

5  
24



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

AUTOMATIZACION UTILIZANDO PLC

**T E S I S**

QUE PARA OBTENER EL TITULO DE

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A N :

NORMA ALEXANDRA ALMANZA RUIZ

EDGARDO URMARTIN GUZMAN

DIRECTOR: ING. RICARDO GARIBAY JIMENEZ



MEXICO, D. F.

1996

**TESIS CON  
FALLA DE ORIGEN**

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Tesis de Licenciatura

Título: *Automatización Utilizando PLC*

Integrantes:

Norma A. Almanza Ruiz

Edgardo U. Martín Guzmán

Director: Ing. Ricardo Garibay Jimenez

Fecha: 16 de Septiembre de 1996

Tesis de Licenciatura

Título: *Automatización Utilizando PLC*

Integrantes:

Norma A. Almanza Ruiz

Edgardo U. Martín Guzmán

Director: Ing. Ricardo Garibay Jimenez

Fecha: 16 de Septiembre de 1996

Revisión: 6

Rel. anterior: 8 junio 1996 rev. 5

## Índice

### **Introducción: Controladores Lógicos Programables**

- i Alcances de la presente Tesis
- ii Definición de PLC
- iii Antecedentes
- iii Características Generales de los PLC

### **Capítulo I: Arquitectura del sistema Simatic S5-100U**

- I.1 Descripción del sistema Simatic S5-100U.
  - I.1.1 Características Generales
  - I.1.2 Bloques Funcionales
- I.2 Sistema de entradas y salidas.
  - I.2.1 Módulos de entrada y salida digital
  - I.2.2 Módulos de entrada y salida analógica
  - I.2.3 Módulos Especiales
- I.3 El bus Simatic.
  - I.3.1 Construcción del bus en el S5-100U
  - I.3.2 Concepto de Dirección
- I.4 Aparatos de programación.
  - I.4.1 Descripción del PG 710
  - I.4.2 Conexiones del PG 710

### **Capítulo II: Arquitectura en Software de un sistema Simatic**

- II.1 El lenguaje de programación STEP 5
  - II.1.1 Tipos de representación
  - II.1.2 Definiciones
- II.2 Operaciones Básicas
  - II.2.1 Combinaciones
  - II.2.2 Operaciones de memoria
  - II.2.3 Operaciones de temporizado
  - II.2.4 Operaciones con contadores
- II.3 Operaciones digitales con palabras
  - II.3.1 Operaciones de carga y transferencia
  - II.3.2 Combinaciones Digitales
  - II.3.3 Operaciones aritméticas

- II.3.4 Operaciones de comparación
- II.3.5 Operaciones complementarias
  
- II.4 Organización de un programa en STEP 5
  - II.4.1 Tipos de programación
  - II.4.2 Tipos de bloques de programa disponibles
  - II.4.3 Módulos funcionales integrados
  
- II.5 Documentación de programas

### **Capítulo III: Aplicaciones con Simatic S5-100U, CPU-102**

- III.1 Tablero de Simulación.
  - III.1.1 Construcción y módulos originales
  - III.1.2 Módulos Sustitutos
  
- III.2 Programas de aplicación de lógica combinatorial
  - III.2.1 Generador de Pulsos
  - III.2.2 Control de Crucero
  
- III.3 Control Secuencial
  - III.3.1 Lavado de automóviles
  
- III.4 Manipulación de datos numéricos
  - III.4.1 Elevador

### **Conclusiones**

### **Apéndice**

Primera Sesión con un Simatic S5

### **Bibliografía**

# INTRODUCCIÓN





## INTRODUCCIÓN

En el ámbito académico y también en el ámbito industrial hay un interés general por conocer los controladores programables (PLC) y su papel dentro del campo de aplicación del control. Simultáneamente, en los laboratorios del Departamento de Control hay disponibles herramientas de este tipo, particularmente de la marca Siemens, el modelo Simatic S5-100 U.

En vista de las necesidades del Laboratorio de Control, se provee a través de este texto una referencia bibliográfica precisa acorde al equipo disponible en el Laboratorio. Este trabajo puede tomarse como base para el desarrollo de proyectos de aplicación más elaborados. Consideramos que también puede servir para brindar capacitación a personal externo a la UNAM, es decir que su campo de aplicación no se limitará a lo académico, sino que se extiende a lo industrial.

Cabe mencionar que el tratamiento que se le da a este trabajo, es el de que sirva como base para que un usuario que lo consulte pueda generar un sistema completo basado en la tecnología de PLC. Es decir, no se deja como una aplicación definida y cerrada, sino que permite adaptar y definir el hardware necesario y aplicar la metodología para el desarrollo del software asociado a la solución de un problema particular.

El trabajo de tesis que se presenta está estructurado de la siguiente manera:

El Capítulo I consiste en una descripción de los elementos de Hardware disponibles en el Laboratorio de Control Digital de la Facultad de Ingeniería.

El Capítulo II es una introducción al lenguaje de programación STEP 5, con indicaciones de la utilización de comandos, a través de breves notas de aplicación.

El Capítulo III plantea la metodología de trabajo a través de la implementación de pequeños proyectos de diverso grado de dificultad, programados en STEP 5 STL, y con documentación de las variables involucradas y la configuración de los módulos de entrada y salida utilizados en el PLC, cuando estos sean requeridos.

## ALCANCES DE LA PRESENTE TESIS

Con la presente tesis se darán los principios básicos de configuración, operación y aplicación de la familia de controles programables Simatic 100 U, esto con el objetivo de que este texto sirva de base para difundir la tecnología de PLC, específicamente del Simatic, a través de la implementación de proyectos y el desarrollo de sistemas de prueba, en el Laboratorio de Control Digital de la Facultad de Ingeniería de la UNAM.

## DEFINICIÓN DE PLC

La definición NEMA establece que las siglas PLC provienen del inglés ***Programable Logic Controller*** el cuál es un sistema electrónico digital basado en un microprocesador, capaz de almacenar instrucciones para implementar funciones de control, tales como secuencias, temporizados, comunicaciones, control PID y otras, con la finalidad de controlar la operación de una máquina o de un proceso en ambientes industriales, es decir, en presencia de ruido eléctrico, humedad, vibraciones y variaciones de temperatura. Debido a su capacidad de almacenar el programa en una memoria de estado sólido del tipo RAM, los PLC hacen posible un sistema de control flexible, reemplazando los sistemas basados en lógica alamburada.

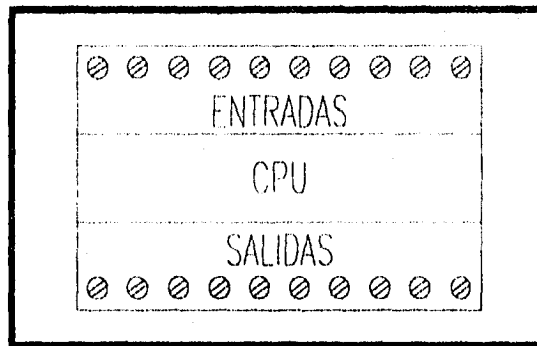


Fig. 1.1

Diagrama de un sistema PLC típico

Actualmente la capacidad de control de un PLC's ha sobrepasado las primeras expectativas de la lógica alamburada por lo cual el término "Lógico" de la definición original ya no se aplica más en los textos para referirse a esta técnica.

## ANTECEDENTES HISTÓRICOS

La tecnología de los controles programables se origina hace más de dos décadas, cuando en 1968 una compañía constructora de autos, buscando reducir el costo de sus sistemas de control en líneas de producción, estableció los criterios de diseño de una nueva forma de control.

Las especificaciones del nuevo sistema requerían de:

- Un sistema de estado sólido
- Basado en microprocesador
- Orientado a ambientes industriales
- Flexible y fácil de programar
- Utilizable por personal técnico de planta

Hoy en día los fabricantes de controles programables son muchos y sus lugares de origen son diversos. No es raro encontrar compañías grandes como la General Electric, que es conocida por los industriales como fabricante de turbinas de gas y alternadores así como por las amas de casa como fabricante de lavadoras y refrigeradores, que haya diversificado sus productos hasta el caso de comercializar también equipos PLC. Otros ejemplos de este fenómeno se encuentran también en Europa y Asia. El efecto de esta diversificación se refleja en el precio, y en la gran cantidad de marcas de equipos para control. Los usuarios finales a la hora de elegir una marca deben considerar no solamente las características intrínsecas del equipo, sino también el prestigio del fabricante.

Por supuesto que es posible diseñar una solución específica para un problema particular sin la necesidad de utilizar un PLC comercial, a una fracción del precio de este último. Sin embargo, no está sólo en juego la factibilidad sino también una serie de factores que involucran tiempo, calidad, presentación, servicio, mantenimiento, refaccionamiento y garantía. Estas son cosas que únicamente se pueden encontrar en compañías establecidas a nivel mundial.

## **CARACTERÍSTICAS DE LOS CONTROLADORES PROGRAMABLES:**

### **1. Independencia del cableado con respecto a la estrategia de control.**

Debido a que las interfaces de entrada y de salida están aisladas galvánicamente unas con respecto de 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.

### **2. 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.

### **3. 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.

### **4. Flexibilidad**

Los sistemas son de tipo modular por lo que cuando es necesario incrementar el número de I/O 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 ó inteligentes.

### **5. Ejecución Cíclica.**

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

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

### **6. 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, tienen como resultado menores tiempos de instalación, de arranque, de paro, de detección de fallas, de mantenimiento, etc.

