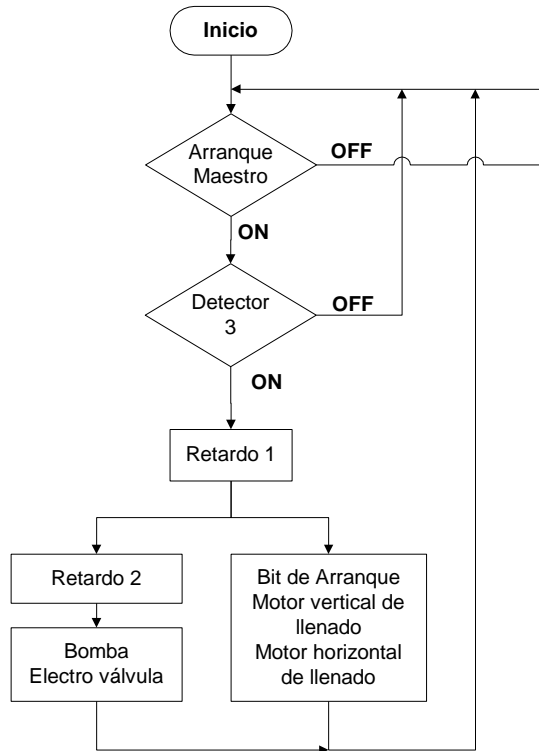


Figura 4-6 Diagrama de flujo del llenado de botellas

Su diagrama de tiempos se muestra a continuación:

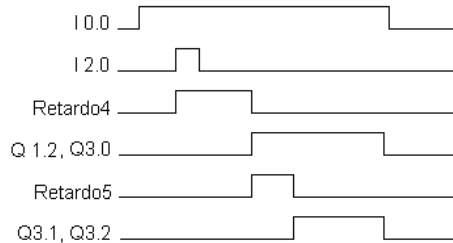
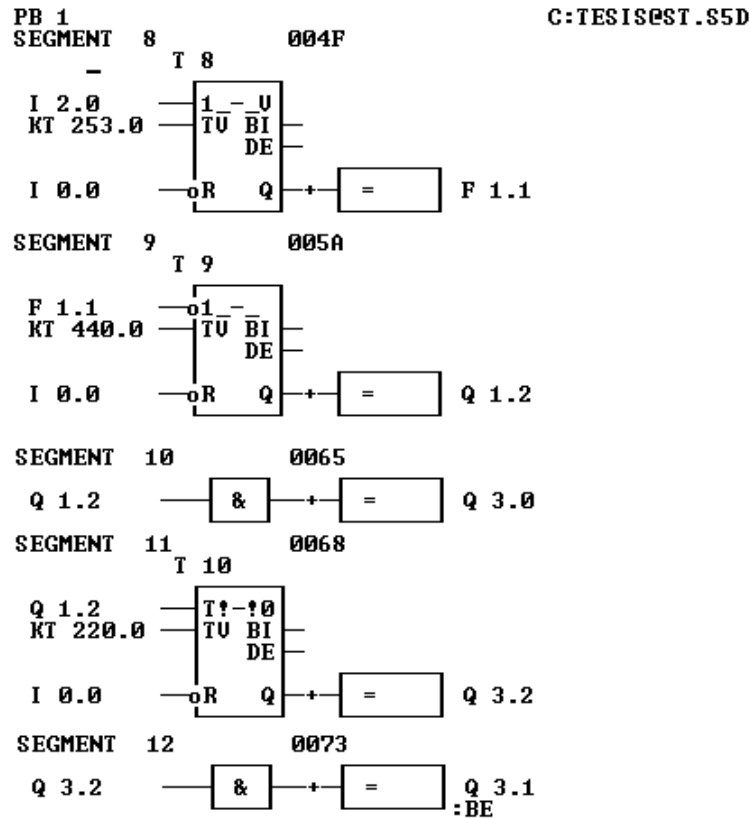


Figura 4-7 Diagrama de los bits de control para el llenado de botellas.

Como se puede observar cuando se detecta la botella (I 2.0) y el control maestro está activado, se genera el retardo 4 necesario para que la botella llegue el punto de inicio del llenado de botellas este tiempo es de 2.53 segundos, allí se activan los 2 motores de desplazamiento y se mantiene el pulso, al mismo tiempo se genera el retardo 5 (2.2 segundos), el cual da tiempo para que la boquilla se ponga en posición de llenado, en este instante se activan la bomba y la electro válvula durante un periodo necesario para llenar la botella (también de 2.2 segundos), cuando se cumple esto, tanto la salida de los motores ( Q 1.2 y Q3.0) así como la bomba y la electro válvula (Q3.1 y Q3.2 respectivamente) se desactivan, lo cual hace que el control automático lleve al sistema a su punto de inicio y espere a la siguiente botella para repetir el mismo proceso.

El programa de esta sección se muestra a continuación cabe recordar que es una visualización en modo CSF.





El mismo programa se muestra ahora en el modo STL

```

PB 1                                C:TESIS0ST.S5D
SEGMENT 8                            004F
:A  I    2.0
:L  KT 253.0
:SE T    8
:AN  I    0.0
:R  T    8
:NOP 0
:NOP 0
:A  T    8
:=  F    1.1
:***

SEGMENT 9                            005A
:AN  F    1.1
:L  KT 440.0
:SP T    9
:AN  I    0.0
:R  T    9
:NOP 0
:NOP 0
:A  T    9
:=  Q    1.2
:***

SEGMENT 10                           0065
:A  Q    1.2
:=  Q    3.0
:***

SEGMENT 11                           0068
:A  Q    1.2
:L  KT 220.0
:SD T    10
:AN  I    0.0
:R  T    10
:NOP 0
:NOP 0
:A  T    10
:=  Q    3.2
:***

SEGMENT 12                           0073
:A  Q    3.2
:=  Q    3.1
:BE

```

## Retirador

La parte final de la planta retirará la botella cuando se presente una botella en el detector de botellas de la sección y el arranque maestro se encuentre activado.