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

7. 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 con matemática de punto flotante.

8. Comunicaciones.

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

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.

# CAPITULO I

## Elementos de Hardware del Simatic S5 100-U



## **I.1 DESCRIPCIÓN DEL SISTEMA SIMATIC S5-100U**

El Simatic 100U es un Controlador Programable (PLC). Es parte de una familia de controladores programables, siendo el equipo de control 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.

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 interfaces de I/O 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.

### **I.1.1 Características Generales**

Hemos dicho que este equipo es 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 I/O

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

- 1 CPU
- 1 Fuente de Alimentación
- 1 Módulo de interface para expansión en varias filas
- Hasta 16 elementos de bus
- Hasta 32 módulos de I/O

Existen para el modelo Simatic 100 U tres distintas CPU's, 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	si	si
Coprocadores	no	no	si
Memoria de usuario	2 KB	4 KB	20 KB
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	si	si
Reloj Hardware	no	no	si
Control PID	no	no	si
Parametrización de FB	no	no	si

Tabla 1.1  
Comparación entre diferentes CPU's Simatic

Puede observarse que las CPU's más avanzadas poseen una mayor cantidad de elementos direccionables (Temporizadores, Contadores, Banderas, Bloques de programa) así como también funciones especiales como Reloj de tiempo real, Módulos de Funciones parametrizables estándar y de usuario, algoritmo de control PID.



## I.1.2 Bloques Funcionales

Analizaremos primeramente 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 5 diferentes CPU's que son : 90U, 95U, 100U, 102U y 103 U.

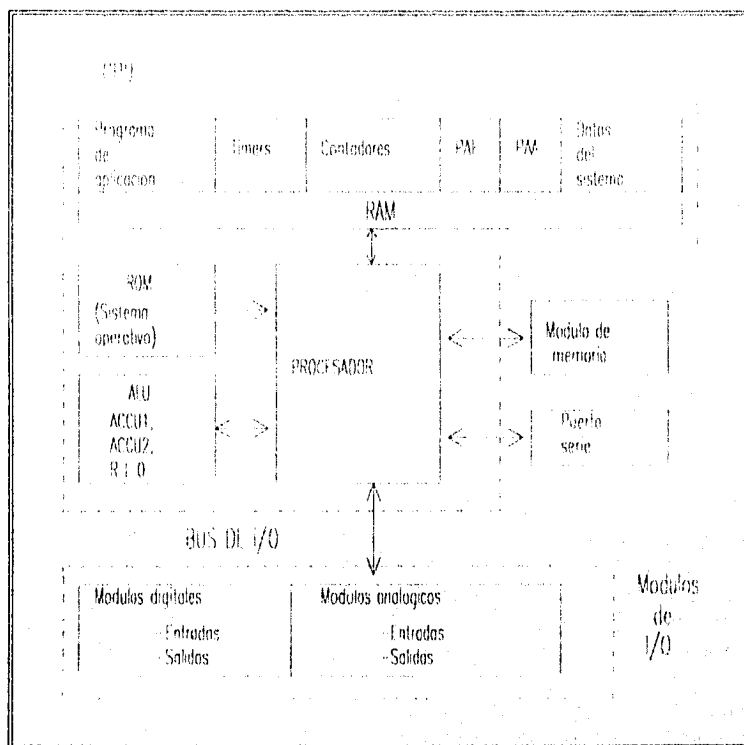


Fig. 1.1

Intercambio de datos entre CPU y 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 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 PAA.

La unidad lógica y aritmética se compone de dos acumuladores de 16 bits, llamados ACCU 1 y ACCU 2. 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, *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 E/Prom 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 de proceso, y los valores de los contadores y temporizadores. A continuación veremos el mapa de memoria para dos diferentes CPU's, de la serie Simatic.

CPU	102	103
Mem. 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
Cont. remanentes	ED00...ED0F	ED00...ED0F
Cont. no remanentes	ED10...ED3F	ED10...ED3F
Banderas remanentes	EE00...EE3F	EE00...EE3F
Banderas no remanentes	EE40...EE7F	EE40...EE7F

Tabla 1.2  
Mapa de Memoria (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.

Adicionalmente podemos observar en la figura 1.1:

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.

## 1.2 SISTEMA DE ENTRADAS Y SALIDAS

El sistema de entradas/salidas forma la interfaz 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, presión, etc.

Cada tarjeta de entrada/salida es alimentada 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 VDC, mientras que la parte de mando trabaja a 115 VAC.

### 1.2.1 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 I/O mas frecuentemente utilizados.

DISPOSITIVOS DE ENTRADA DIGITAL	DISPOSITIVOS DE SALIDA DIGITAL
Selectores	Alarmas sonoras
Botones	Control de relés
Sensores fotoeléctricos	Ventiladores
Retroavisos de contactos	Lámparas
Contactos	Contactores
Interruptores de proximidad	Válvulas
Indicadores de nivel	Motores
Arranque/paro de motores	Válvulas Solenoides

Tabla 1.3  
Elementos usuales de entrada y salida

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

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

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

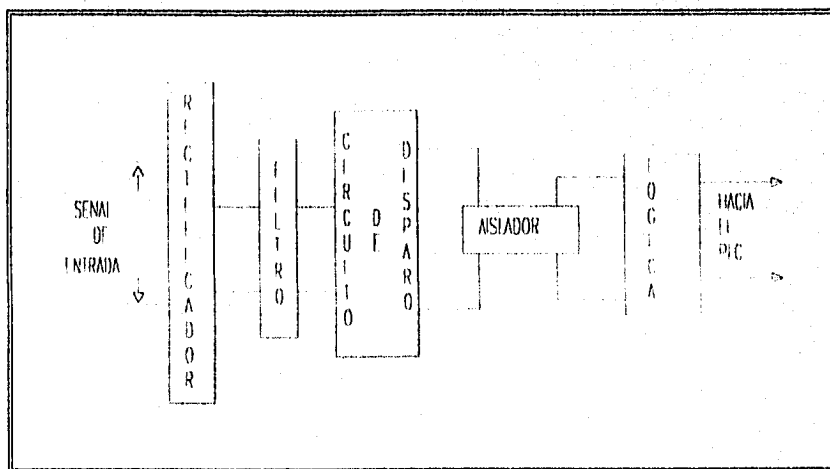


Fig. 1.2  
Diagrama de bloques para interfaz de entrada digital

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. Este nos provee

de separación galvánica, que limita la propagación de una sobretensión, polaridad inversa, o corto circuito, protegiendo de esta forma todo el sistema.

Similarmente, para una salida digital, se suele disponer de una red de protección, a través de un MOV, ó una red RC.

**Módulo de entrada digital 8 x DC 24V  
6ES5 421-8MA12**

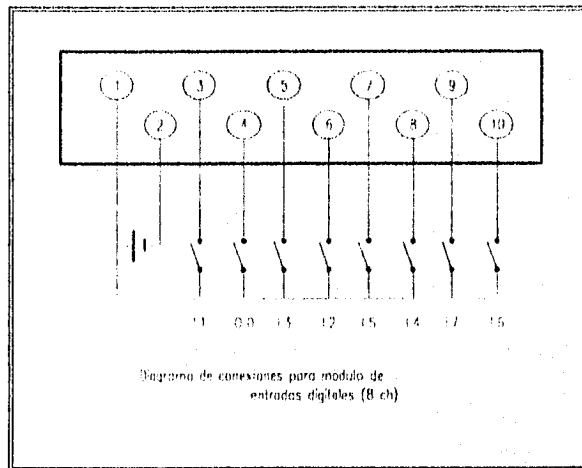


Fig. 1.3  
Diagrama conexiones para un módulo 421 8MA12

Este módulo tiene la función de acoplar señales provenientes de contactos secos (p. ej. 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 señala por medio de un diodo emisor de luz en la parte frontal de la tarjeta.

Direccionamiento: Esta tarjeta puede ser direccionada 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 DC 24V/2A  
6ES5 440-8MA21**

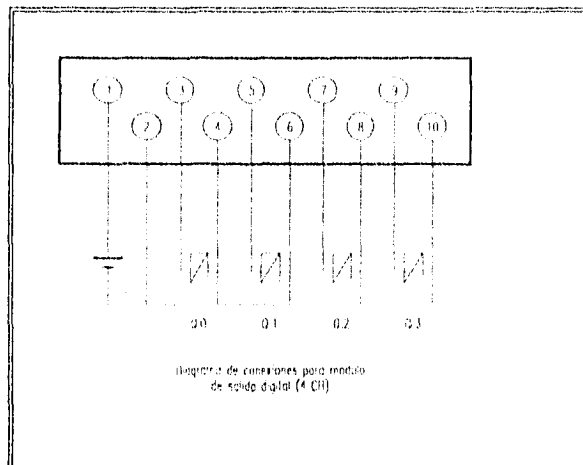


Fig. 1.4  
Diagrama de conexiones para un módulo 440 8MA21

Esta interfaz de salida permite energizar cuatro puntos con una tensión de 24 VDC a una intensidad máxima de 2A. El voltaje de salida se provee a través de transistores. En el diagrama de arriba puede observarse que 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 (p. ej. un contactor) se conecta entre un borne de salida y el potencial 0, tal como se muestra en el diagrama de la figura.

**Notas:**

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

2. Se tiene protección contra corto circuito, con reconexión automática.

3. La tensión de salida baja máxima es de 3V para un 0 lógico.

**Direccionamiento:** Esta tarjeta puede ser direccionada 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 una tarjeta no es la suma de la corriente admitida por cada punto. De esto se deriva la práctica frecuente de conectar en una tarjeta señales que no pueden coexistir simultáneamente.