Cuando se ha detectado la botella, el PLC deberá generar un retardo para poder activar el bit de arranque del motor.

Su diagrama de flujo será el siguiente:

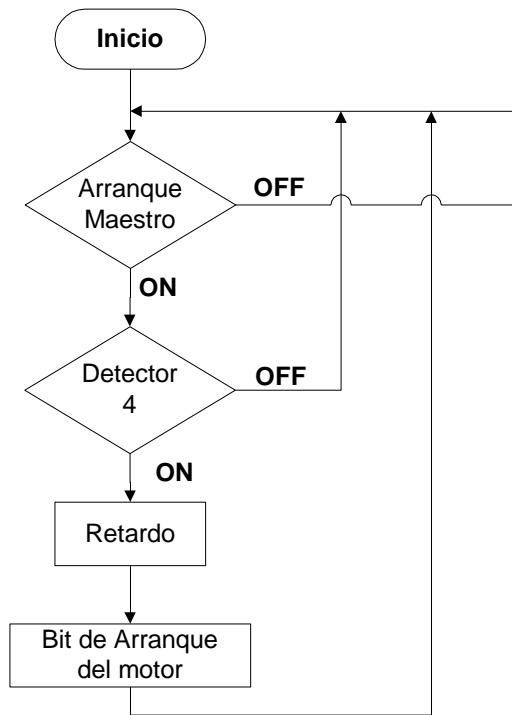


Figura 4-8 Diagrama de flujo del desviador

En este caso el diagrama de tiempos se muestra a continuación:

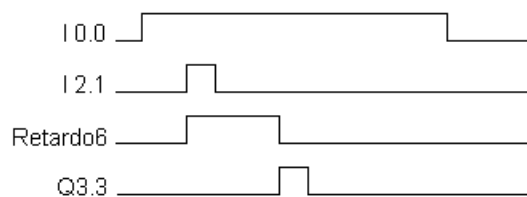
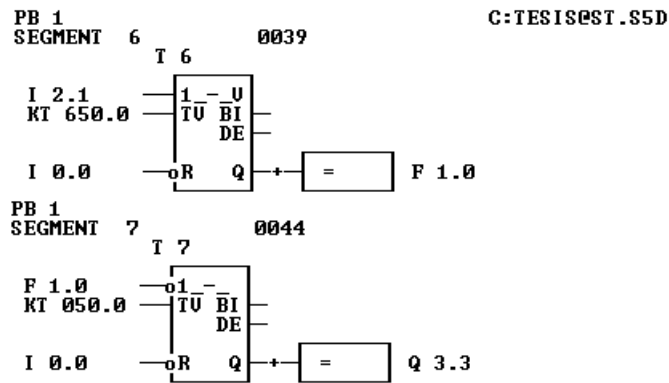


Figura 4-9 Diagrama de los bits de control para el desviador de botellas.

En esta sección de la planta se detecta la presencia de la botella en su cercanía mediante el detector 4 (I 2.1), lo cual genera el retardo 6 necesario para que la botella este en posición y se active el bit de arranque del motor (Q3.3).

La visualización de esta sección del programa se muestra a continuación, el retardo que se necesita es de 6.5 segundos y lo proporciona T6; cabe destacar que este retardo es de acuerdo a la velocidad de la banda, por lo cual es propio del sistema y no es recomendable variarlo.



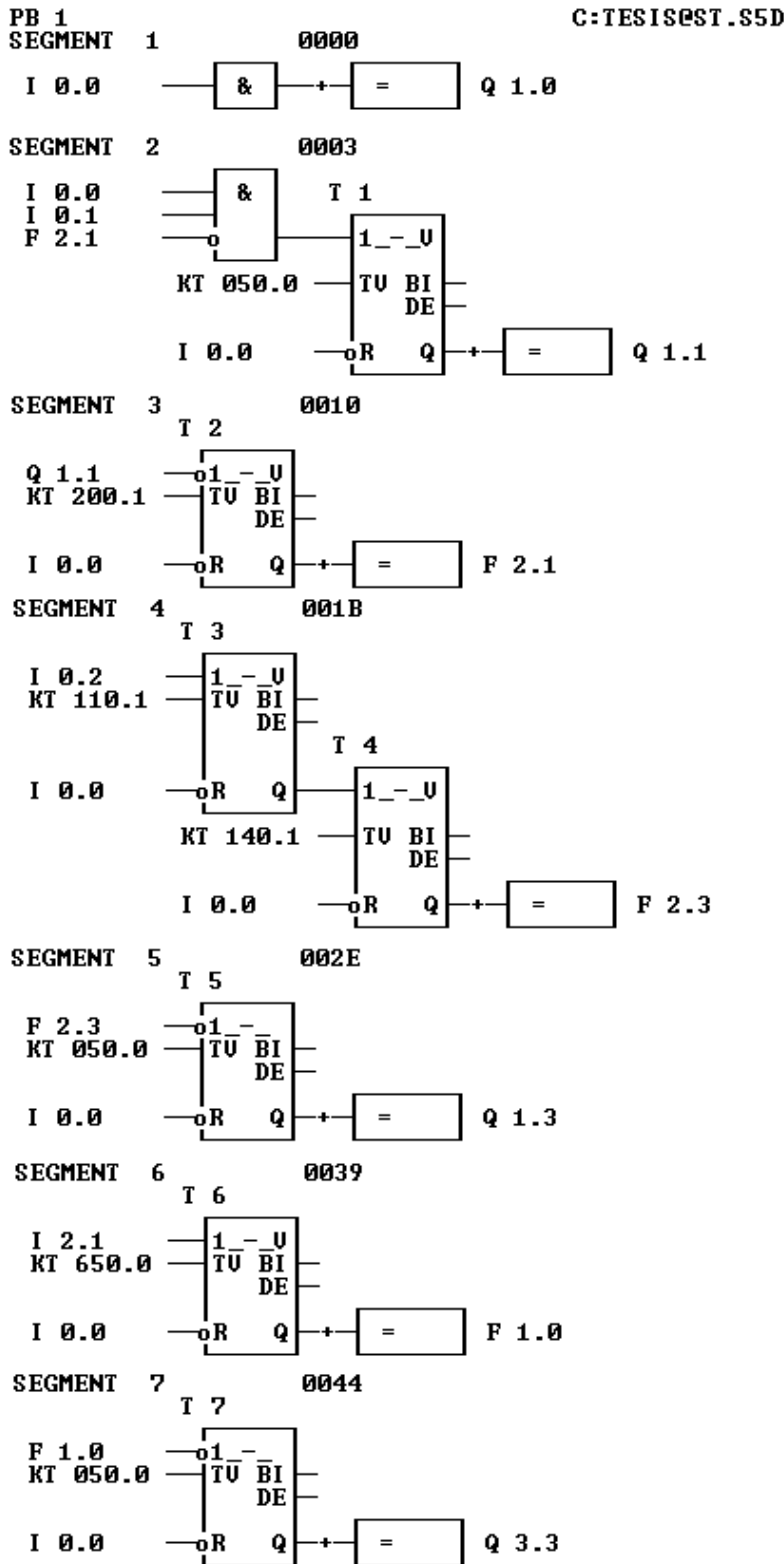
Finalmente el mismo programa en versión STL se muestra a continuación:

```

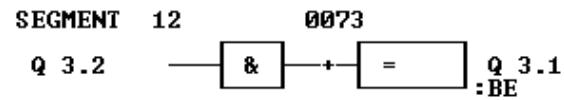
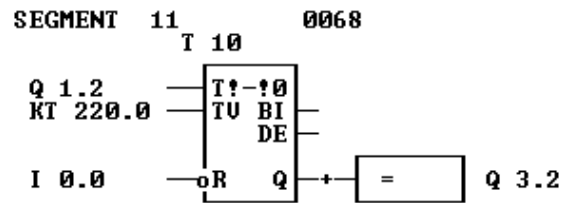
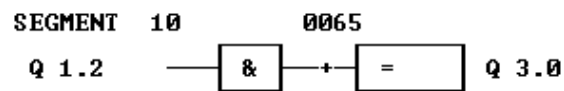
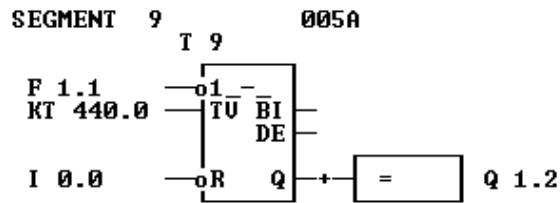
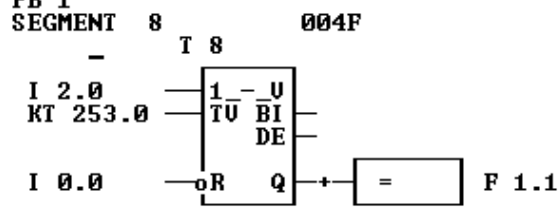
PB 1
SEGMENT 6          0039
: A I 2.1
: L KT 650.0
: SE T 6
: AN I 0.0
: R T 6
: NOP 0
: NOP 0
: A T 6
: = F 1.0
: ***

SEGMENT 7          0044
: AN F 1.0
: L KT 050.0
: SP T 7
: AN I 0.0
: R T 7
: NOP 0
: NOP 0
: A T 7
: = Q 3.3
: ***
    
```

Finalmente el programa completo en modo CSF se muestra a continuación:



PB 1 C:TESIS@ST.S5D



El programa completo en modo STL se muestra a continuación:

```

PB 1                                C:TESIS0ST.S5D
SEGMENT 1                          0000
:A  I  0.0
:=  Q  1.0
:***
SEGMENT 2                          0003
:A  I  0.0
:A  I  0.1
:AN F  2.1
:L  KT 050.0
:SE T  1
:AN I  0.0
:R  T  1
:NOP 0
:NOP 0
:A  T  1
:=  Q  1.1
:***
SEGMENT 3                          0010
:AN Q  1.1
:L  KT 200.1
:SE T  2
:AN I  0.0
:R  T  2
:NOP 0
:NOP 0
:A  T  2
:=  F  2.1
:***
SEGMENT 4                          001B
:A  I  0.2
:L  KT 110.1
:SE T  3
:AN I  0.0
:R  T  3
:NOP 0
:NOP 0
:A  T  3
:L  KT 140.1
:SE T  4
:AN I  0.0
:R  T  4
:NOP 0
:NOP 0
:A  T  4
:=  F  2.3
:***
SEGMENT 5                          002E
:AN F  2.3
:L  KT 050.0
:SP T  5
:AN I  0.0
:R  T  5
:NOP 0
:NOP 0
:A  T  5
:=  Q  1.3
:***