**1.2.2 Módulos de entrada y salida analógicas**

Las interfaces de entrada/salida analógicas permiten el monitoreo y control de rangos 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, controles de motores (drivers) e instrumentos de proceso.

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

**Módulo de entrada analógica 4 x  $\pm$  50mV  
6ES5 464-8MA11**

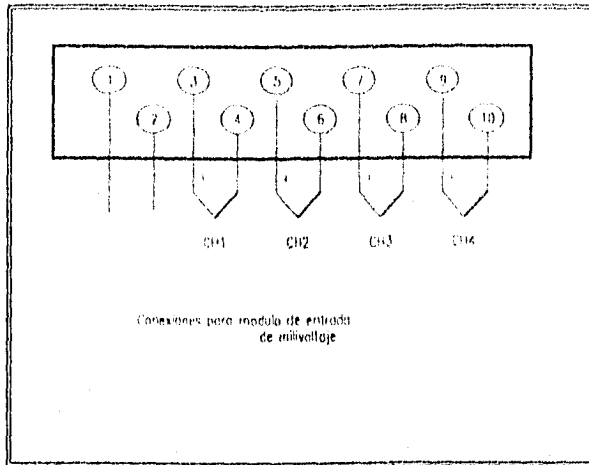


Fig. 1.5  
Diagrama de conexiones para un módulo 464 8MA21

Esta interfaz permite la adquisición de datos analógicos, dentro del rango de  $\pm$ 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 de doble rampa, que realiza con un tiempo de conversión máximo de 80 ms por entrada. La impedancia de cada entrada es de 10 M $\Omega$ , 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 100%. La conexión de las 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
		SW2
		SW3
no	si	Hilo roto

Fig. 1.6  
Significado de los selectores del módulo 464 8MA21

Los Sw2 y 3 se utilizan para definir el número de canales activos, de acuerdo con lo mostrado en la tabla.

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

Nota:

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



Fig. 1.7

Ejemplo de posición de interruptores para un módulo 464 8MA21

X			X	X	
	X	X		X	
1	Ch	2	Ch	4	Ch

Tabla 1.4

Cantidad de canales utilizados

El direccionamiento de un canal de este módulo se hace tomando en cuenta la posición que ocupa dentro de los primeros 8 slots, siendo el primer canal analógico direccionable el PW128.

**Módulo de salida analógica 2 x 1 .. 5 V  
6ES5 470-8MD11**

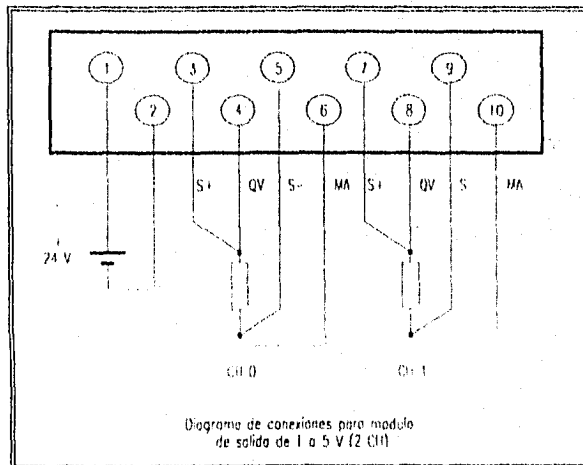


Fig. 1.8

Diagrama de conexiones para un módulo 470 8MD11

Este módulo contiene un convertidor D/A de 11 bits + signo con lo cual es posible la salida de 2048 distintos valores de tensión dentro del margen 1 a 5 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 3300 ohms.

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 tensión 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 entradas de alta impedancia.

Notas:

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

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

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



### 1.2.3 Módulos Especiales

Los módulos que se describen a continuación no son de uso convencional. Algunos de ellos se utilizan como interfaz con el operador, otros como 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

Este módulo se utiliza como una interfaz 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 aún 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.

Nota: este módulo toma su alimentación del bus. No posee conexiones externas.

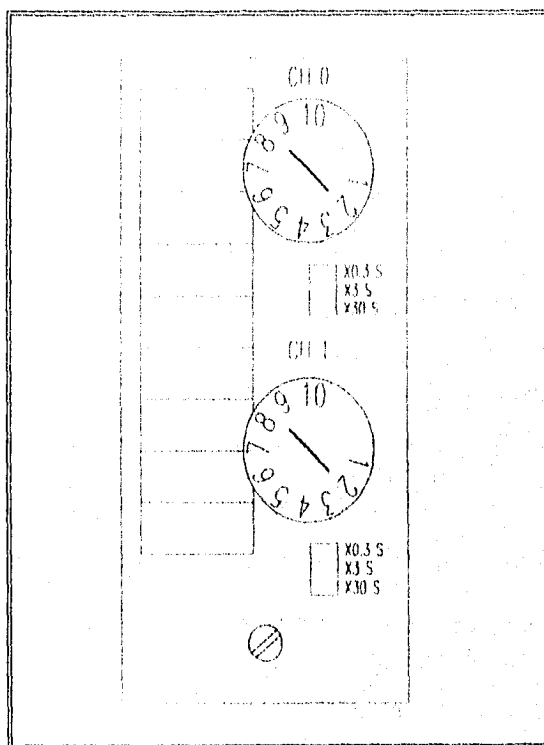


Fig. 1.9  
Módulo de temporizadores

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

Para que un cambio de estado en una entrada digital sea detectado, esta debe mantenerse por lo menos durante un ciclo de scan 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 (Encoders). 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 decrementales 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. La tensión de la señal de entrada puede seleccionarse entre 5 y 24 VDC. Se tiene separación galvánica entre las entradas y la CPU.

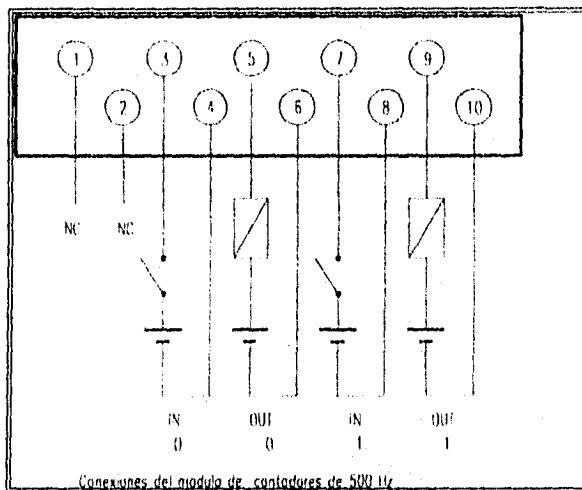


Fig. 1.10

Diagrama de conexiones del módulo de contadores

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**

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

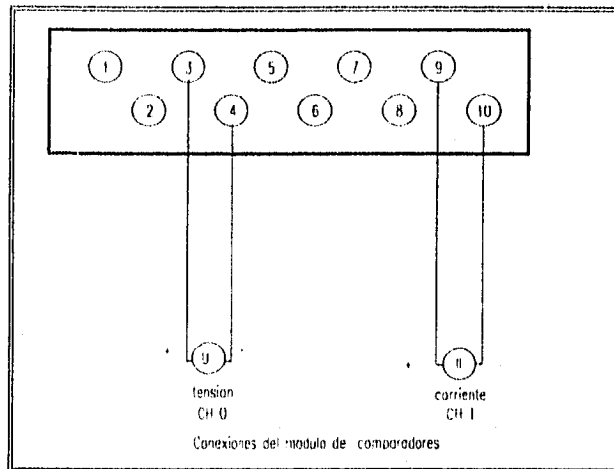


Fig. 1.11

Diagrama de conexiones para un módulo de comparadores

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  $47k\Omega$  para medición de voltaje, y de  $500\Omega$  para medición de corriente. La histéresis es menor o igual al 10%

**Módulo simuladores de entradas ó salidas digitales  
6ES5 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 las perillas 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. Las perillas son inoperantes cuando el módulo ha sido configurado como salida.

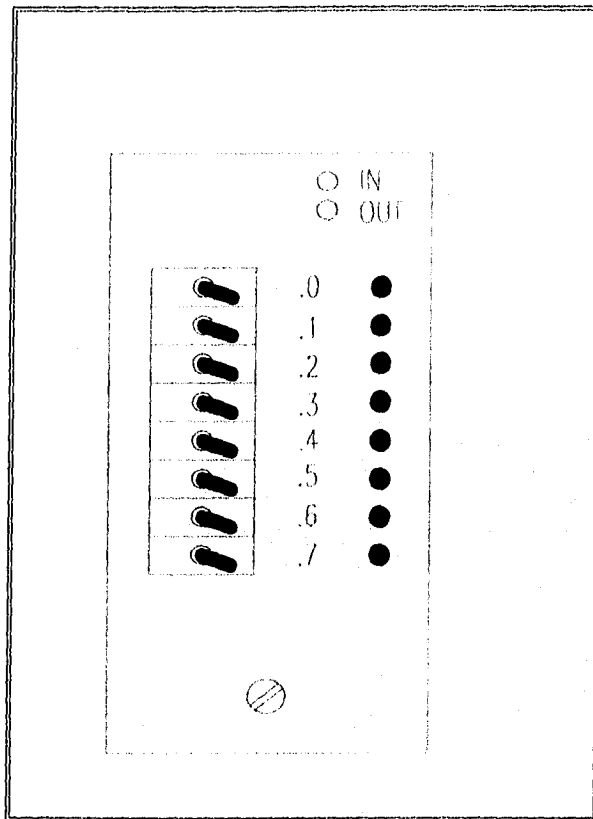


Fig. 1.12

Módulo de simulación de entradas y salidas digitales

### 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 autómata, 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:

1. 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.