```

```
SEGMENT 6          0039
:A  I    2.1
:L  KT 650.0
:SE T    6
:AN I    0.0
:R  T    6
:NOP 0
:NOP 0
:A  T    6
:=  F    1.0
:***
SEGMENT 7          0044
:AN F    1.0
:L  KT 050.0
:SP T    7
:AN I    0.0
:R  T    7
:NOP 0
:NOP 0
:A  T    7
:=  Q    3.3
:***
SEGMENT 8          004F
:A  I    2.0
:L  KT 253.0
:SE T    8
:AN I    0.0
:R  T    8
:NOP 0
:NOP 0
:A  T    8
:=  F    1.1
:***
SEGMENT 9          005A
:AN F    1.1
:L  KT 440.0
:SP T    9
:AN I    0.0
:R  T    9
:NOP 0
:NOP 0
:A  T    9
:=  Q    1.2
:***
SEGMENT 10         0065
:A  Q    1.2
:=  Q    3.0
:***
SEGMENT 11         0068
:A  Q    1.2
:L  KT 220.0
:SD T    10
:AN I    0.0
:R  T    10
:NOP 0
:NOP 0
:A  T    10
:=  Q    3.2
:***
SEGMENT 12         0073
:A  Q    3.2
:=  Q    3.1
:BE
```

## Capítulo 5 Conclusiones

### Evaluación de la construcción del prototipo.

Todos los componentes descritos en el capítulo 3 forman parte del prototipo del proceso industrial, los cuales interactúan para poder obtener un funcionamiento deseado.

Dada la complejidad de algunos componentes especialmente los arreglos mecánicos, se invirtió mucho tiempo para construir mecanismos funcionales, en algunos casos se tuvo que cambiar mecanismos que presentaron problemas cuando fueron montados para sus pruebas individuales e incluso después cuando la falla resultó de un accidente o de una falla del material.

Una gran ventaja para el manejo de los motores de paso y también de la automatización de cada sección del proceso, fue el uso de las GAL, que representan una buena alternativa sobre otros dispositivos programables (microprocesadores, microcontroladores y FPGA), además de tener un precio muy bajo. Las desventajas de estos dispositivos son su gran debilidad ante cargas electroestáticas y la poca disponibilidad de los software para su programación como lo es el OPAL y el OPAL jr.

Con el fin de evitar que las GAL estuvieran expuestas a una descarga electrostática cuando el sistema esté en funcionamiento, se le agregó al diseño final un relevador de aislamiento a la entrada de los módulos de control. Con esto, los módulos de control están aislados tanto a la entrada como a la salida.

La respuesta de los motores a las órdenes de mando (paro y cambio de dirección), resultan muy buenas, además con la integración de circuitos VCO dentro de los módulos de control, es muy fácil sincronizar a los motores que interactúan en la banda como por ejemplo el motor de la banda y el motor para el desplazamiento de la sección de llenado, simplemente girando un potenciómetro.

El problema más drástico que se manifestó en el uso de motores de pasos, fue una vibración generada por el tipo de movimiento del rotor del motor de pasos, que se vio reflejada sobre la banda, la cual a su vez hizo que la trayectoria de las botellas variara mucho dado su bajo peso y la poca fricción que hay entre la botella y el material de la banda. Para aminorar este problema se le agregó a la cadena de transmisión de la banda un amortiguador de vibraciones, el cual disminuyó las vibraciones, sin embargo no las eliminó por completo.

Adicionalmente se le agregaron guías de trayectoria para las botellas, las cuales consistían en pedazos de plástico dispuestos en las orillas y que dirigen a las botellas en la ruta deseada. Esto ayudo mucho en la trayectoria de las botellas, sin embargo, se presentaron problemas de tracción originados por la vibración del motor de la banda. Una opción idónea era el cambio del material con el cual se construyó la banda por otro de mayor tracción. Dada la construcción del prototipo, resultó imposible cambiar la banda por otra de un material con mayor tracción sin desmontar otras tres secciones del proceso, una solución alternativa fue el de agregar palillos de tracción los cuales arrastran las botellas aun con la vibración.

El control maestro de arranque y paro, resultó ser una parte fundamental de todo el proceso, ya que con éste, todo el sistema se controla, ya sea en el arranque o en un paro de emergencia o de fin de proceso.

En el caso del PLC, se demostró su gran versatilidad, cuando se requiere afinar detalles para el control del proceso. Un caso muy específico fue el ajuste de los retardos para la operación de los motores de cada sección, el tiempo en un principio fue cronometrado con un reloj digital, con el fin de obtener el valor del retardo. Cuando se empleo el PLC para verificar el tiempo obtenido, se encontraron errores de medición con el reloj, sin embargo, dado que el PLC puede manejar hasta centésimas de segundo se llegó a los tiempos exactos de retardo para cada sección. Además estas pruebas se realizaban solamente con la sección de interés, sin la necesidad de poner todo el sistema en operación, así era posible realizar las pruebas independientemente e instantes después se podía hacer una prueba con varias secciones a la vez con solo cambiar el programa del PLC, para esto cada sección ya estaba cableada a una salida del PLC, por lo cual no fue necesario recablear para realizar pruebas completas, solamente se reasignaron las salidas en el programa del PLC.

Finalmente podemos afirmar que si fue posible controlar un proceso industrial basado en una banda de transporte, con cinco secciones:



1. Banda transportadora
2. Despachador de botellas
3. Desviador de botellas
4. Llenado de botellas:
  - a. Motor para el desplazamiento de toda la sección
  - b. Motor para el movimiento de la boquilla de llenado.
  - c. Control para la bomba y la electro válvula.
5. Retirador de botellas.

Sin embargo se le pueden agregar más secciones a este proceso industrial para completar el mismo. Otra mejora es la inclusión de módulos manejadores de motores de pasos, los cuales no estaban disponibles para la realización de este proyecto y que fueron los que realmente nos llevó al diseño y desarrollo de los módulos de control que incluían GAL.

La implementación de este tipo de prototipos puede ser una buena opción para su utilización en laboratorios de control o en asignaturas afines, dado que un prototipo importado resulta mucho más caro, aun cuando se construyera uno similar en México con los mismos materiales, el precio resulta mucho más barato.

## Conclusiones

Como se observó a través del desarrollo de esta tesis el PLC es una herramienta muy poderosa y flexible, capaz de solucionar el control de cualquier proceso de tipo industrial. Esta flexibilidad permite también corregir algunos detalles dentro del mismo proceso sin la necesidad de desconectar algún componente, todos los ajustes son vía computadora o vía programador, con lo cual simplemente configurando correctamente los controladores a través de la pantalla y cargándolo en el CPU del PLC, se realiza el ajuste. Esto nos permite que el proceso se pause en un periodo corto e inclusive si el detalle a corregir es mínimo, el proceso puede continuar mientras se ajustan los errores.

La representación gráfica del lenguaje de programación del PLC resulta amigable, dado que con un poco de práctica es fácil comenzar a entender y aplicar los comandos del PLC como lo son compuertas lógicas, temporizadores, comparadores y otras funciones. Esto añade un factor más a la facilidad de uso de los PLC.

Dada la amplia difusión del PLC en las industrias, es muy recomendable la difusión del mismo en la formación de los ingenieros, puesto que los sistemas de producción tienden a automatizarse y lo cual implica el uso de uno o varios PLC, y dadas las necesidades de producción y competitividad actual entre las empresas, se debe tener el conocimiento para el manejo del PLC.

Otro punto de interés que cabe recalcar fue el desarrollo del prototipo, el cual se realizó desde cero y que involucró tanto el desarrollo mecánico como el desarrollo electrónico del mismo, se tuvo que determinar cuales eran los componentes adecuados, se planeó cada parte y se tuvieron que hacer mejoras e incluso rediseñar partes que no cumplían con el funcionamiento deseado del prototipo, en si el desarrollo del prototipo tomó alrededor de 1500 horas.

El uso de GAL es cuestionable dado que lo deseado es el uso de microprocesadores, sin embargo, cada módulo de control del movimiento de motores, en sí realiza tareas simples y repetitivas, las cuales se les puede asignar a circuitos programables más sencillos como las GAL, que a su vez resulta en un beneficio económico sobre los microprocesadores.

El desarrollo del prototipo vino a complementar al PLC, de tal forma que podemos observar de manera física como se controla un proceso real, que aunque en miniatura, resulta una mejor opción que si se tratara de analizar el mismo proceso mediante un tablero.

## APENDICE

**El lenguaje de representación de un programa en STEP5, tiene tres posibles formas de representación:**

**CSF (Control System Flowchart).** La representación CSF es una manera de programación mediante bloques de funciones lógicas (AND, OR, etc).

**LAD (Ladder Diagram).** LAD es la forma de representar un programa mediante símbolos eléctricos (bobinas, Contactos, etc.), es decir, mediante diagramas en escalera.

**STL (Statement List).** El tipo de representación STL es mediante lista de instrucciones, es decir, mnemónicos que indican el tipo de operaciones que se desea realizar. Este tipo de programación es similar al lenguaje ensamblador utilizado en programación de microprocesadores.

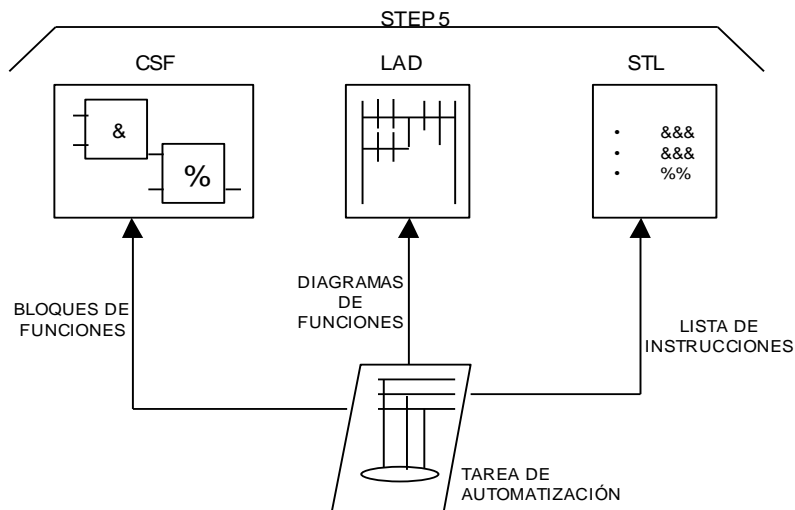


Fig.1 Formas de representación en STEP 5

### Instrucciones básicas en STEP5

Una instrucción en STEP 5 se compone de una parte operacional, y un operando. Los operandos son utilizados en los tres tipos de representación e indican con que parámetros se va a ejecutar la operación.

- I1.1 Señal de entrada tipo bit localizada en el byte 1, bit 1 del mapa de memoria.
- FW3 Bandera tipo palabra localizada en el byte 3 y 4 del mapa de memoria.
- Q2.3 Señal de salida tipo bit localizada en el byte 2, bit 3 del mapa de memoria.
- C4 Contador # 4.
- T5 Temporizador # 5.
- PB 20 Bloque de programa # 20

Un operando queda identificado por dos informaciones, la etiqueta del tipo de operando o tipo de señal y su dirección.

- I Entrada.
- Q Salida.
- F Bandera.
- D Dato.
- T Temporizador.
- C Contador.
- P Periferia (tarjetas analógicas).
- K constante.
- OB, PB, etc. Bloques.

La parte operacional de una instrucción es utilizada en conjunto con los operandos cuando se programa en representación STL; y es la de que describe el trabajo o función a realizar.

- A And.
- O Or.
- = Asignar un resultado.
- C Llamar un bloque de datos.
- JU Saltar incondicionalmente a un bloque determinado

## Programador

El programador de PLC's disponible consiste de una microcomputadora, la cual tiene cargado el lenguaje de programación STEP 5, y cuenta con la interfase necesaria para tener comunicación con el PLC.