2. Módulo EEPROM: Este dispositivo puede ser borrado y grabado directamente por los aparatos de programación o por el autómata 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.

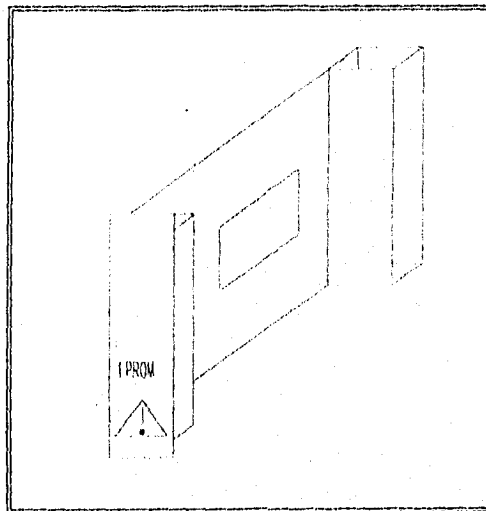


Fig. 1.13

Cartucho de memoria EPROM/EEPROM

### I.3 EL BUS SIMATIC

El bus sirve para comunicar las interfaces 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 de scan, por medio del bus. Esto permite una gran reducción del cableado porque todas las tarjetas de I/O envían la 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.

Debido a su configuración modular, en un sistema Simatic S5-100U el bus se construye al unir componentes de hardware que reciben el nombre de *elementos de bus*. Estos se pueden observar en la figura a continuación:

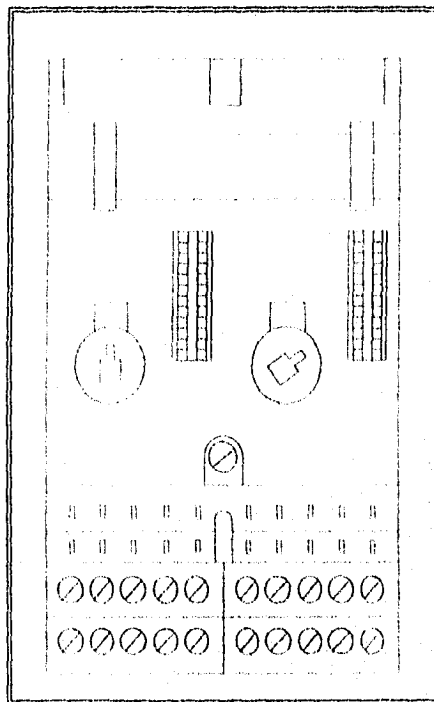


Fig. 1.14  
Elemento de bus 6ES5 700-8MA11

#### I.3.1 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. Cuentan 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 rotar para dar acceso a módulos analógicos de entrada y salida, y módulos digitales de I/O, así como también

módulos inteligentes. La llave es una pieza de plástico que se encuentra ubicada en el centro inferior del elemento de bus.

En cada CPU pueden conectarse hasta 16 elementos de bus, los cuales se interconectan por medio de un cable plano, y dos terminales ya integradas. Se pueden colocar dos módulos en cada elemento, con lo cual se tiene una 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 de I/O
- Ganchos para colocar los módulos de I/O
- Tuercas para asegurar en su lugar los módulos de I/O
- Bornes de conexiones para alambrear 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 I/O 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 autómata 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 conector 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 desplazamiento. 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 desplazamiento propio, que sustituye al registro de desplazamiento del puesto de enchufe.

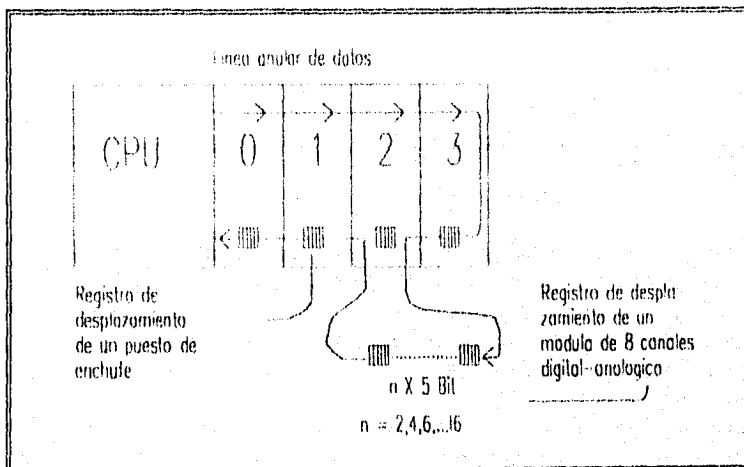


Fig. 1.15  
Bus de comunicación

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

### 1.3.2 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 se conoce como mapa de direcciones. Las direcciones contemplan siempre información referente a un canal de alguna tarjeta específica. La configuración de entradas/salidas es un dato importante que se debe tener en mente al hacer el programa de aplicación.

Para la serie Simatic S5-100U el direccionamiento es fijo, es decir, depende exclusivamente de la posición física del módulo. Para intercambiar información con un módulo es preciso conocer su ubicación dentro del bus, es decir, su dirección, porque ésta se usa en el programa como parte de la definición del operando. Toda la periferia del Simatic se arregla en grupos de 8 bits, llamados entonces Byte. Se empieza a numerar desde 0, que es la posición contigua a la CPU, y se prosigue en orden ascendente hacia la derecha. Estos números son llamados "puestos de enchufe". Vea la figura a continuación en la que se ilustra la numeración para la configuración en una fila de 32 módulos de I/O.

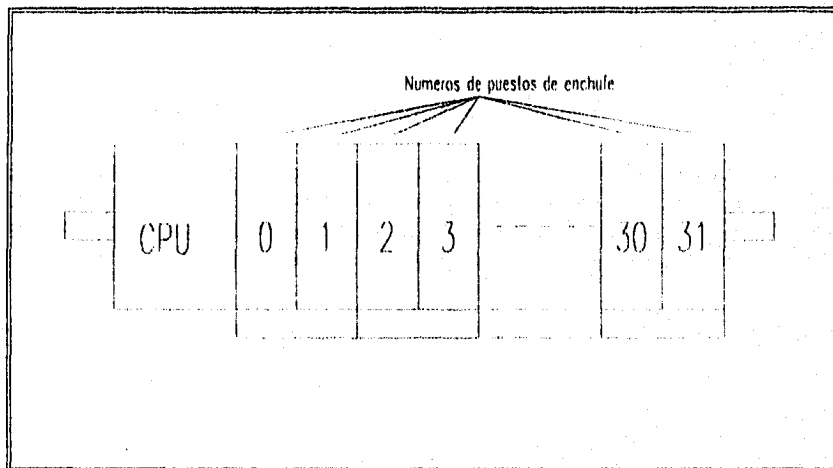


Fig. 1.16  
Configuración en una fila

Obsérvese que en cada elemento de bus pueden enchufarse hasta dos módulos, y que cada puesto de enchufe corresponde a un Byte, y cada elemento de bus a una palabra. ( El primer elemento de bus corresponde a la palabra 0, cuya parte alta es el byte 0, y su parte baja el 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 capítulo 2.

Cuando el autómata se configura en varias filas, por medio de la utilización de interfaces IM 315 / IM316, la numeración de las filas de ampliación prosiguen en el puesto de enchufe situado más a la izquierda.



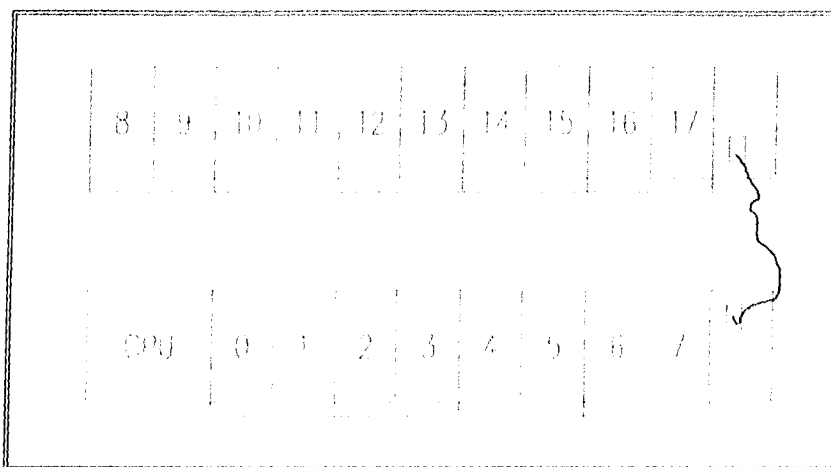


Fig. 1.17

Configuración en dos filas con una IM315

La interface de comunicación IM 316 permite configurar el 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 o desde un módulo digital sólo es posible transmitir por cada canal dos estados, "0"(desconectado) ó "1"(conectado); esto puede realizarse con operaciones combinatoriales (lógicas) binarias o 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 no es un estado "0" ó "1" como en los módulos digitales, sino que es posible comunicar por cada canal 65,536 informaciones diferentes(16 bits=2 bytes=1 palabra). Los módulos pueden direccionarse byte a byte o palabra a palabra usando operaciones de carga o transferencia.