El teclado de la computadora será la herramienta mediante la cual se accesarán los comandos de programación; dicho teclado presenta una correspondencia teclado-función.

## Acceso a STEP 5

Dentro del sistema operativo de la microcomputadora, se escribe el comando S5 para acceder a STEP 5. a continuación aparecerá la primera pantalla de trabajo, la cual recibe el nombre de COMÍ, y es donde se presentan todas las operaciones con que cuenta el programador. La selección de una tarea determinada se realiza posicionando el cursor delante de la opción deseada y presionando la tecla funcional (PACKAGE).

SELECT	PACK	AGE	SIMATIC				S5/KOMI
LAD,	CSF,	STL .....	V3.0	DS5PXS01X.	CMD		
			TTY /	AS 511 -	INTERFACE	(STANDARD)	
F1	F2	F3	F4	F5	F6	F7	F8
PACKAGE	UTILITY	INFO	VERSION	INTERFACE	DRIVE	NEW SEL	RETURN

Fig. 2 Pantalla de selección de opciones (KOMI)

Todo el software S5 que se utilice a partir de este punto se caracteriza por el uso de pantallas con una misma estructura. En la parte superior de la pantalla se indica en todo momento la función que se está realizando y la información, si es necesario, del archivo correspondiente. En la parte inferior se encuentra la región de diálogo con el usuario, compuesta básicamente de dos subregiones: el menú con la correspondencia de función-tecla de función; y la línea de avisos y mensajes de error. En la parte central de la pantalla se desarrolla propiamente la función de programación.

FUNCION				ARCHIVO			
F1	F2	F3	F4	F5	F6	F7	F8
		M	E	N	U		

**Fig. 3 Estructura de las pantallas de STEP5**

Para desarrollar los programas de aplicación S5 en cualquiera de sus tres formas de representación se selecciona la opción LAD, CSF, STL que aparece en la pantalla KOMI.

A continuación aparece una pantalla denominada máscara de ajustes previos (PRESETS). En ella se fijan las condiciones de trabajo: nombre del archivo, tipo de representación, modo de operación, etc.

PRESETS				SIMATIC		S5/PES01	
REPRESENT.	:	(LAD, CSF, STL)	PROGRAM	FILE :	D:@@:@@ST .S5D [RW]		
SYMBOLS	:	(NO, YES)	SYMBOLS	FILE :			
COMMENTS	:	(YES, NO)	FOOTER	FILE :			
FOOTER	:	(NO, YES)	PRINTER	FILE :			
CHECKSUM	:	(NO, YES)					
MODE	:	(OFF, ON)					
PATH NAME	:		PATH	FILE:			
F1	F2	F3	F4	F5	F6	F7	F8
		SELECT			ENTER	INFO	

**Fig. 4 Mascara de ajustes previos (PRESETS)**

PRESETS				SIMATIC		S5/PES01	
REPRESENT	:	(LAD, CSF, STL)	PROGRAM	FILE :	D:@@:@@ST .S5D [RW]		
SYMBOLS	:	(NO, YES)	SYMBOLS	FILE :			
COMMENTS	:	(YES, NO)	FOOTER	FILE :			
FOOTER	:	(NO, YES)	PRINTER	FILE :			
CHECKSUM	:	(NO, YES)					
MODE	:	(OFF, ON)					
PATH NAME	:		PATH	FILE:			
F1	F2	F3	F4	F5	F6	F7	F8
INPUT	OUTPUT	TEST	PC FCT	PC INFO	PRESETS	AUX FCT	RETURN

**Fig. 5 Máscara de Selección de Funciones.**

El nombre del archivo debe tener un máximo de 6 caracteres. El tipo de representación se selecciona presionando la tecla F3.

El modo de operación tiene dos opciones ON y OFF. On indica que la comunicación programador-PLC está activada y OFF que se encuentra desactivada. Al igual que para el tipo de representación, el modo de operación se selecciona con F3.

Una vez hechos y aceptados los ajustes de la máscara PRESETS aparece en la pantalla el menú principal LAD, CSF, STL. En este punto se puede ejecutar cualquier función disponible en el menú. Por ejemplo: F1 INPUT para empezar a crear un bloque; F2 OUTPUT para acceder a un bloque anteriormente realizado; F3 TEST para probar el funcionamiento de bloques; F8 RETURN para regresar a la pantalla KOMÍ.

### Edición de un programa nuevo: modo INPUT

Para introducir un programa por primera vez se seleccionara en el menú principal la opción F1 INPUT. Esta función tiene a su vez un submenú con las operaciones F1 BLOCK y F4 MASK. Con F1 se pasa a una pantalla donde se selecciona el dispositivo de entrada (PC = PLC, PG = programador o FD = disco duro o flexible), y el tipo y número de bloque a editar (OB, PB, etc).

INPUT				SIMATIC				S5 / PES01	
F1	F2	F3	F4	F5	F6	F7	F8		
BLOCK			SCR FORM						RETURN

(A)

INPUT DEVICE		BLOCK:	
--------------	--	--------	--

(B)

Fig. 6 Acceso a modo INPUT (F1)

### Edición en representación CSF

Para efectos de edición CSF, la pantalla está dividida en 8 niveles horizontales, donde puede situarse un símbolo funcional o un operando. Verticalmente no hay límites de niveles.

Los símbolos básicos que se utilizan son: compuertas AND y OR, complementados con Flip-Flop, Timers, Contadores.

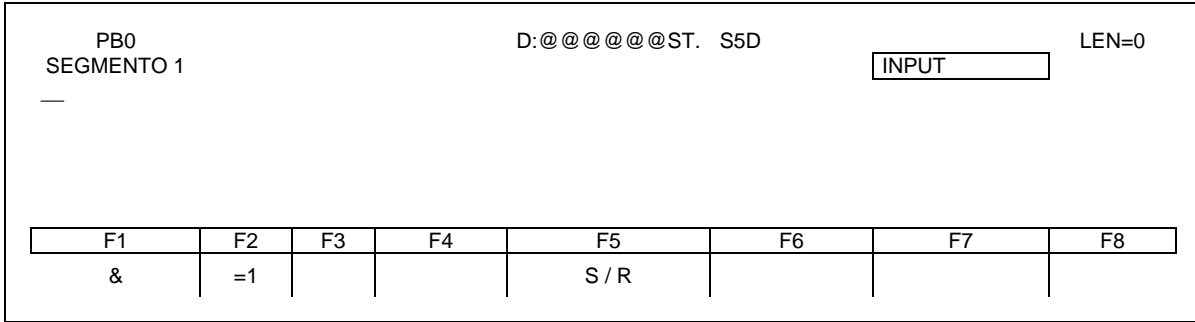


Fig. 7 Pantalla de programación en CSF.

### Edición en representación LAD

La pantalla en una representación LAD se encuentra dividida de la misma manera que para CSF. Los elementos eléctricos característicos de la representación LAD son contactos y bobinas. Los tipos de contadores con los que se cuenta son dos: normalmente cerrados y normalmente abiertos.

De la interconexión de contactos y bobinas se pueden realizar diferentes tipos de operaciones lógicas, como AND's y OR's, ya que los demás símbolos son similares, tanto en edición CSF como en LAD.

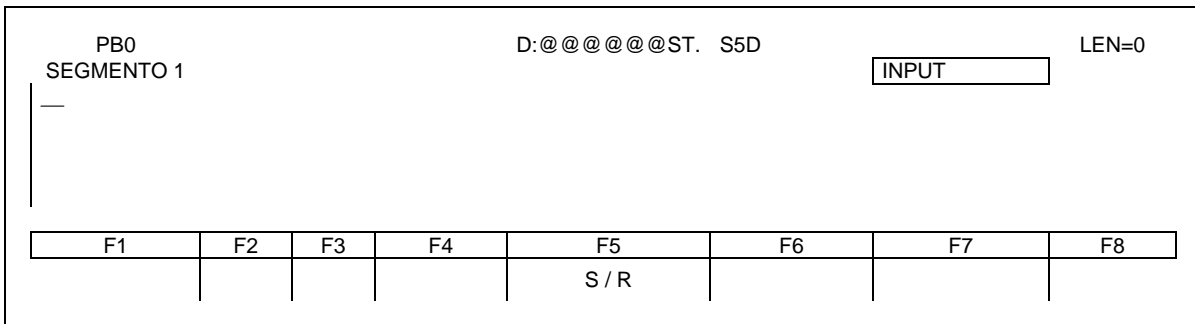


Fig. 8 Pantalla de Programación en LAND

### Edición en representación STL

Cuando se trabaja en representación STL, en lugar de utilizar gráficos se utilizan listas de instrucciones, esto es, mnemónicos similares a los que se utilizan en lenguaje ensamblador.

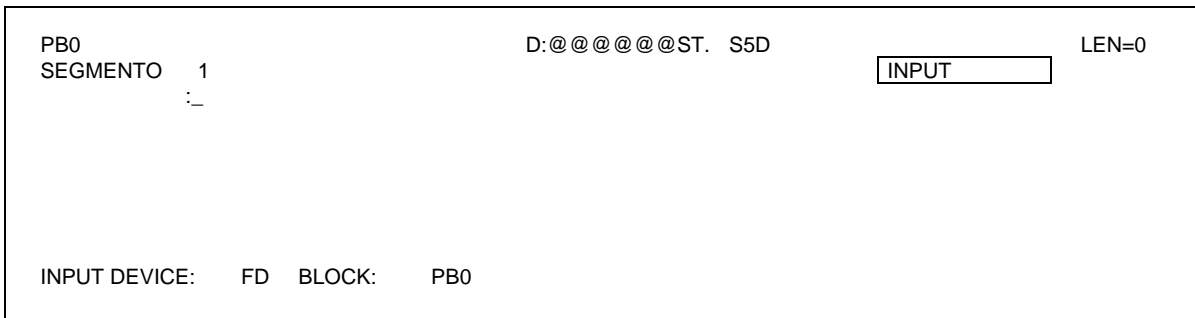


Fig. 9 Pantalla de programación en STL.

## Listar programa ya creado: modo OUTPUT

Para listar un programa ya creado se selección desde el menú principal la función F2 OUTPUT. Esta función tiene a su vez un submenú con las opciones F2 BLOCK y F4 MASK. Con F2 se pasa a una pantalla donde se selecciona el dispositivo donde se encuentra almacenado el programa, el tipo y número de bloque, y el número de segmento a visualizar.

OUTPUT				SIMATIC				S5 / PES01	
F1	F2	F3	F4	F5	F6	F7	F8		
BLOCK			SCR FORM				RETURN		

(A)

OUTPUT	DEVICE:	BLOCK:	SEARCH:	PTR:
--------	---------	--------	---------	------

(B)

Fig. 10 Acceso a modo OUTPUT (F2)

## Inserción y borrado de segmentos: modo INSERT y DELETE

Para insertar segmentos a un programa ya creado se selecciona desde el menú principal el modo OUTPUT. Se posiciona el cursor donde se desea realizar la inserción y se oprime SHIFT F10. En este momento se borra la pantalla y se puede editar el segmento deseado. Una vez editado el segmento se oprime la tecla de aceptación total. El sistema se encarga de hacer un corrimiento en numeración a los segmentos bajo el segmento insertado.

Para borrar segmentos de un programa ya creado se selecciona desde el menú principal el modo OUTPUT. Se posiciona el cursor al principio del segmento que se desea borrar y se oprime la tecla X, y el sistema pregunta si en realidad se desea borrar. Para aceptar el borrado del segmento se oprime la tecla de aceptación total.