Es válida cualquier combinación de módulos de entrada y salida analógica en los puestos del 0 al 7.

## 1.4 APARATOS DE PROGRAMACIÓN.

El Programador (PG) es un dispositivo que sirve como interfaz entre el usuario y el PLC. A través de este dispositivo es posible introducir (cargar) el programa desde una unidad de almacenamiento como puede ser un cartucho de memoria, un disco flexible ó un disco duro, hacia la memoria RAM del PLC, y también en sentido opuesto, desde el PLC hacia una unidad de disco. El programador debe contener el software y el hardware adecuados para lograr su objetivo.

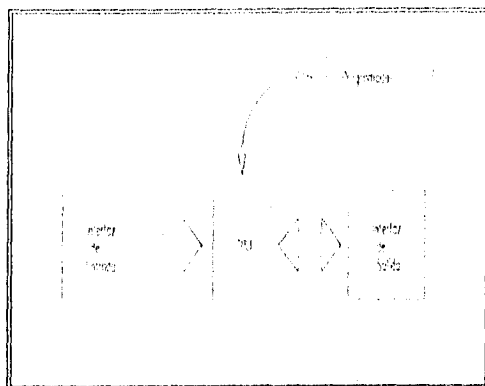


Fig. 1.18

Interacción entre PLC y PG

El programador tiene también funciones de diagnóstico, a través del forzado y la visualización de variables, ejecución del programa paso a paso, y muestra de datos del programa en línea. Estas funciones son estándares en el Software conocido como STEP 5.

Otras funciones del programador son la parametrización de diversos dispositivos periféricos, tales como procesadores de comunicación, paneles de operación, tarjetas inteligentes, tarjetas especiales, redes de comunicación. Esto se hace a través de Software dedicados (COM TEXT, por citar un ejemplo), y del cable apropiado para la tarea en cuestión.

En adición a las tareas propias de los programadores, ya que están basados en computadoras AT, es posible correr cualquier otro software que sea compatible con MS-DOS (como es el caso de Windows).

La familia de programadores de Siemens comprende estaciones de mano (PG 605 y PG 615), portátiles tipo Laptop (PG 710 y PG 730), y programadores universales (PG 740, PG 750 y PG 770).

### 1.4.1 Descripción del PG 710.

El PG 710 es una unidad de programación de la compañía SIEMENS, computadora tipo Laptop, con un procesador 386, y una pantalla monocromática de cristal líquido. Cuenta con un disco duro de 80 MB, y una unidad de disco de 3.5 de alta densidad. Su mayor ventaja es su alto grado de portabilidad, debido a su bajo peso y a su operación con baterías recargables.

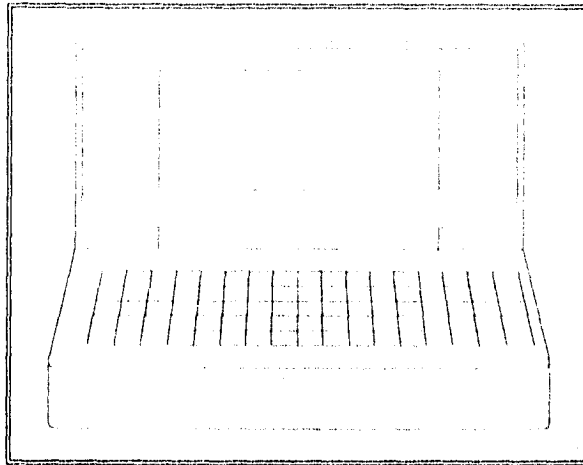


Fig. 1.19

El aparato de programación PG 710

En cuanto al hardware necesario para la programación de autómatas Simatic, cuenta con un puerto serie de tipo lazo de corriente TTY (COM 1), un puerto para programación de módulos de memoria del tipo EPROM y EEPROM, y teclado específico para programación en STEP 5.

Para la programación ó lectura de un módulo de memoria E/EPROM, es necesario que esté conectada la fuente de alimentación externa. El cartucho se conecta en el receptáculo que se ubica en la parte superior derecha del teclado, y se selecciona el paquete E/EPROM dentro del paquete de Software STEP 5. Se cuenta con un Diodo emisor de luz que señala cuando se está ocupando el puerto de programación de E/EPROM. El cartucho no debe ser retirado mientras esté encendido el diodo emisor de luz.

#### **1.4.2 Modo de conexión de un PG 710.**

El Programador PG710 se comunica con una variedad de dispositivos periféricos, los mismos que se podrían utilizar en cualquier computadora AT compatible.

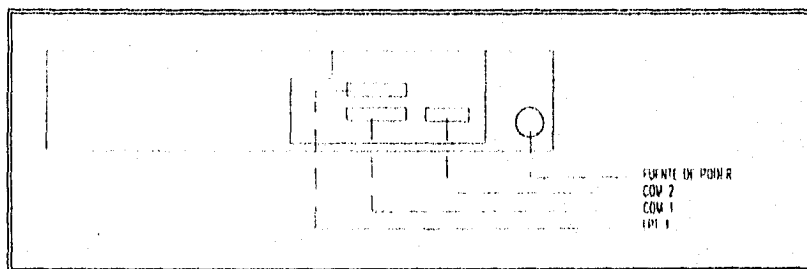


Fig. 1.20

Puertos del PG 710 (vista posterior)

Para conectar un PG 710 con un PLC Simatic de la familia S5-100U es necesario un cable para la transmisión serial en lazo de corriente, el cual es suministrado con el número de parte 6ES5 734-2BD20. Este cable tiene un conector de 25 puntas del lado del programador, y un conector de 15 puntas en el extremo del PLC. Este cable se conecta en el puerto COM 1 del PG710, el cual es de 25 puntas, y se asegura por medio de tornillos. El otro extremo del cable se enchufa en el conector frontal de la CPU del PLC, y se asegura con un candado integrado en el conector. Es necesario que las conexiones sean seguras para evitar malfuncionamiento de la interfaz, y daños al puerto COM 1 del programador.

En el caso de que se tenga un puerto serie de 9 puntas, como es el caso cuando se utiliza una computadora AT y no un programador Siemens, se puede utilizar un adaptador de 25 a 9, en

conjunto con el cable 6ES5 734-1BD20, el cual tiene integrados los componentes adecuados para convertir una señal de voltaje a una de corriente.

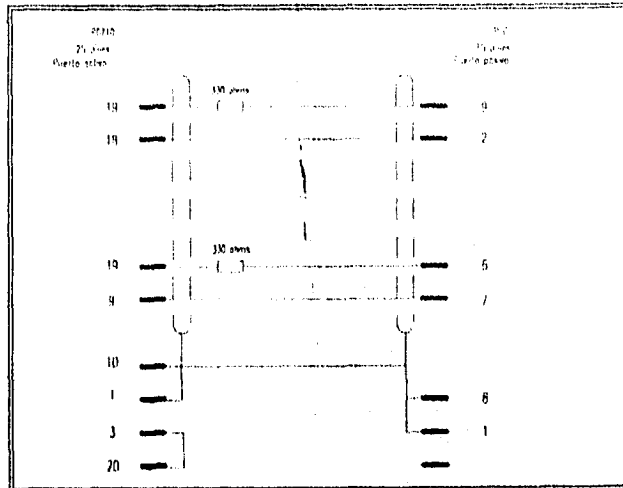


Fig. 1.21

Cable de conexión PG-PLC 2BD20

## CAPITULO II

### Arquitectura en Software del Simatic S5 100-U



## II.1. EL LENGUAJE DE PROGRAMACIÓN STEP 5

La familia Simatic está integrada por cuatro PLC de alcances diversos, conocidos como Simatic S5-100U, S5-115U, S5-135U y S5-155U. Todos estos poseen un mismo lenguaje de programación, llamado STEP 5.

El Step 5 es un intérprete. Sirve de interfaz entre el operador y la CPU, y le indica a esta última las acciones que debe ejecutar. El Step 5 utiliza mnemónicos para todas las operaciones. Aún cuando este lenguaje no es de alto nivel, tampoco es un código de máquina. Más bien se asemeja a 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 computadora portátil (PG 750/740), y también se dispone del programa para ser instalado en una computadora personal 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 lista 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 Simatic 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.

### II.1.1 Tipos de Representación.

Como ya ha sido expuesto 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 bloques 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 PLC, por la 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 de 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 con el familiar símbolo del círculo en la entrada. Las 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.

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 relacionado 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 equivalente en LAD o CFS.

El presente trabajo utiliza programación en STL debido a que esta es la única forma de tener acceso a comandos avanzados.

## II.1.2 Definiciones

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

### Ciclo de scan:

El PLC Simatic corre permanentemente en una secuencia definida, que se conoce como ciclo de scan. Esta sucesión de pasos se ilustra a continuación.

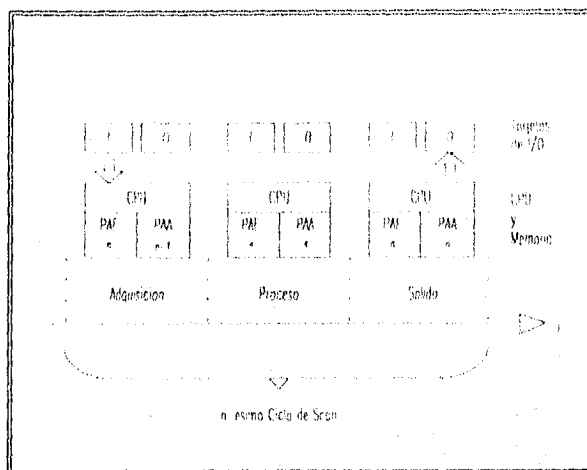


Fig. 2.1  
Ciclo de Scan

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 un área de memoria conocido como *Imagen de proceso de entradas*, ó 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 un área de memoria conocido como *Imagen de proceso de salidas* ó PAA. El tercer paso consiste en utilizar el bus para transferir la PAA hacia las tarjetas de salida que se tienen conectadas.

En resumen, el ciclo de scan se compone de tres pasos, que son:

- Adquisición de la imagen de entrada PAE
- Proceso ó ejecución del programa de aplicación
- Salida ó 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 o hasta tres letras o símbolos) o por un símbolo gráfico

### Operando:

Indica la variable sobre la cual se ejecutará la operación. Los tipos de operandos pueden ser: Entradas, salidas, banderas, 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 que esté 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, de un salto, 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 estándar para todos 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.

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:

A I 0.1

En donde:

- Operación : "A", And ó scan por un 1.
- Operando: "I 0.0", Entrada Digital 0.1
- Identificador : "I", Entrada Digital
- Dirección: "0.", Puesto de enchufe 0
- Dirección: ".1", segundo canal

**Estructuras Bit, Byte, Word**

Algunas veces es necesario manejar simultáneamente más información que un sólo bit por lo que el procesador trabaja con un conjunto de bits, llamado byte. Un byte contiene 8 bits de información. La tercera y última estructura de información es word, que consiste en un grupo de bits. La longitud de palabra utilizada en el Simatic es de 16 bits.

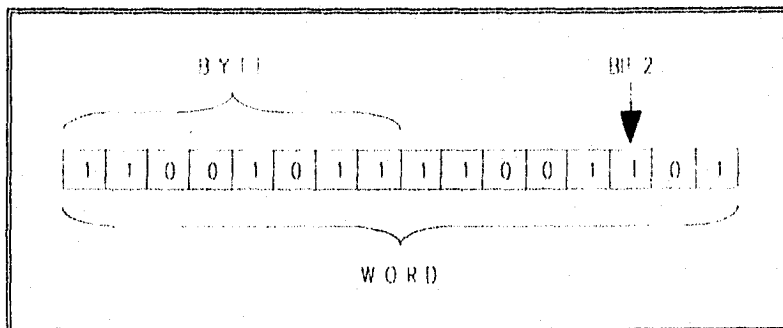


Fig. 2.2

Esquema de la organización por Word, Byte y Bit

## II.2 OPERACIONES BÁSICAS

Este tipo de operaciones pueden ser programadas con un PG 605 ó con un PG 710. permiten el control de I/O digitales, bits de banderas, operaciones de tiempo, y de contadores.

OPERACIONES	OPERANDOS	DESCRIPCION
A AN O ON	Q, I, M	Consulta el estado de uno seno o estado.
=	Q, F	Asigna el resultado de una operacion logica a un operador digital, tipo salida o bandera.
S R	Q, F, C	Coloca un valor determinado en el operador. (S=1, R=0)
Sr SD SF SS SL R	T	Operaciones con temporizadores
CU CI	C	Operaciones con contadores
O( ) A( )	Q, I, F, C	Encadena los resultados de dos operaciones logicas separadas.

Fig. 2.3  
Operaciones básicas del STEP 5

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, inclusive en ecuaciones con paréntesis.

Las operaciones binarias comprenden:

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

## II.2.1 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 de tipo bit, y utilizar posteriormente para la 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 ó 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 lógico con el RLO ó scan por un 0

Tabla 2.1

### Operación "A":

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

RLOi0	Entrada	RLOi1
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 2.2

Esta es igual a la tabla de verdad de la función lógica AND, con la diferencia de que uno de los operandos se modifica después de ejecutada la instrucción.

Nota: Esta tabla no es válida cuando se trata de una primera consulta. En ese caso se interpreta más bien como "scan por un 1", y 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 está activada. Equivale a la combinación en serie de contactos normalmente abiertos.

### Operación "AN":

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

RLOi0	Entrada	RLOi1
0	0	0
0	1	0
1	0	1
1	1	0

Tabla 2.3

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

**Operación "O":**

Equivalente a OR. En la primera consulta asigna un valor "1" en el RLO cuando la entrada está activada, o tiene un valor "1". En otros casos se obtiene la tabla:

RLOt0	Entrada	RLOt1
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 2.4

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

**Operación "ON":**

Equivalente a NOR. En la primera consulta coloca un "1" en el RLO cuando la entrada en cuestión está desactivada, o tiene un valor "0".

RLOt0	Entrada	RLOt1
0	0	1
0	1	0
1	0	1
1	1	1

Tabla 2.5

En un diagrama de escalera equivaldría a la conexión en paralelo de contactos normalmente cerrados.







EQUIVALENCIA LÓGICA		LISTA DE INSTRUCCIONES	
100 101		Q 50	A+00 A+01 Q 50
100 101		Q 50	AN+00 AN+01 Q 50
100 101 102		Q 50	A+00 A+01 AN+02 Q 50
100 101		Q 50	O+00 O+01 Q 50
100 101		Q 50	ON+00 ON+01 Q 50
100 101 102		Q 50	ON+00 ON+01 O+02 Q 50

Fig. 2.4

Tabla de equivalencia entre listas de instrucciones y compuertas lógicas

## II.2.2 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 banderas.

Nota: El set y reset se ejecutan con dependencia del RLO, mientras que la asignación es incondicional.

Ejemplo:

S F 0.0: El bit 0 de la bandera 0 es colocado a "1", a condición de que el RLO sea "1"

R F 0.1: El bit 1 de la bandera 0 es borrado (se le asigna un "0"), a condición de que el RLO sea "1"

= F 0.2: El valor contenido en el RLO es asignado al bit 2 de la bandera 0, sin condiciones.

La última instrucción ejecutada tiene prioridad sobre las demás, ya sea un set o reset.

## II.2.3 Operaciones con temporizadores

Generalmente es necesario utilizar temporizado 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:

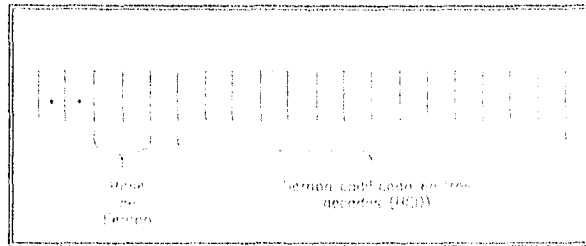


Fig. 2.5

Formato de temporizadores en una Data Word

Una vez que se ha cargado un 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 por la base de tiempo.

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

- 0: es el multiplicador X 0.01 seg
- 1: es el multiplicador X 0.1 seg
- 2: es el multiplicador X 1 seg
- 3: es el multiplicador X 10 seg

De esta forma la mínima cuenta posible es de 0.01 segundo (1 milisegundo) y la mayor cuenta posible es de 9990 segundos (2.775 horas).

Para mayores lapsos de tiempo se pueden conectar varios temporizadores en cascada, ó combinaciones de temporizadores y contadores, ó contadores.

Se recomienda siempre para lograr la mayor precisión, utilizar la menor base de tiempo disponible.

Ejemplo:

Para contar 30 segundos, es posible utilizar LKT 30.2. Sin embargo, la instrucción LKT 300.1 producirá una mayor exactitud.

En las próximas páginas se muestran los 5 tipos de temporizadores que se disponen en el software Step 5.

**SP Tipo pulso.**

Arranca la temporización con el pulso de subida del RLO. Cuando el RLO vale "0", la salida del temporizador se fuerza a "0".

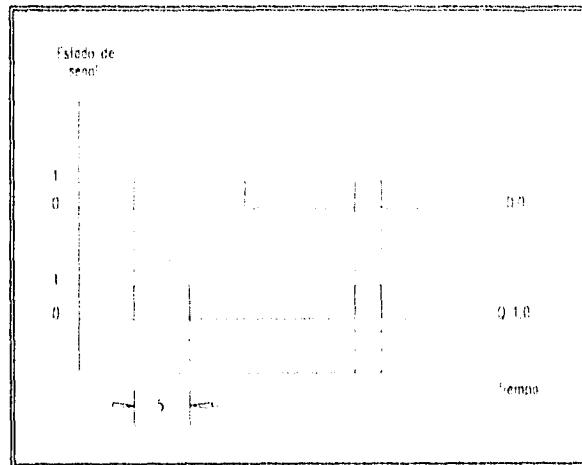


Fig. 2.6

Respuesta de un temporizador tipo Pulso

**Ejemplo:**

Supongamos que se desea limitar la anchura de una bandera F3.5 a un máximo de 5 segundos cuando la entrada 3.1 está presente. Esto se programa como a continuación:

```
A I 3.1  
LKT 500.0  
SP T 1  
A T 1  
= F 3.5
```

**SE Tipo pulso prolongado.**

Arranque con el pulso de subida del RLO. Una vez que se ha iniciado la temporización, el temporizador no es afectado por el estado del RLO.