## Corrección de un segmento: modo CORRECT

Para corregir segmentos de un programa ya creado se selecciona desde el menú principal el modo OUTPUT. Se posiciona el cursor en el segmento que se desea corregir y se presiona la tecla CORR. Se realiza la corrección del segmento. La corrección total debe aceptarse con la tecla de aceptación total.

### Transferencia de un programa: modo TRANSFER

Para transferir un programa residente en disco duro o en disco flexible al PLC se selecciona desde el menú principal la función F7 AUX FCT (Funciones Auxiliares), el cual a su vez tiene un submenú conformado por la siguientes opciones: F1 TRANSFER, F2 DELETE, F3 DIR, F6 PRG FIL y F8 RETURN. Con F1 se pasa a una pantalla donde se piden fuente del programa, bloque a transferir, destino de transferencia y nombre del bloque en el destino. La transferencia es realizada cuando se escribe toda la información que el programador requiere y se presiona la tecla de aceptación total. La fig. muestra los pasos a seguir (por medio de pantallas) para realizar este procedimiento.

AUXILIAR    Y    FUNCTIONS				SIMATIC    S5 / PES01			
F1	F2	F3	F4	F5	F6	F7	F8
TRANSFER	DELETE	DIR			PRG FILE		RETURN

(A)

TRANS	SOURCE:	BLOCK:	TO DEST:	BLOCK:
-------	---------	--------	----------	--------

(B)

Fig. 11 Mascara de Transferencia de un Programa

### Prueba de u programa: modo TEST/STATUS.

El programador tiene una función que permite verificar el funcionamiento de un programa residente en el PLC. Esta función consiste en verificar en tiempo real el estado lógico de las diferentes señales involucradas en los bloques que conforman el programa.

Para entrar a dicha función es necesario localizarse en el menú principal y oprimir la función F3 TEST, la cual tiene un submenú con las opciones: F1 PRO CTRL., F2 PRO CTRL., F3 STATUS y F8 RETURN. Con F3 se pasa a una pantalla que pide la información del bloque y el segmento a analizar. Si no se da el número de segmento se empezará a partir del número 1. Con aceptación total se despliega el segmento del bloque solicitado. Dentro de esta función se tiene la posibilidad de realizar correcciones.



TEST		FUNCTIONS				SIMATIC		S5 / PES01	
F1	F2	F3	F4	F5	F6	F7	F8		
PROG TEST	END TEST	STATUS						RETURN	

(A)

TRANS	SOURCE:	BLOCK:	TO DEST:	BLOCK:
-------	---------	--------	----------	--------

(B)

Fig. 12 Función TEST/STATUS (F3)

## Tipo de operaciones

Existen tres tipos de operaciones en STEP5:

- Operaciones complementarias.
- Operaciones sistema.
- Operaciones básicas.

## Operaciones Complementarias

Las operaciones complementarias comprenden funciones complejas tales como instrucciones de sustitución, funciones de prueba de bit, operaciones de desplazamiento y transformación, las cuales sólo pueden programarse en STL.

## Operaciones de Sistema

Las operaciones de sistema accesan directamente al sistema operativo, y también sólo son programables en STL.

## Operaciones Básicas

Las operaciones básicas comprenden funciones ejecutables en los diferentes tipos de bloques. Pueden ser programadas en los tres tipos de representación.

Las operaciones básicas de las tres representaciones son: AND, OR, funciones de memoria R-S, temporizadores, contadores y comparadores.

## Bibliografía

1. *Simantic S5-100U System Manual*, Siemens Aktiengesellschaft, Berlin/Munich, Alemania, 1989.
2. Hans Berger, *Automating With The Simantic S5-115U*, Siemens Aktiengesellschaft, Berlin/Munich, Alemania, 1992.
3. Timothy J. Maloney, *Industrial Solid State Electronics, Devices and Systems*, Prentice Hall Inc., New Jersey, Estados Unidos, 1986.
4. Francisco J. Rodríguez Ramírez, Ricardo Garibay Jiménez, *Introducción a los Controladores Lógicos Programables*, Sociedad de Instrumentistas de México, México, 1988.
5. John W. Webb, *Programmable Logic Controllers*, Dave Inc. Estados Unidos, 1989.
6. *Manual STEP 5 Básico*, Centro de Información y Entrenamiento para Automatización, Siemens. México, 1995.
7. Simatic S5/PC/505 Automation Systems, Siemens. Alemania, 2000.
8. STEP 5: Programming Package for Personal Computers, Siemens, Alemania, 1995.
9. A. E. Fitzgerald, Charles Kingsley Jr., Stephen D. Umans, *Máquinas Eléctricas*, McGraw Hill, México 1992.
10. *Stepping Motors Operation & Theory*, SKC Inc., Japón, 2000.
11. Fredrick J. Hill, Gerald R. Peterson, *Introduction to Switching Theory & Logical Design*, John Wiley & Sons, Estados Unidos, 1981.
12. Ronald J. Tocci, *Sistemas Digitales Principios y Aplicaciones*, Prentice Hall Hispanoamericana S. A., México 1987.
13. James D. Broesch, *Practical Programmable Circuits*, Academic Press Inc., California, Estados Unidos, 1991.
14. *GAL Data Book*, Lattice Semiconductor Corp., Oregon, Estados Unidos, 1992.
15. *Opal Junior User's Guide/Manual*, National Semiconductor Corp., California, Estados Unidos, 1991.
16. *National Application Specific Analog Products Databook*, National Semiconductor Corp., California, Estados Unidos, 1995.
17. *Linear Applications Handbook*, National Semiconductor Corp., California, Estados Unidos, 1994.
18. *Optoelectronics Device Data*, Motorola Inc., Estados Unidos, 1989.