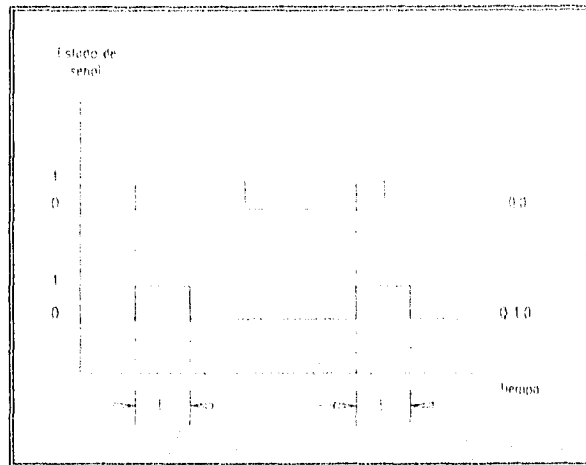


Fig. 2.7

Respuesta de un temporizador tipo Pulso prolongado

**Ejemplo:**

Supongamos que queremos activar la salida 3.7 por 3.4 seg. apenas se percibe un 0 lógico en la bandera 27.5

```
AN F 27,5  
LKT 340.0  
SE T 17  
A T 17  
= Q 3.7
```



**SD Tipo retardo al encendido.**

La temporización se inicia con el pulso de subida del RLO. Cuando el RLO vale "0", la salida del temporizador se fuerza a "0". La operación de examinar el estado de la señal resultará en "1" cuando el temporizado haya terminado y el RLO todavía esté activado.

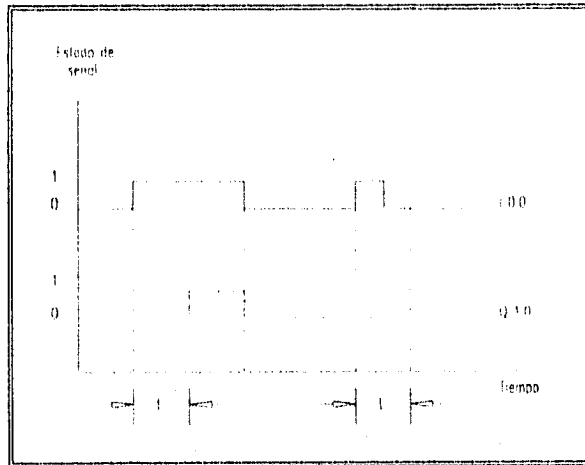


Fig. 2.8  
Retardo al encendido

**Ejemplo:**

Supongamos que se desea activar la bandera 5.0 7 segundos después de que se ha presentado la entrada 15.3. Utilizaremos para esto el temporizador 21.

```
A I 15.3
LKT 7.2
SD T 21
A T 21
= F 5.0
```

**SS Tipo retardo a la conexión memorizada.**

Este es similar al temporizador SD, con la diferencia de que los cambios de la señal de entrada (RLO) no reinician el temporizado.

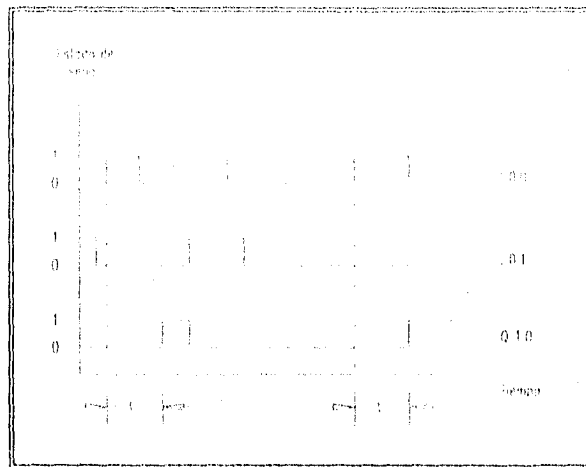


Fig. 2.9  
Retardo a la conexión con memoria

Ejemplo:

Se desea que la salida 2.5 se energice 2.5 seg. después de activada las banderas f 15.7, 4.0, o de borrada la bandera 17.0, aún momentáneamente. La salida debe permanecer energizada mientras no se aplique un reset al temporizador 27.

O F 15.7  
O F 4.0  
ON F 17.0  
LKT 25.1  
SS T 27  
A T 27  
= Q 2.5

**SF Tipo retardo a la desconexión.**

El arranque se efectúa con el pulso de bajada del RLO. Cuando el RLO es "1", el temporizador se carga con el valor de temporizado. La operación de examinar el estado de la señal resultará en "1" mientras el RLO en la entrada sea "1" o el temporizador se encuentre corriendo.

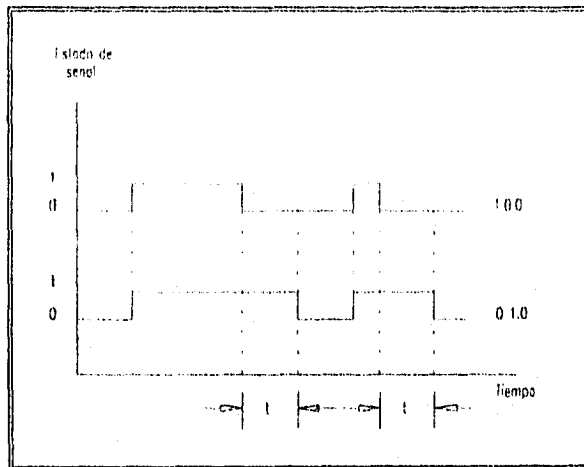


Fig. 2.10  
Retardo a la desconexión

**Ejemplo:**

Se desea que la salida 4.4 permanezca energizada desde que la entrada IW 20 sea igual a la constante hexadecimal 0033 hasta 300 segundos después de que ya no lo sea. Hacer esto auxiliado por el temporizador T31.

```
LKH 0033
L I W 20
I=F
L K,F 300.2
SF T 31
A F 31
= Q 4.4
```

## II.2.4 Operaciones con contadores

En los controladores programables se tienen implementados elementos de conteo como parte del mapa de 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 scan.

Para utilizar un contador se dispone de una entrada de conteo hacia arriba, y una de conteo hacia abajo. Es posible proporcionar un valor inicial para el contador; dicho número se codifica en BCD, y el rango de conteo es de tres décadas, es decir, desde 000 hasta 999. Al llegar al máximo el contador no se borra automáticamente, sino que conserva dicho valor aún cuando siga recibiendo instrucciones de incrementar la cuenta por medio de pulsos de entrada.

Se tienen, un total de cuatro posibles operaciones con contadores, que son el *set*, *reset*, *incrementar contador*, *decrementar contador*.

La operación *set* se ejecuta en el pulso de subida del RLO. Su efecto es transferir el número contenido en el ACCU 1 hacia la localidad de memoria que constituye el contador. Este es el valor inicial del contador, y a partir de ahí puede contarse hacia arriba ó hacia abajo.

La operación *reset* mantiene el contador en 000 durante el tiempo en que el RLO esté en "1". Obviamente, borrará el contenido previo del contador; y no aceptará pulsos de entrada.

Generalmente el conteo se efectúa en el pulso de subida de una entrada digital, una bandera o una operación de comparación.

Ejemplo:

Borrado de un contador :    A I 7.3  
                                  R C 0

Carga (Set ) del contador 17 con un valor inicial de 730:

                                  A I 32. 0  
                                  L K C 730  
                                  S C 17

El contador 1 cuenta hacia arriba cada vez que la entrada 5.7 recibe un pulso.

                                  A I 5.7  
                                  C U C 1

El contador 28 cuenta hacia abajo cada vez que se activa la salida 3.3

                                  A Q 3.3  
                                  C D C 28

### II.3 OPERACIONES DIGITALES CON PALABRAS

Estas son un conjunto de instrucciones que operan con palabras.(es decir, con grupos de 16 bits). En los PLC Simatic las palabras tienen el siguiente formato:

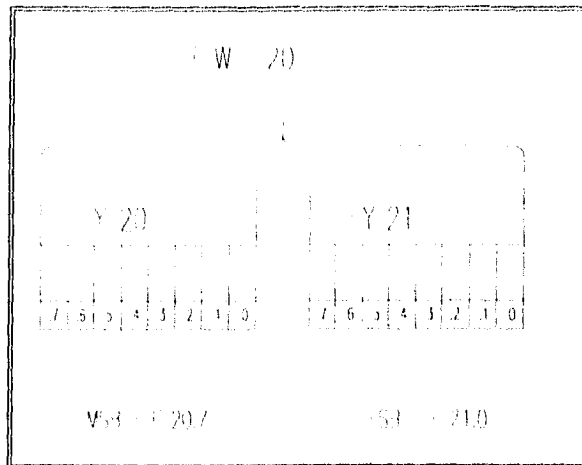


Fig. 2.11  
Formato de palabra

Se puede observar que una palabra está compuesta por dos Bytes, y que están ordenados de manera tal que el Byte alto es aquel con la numeración par (en este ejemplo el Flag Byte FY 20) y a su vez el Byte bajo es el de la numeración impar.

Esto es más fácil de recordar si consideramos que la primera Bandera disponible como palabra es la FW 0 = FY 0 + FY 1

Esta forma de interpretar las banderas es la razón por la cual solamente se deben direccionar palabras pares. Dicho de otro modo, es correcto direccionar las FW 0, 2, 4, .... pero es un error direccionar las FW 1, 3, 5, ...

En los bloques de datos DB las palabras se organizan de manera distinta, por lo que su uso es más transparente. En los DB es permitido direccionar las DW 0, 1, 2, 3, 4, .... hasta la longitud que haya sido creado el bloque de datos.

### II.3.1 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 el ACCU 1 y el 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 ACCU 2.

Existe también la posibilidad de cargar un número codificado en BCD, lo cual suele ser útil para las operaciones con contadores y temporizadores. Esta instrucción debe utilizarse con cuidado, recordando siempre que el comparar un número en BCD contra un número de punto fijo trae generalmente consecuencias inesperadas.

Los acumuladores pueden cargarse con información procedente de las entradas, salidas, banderas, palabras de datos, temporizadores y contadores, así como también es posible utilizar constantes, las que pueden ser de coma fija, configuraciones binarias, números hexadecimales, o caracteres alfanuméricos (ASCII).

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

Ejemplos:

L DW 20	Carga en el ACCU 1 el valor contenido en la palabra de datos 20
LC C 15	Carga codificado en BCD el valor del contador 15
L KB 240	Carga la constante de punto fijo 240
L KH ABAD	Carga la constante hexadecimal ABAD
L KF -31985	Carga la constante de punto fijo -31985 en forma de complemento a dos.
T DW 20	Copia el contenido del ACCU 1 en la palabra de datos 20 del módulo de datos activo.
T QW 6	Copia el contenido del acumulador (ACCU 1) en la palabra de salida

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
```

producirán exactamente el mismo efecto en una operación de comparación, ó de aritmética de punto fijo de 16 bits.

### II.3.2 Combinaciones digitales

En estas operaciones se utilizan como operandos la información depositada en los acumuladores, las cuales se combinan 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 ejemplo que se muestra a continuación se muestra el contenido de los acumuladores 1 y 2 a través de la ejecución de la siguiente secuencia de instrucciones:

```
L I W 37
LKH OFF0
A W
LKH 0880
O W
```

T DW 91

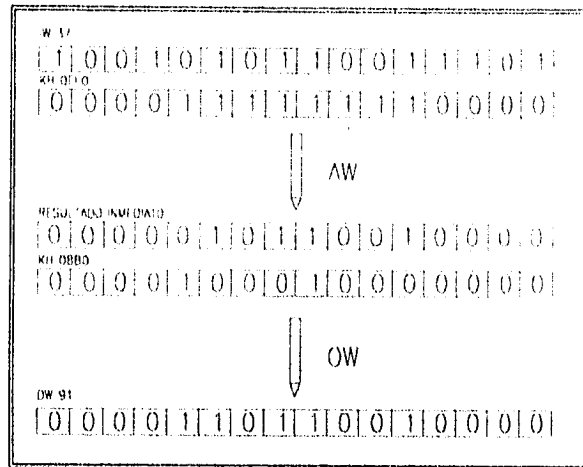


Fig. 2.12

Ejemplo de combinaciones AW y OW

El resultado de cualquier combinación siempre se almacena en el ACCU 1. Debe notarse que no se dispone de una combinación NAND, así como tampoco una NOR. Más adelante veremos la forma de invertir todos los bits de una palabra.

### II.3.3 Operaciones aritméticas

Se tienen solo dos operaciones aritméticas, que son la adición y la sustracción, ambas en punto fijo. Esto significa que los operandos (los números contenidos en los acumuladores), se interpretan como números de punto fijo, y el resultado se almacena en el ACCU 1.

- +F Adición. Se suman los contenidos de ambos acumuladores
- F Sustracción. Al contenido del ACCU 2 se le resta el del ACCU 1.

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

Ejemplo:

En la figura a continuación se muestra el contenido de los acumuladores al ejecutar la siguiente lista de instrucciones:

```
LIW30
LIW32
+F
LDW17
-F
TDW18
```

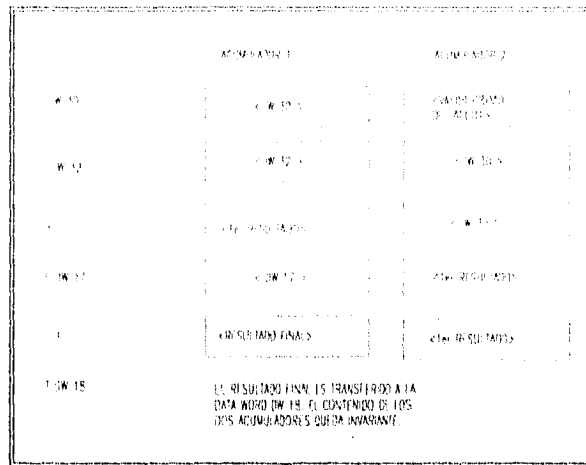


Fig. 2.13

Contenido de los acumuladores durante operaciones de suma y resta

### II.3.4 Operaciones de comparación

Estas operaciones actúan sobre los acumuladores, en los cuales se han cargado previamente los operandos. Las comparaciones se ejecutan incondicionalmente y generan un resultado de tipo bit, el cual se almacena en el RLO. En caso de que la condición se verifique, el RLO contendrá un "1", en cualquier otro caso, contendrá un "0"

En estas operaciones se compara el número del acumulador 2 contra el número del acumulador 1, y suponen siempre que los números están en el formato de punto fijo. Esto significa que FFFF es menor que OFFF. El formato de cualquier operación utiliza el contenido de los acumuladores de la siguiente forma:

ACCU 2 (operación) ACCU 1

Las posibles operaciones y su significado se detallan a continuación:

- != F ¿ACCU 2 es igual a ACCU 1?
- >< F ¿ACCU 2 es distinto de ACCU 1?
- > F ¿ACCU 2 es mayor que ACCU 1?
- >= F ¿ACCU 2 es mayor o igual que ACCU 1?
- < F ¿ACCU 2 es menor que ACCU 1?
- <= F ¿ACCU 2 es menor o igual que ACCU 1?

Ejemplo:

LF W 15

L DW 33

!=F

JC PB 1

El salto al PB1 está condicionado a que la palabra FW 15 sea igual a la palabra DW 33 del bloque de datos activo.

### II.3.5 Operaciones Complementarias

En la CPU 102 existen un grupo de operaciones que **solo pueden ser programadas en bloques de funciones**, y 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 0001 H

### 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.

En la figura a continuación se ilustra las siguientes instrucciones:

LKH 00FF  
SLW 4  
T QW 8

El resultado final será 0FF0

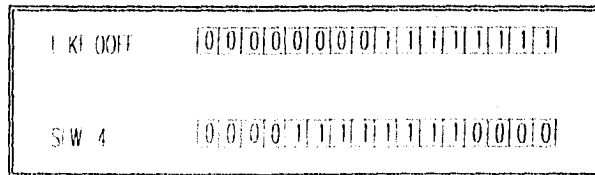


Fig. 2.14  
Operación SLW

SRW n Desplaza a la derecha "n" posiciones el contenido del ACCU 1. Los bits que quedan libres se rellenan con ceros. Los bits desplazados se pierden. Esta función hace lo inverso que la función SLW.

### Operaciones de salto

En los bloques de funciones es posible colocar etiquetas, con las cuales se logra un control óptimo del flujo del programa.

- JU= Salto incondicional a la etiqueta.
- JC= Salto condicionado a la etiqueta.
- JZ= Salta a la etiqueta si el resultado es cero.
- JN= Salta a la etiqueta si el resultado no es cero.
- JP= Salta a la etiqueta si el resultado es positivo.
- JM= Salta a la etiqueta si el signo del resultado es "-".
- JO= Salta a la etiqueta si ocurrió un desbordamiento.

Ejemplo:

L DW 15  
T FW 2  
A F 2.7  
JC = INV

Si el bit 15 de la DW 15 está activado, entonces se salta a la etiqueta " INV".

## II.4 ORGANIZACIÓN DE UN PROGRAMA EN STEP 5

Un programa de mando puede ejecutarse de diversas formas:

- Ejecución cíclica
- Ejecución controlada por alarmas, a partir de la CPU 103
- Ejecución controlada por tiempo, a partir de la CPU 103

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

Como ya se ha explicado, en la última parte del ciclo de Scan, tras la ejecución del programa se corre un ciclo de escritura, durante el cual los datos de la PAA se transfieren a los módulos de salida, y, simultáneamente, se transfieren a la PAE los datos actuales de los módulos de entrada. Con esta PAE actualizada vuelve a realizarse la ejecución del programa.

Antes de la ejecución de la primera línea del OB1 se arranca un tiempo de vigilancia (perro guardián). Si este temporizador expira, el autómata pasa forzosamente a "STOP" y bloquea todos los módulos de salida. Esto ocurre cuando se ha programado un lazo sin fin o cuando surge una avería en la CPU. El tiempo de vigilancia sólo puede modificarse en la CPU 103, a través del OB31.

Esta ejecución cíclica del OB1 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, la CPU retorna al punto de interrupción en el programa cíclico, y prosigue allí su ejecución.

### II.4.1 Tipos de Programación

En la familia S5, el software se ha diseñado de acuerdo a una filosofía modular, lo cual facilita la elaboración de programas, la documentación de los mismos, y estimula la solución de problemas en una forma estructurada.

Para esto existen 5 tipos de módulos o bloques de programa, que son:

OB  
PB  
FB  
SB  
DB

Cada módulo está 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 disponible en la memoria de programa (CPU 100 = 1024 instrucciones, CPU 102 = 2048 instrucciones y la 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.

En el autómata S5-100U el programa de aplicación puede organizarse de dos formas:

- Programas lineales, y
- Programas estructurados.

#### Programación Lineal

El programa de mando está contenido en un solo módulo. La programación lineal es solo posible en el módulo de programa "PB 1" o en el módulo de organización "OB 1". El programa se ejecuta cíclicamente. Este modo de programación es útil solamente en programas muy cortos.

### Programación Estructurada

El programa de mando está organizado en diferentes módulos. El núcleo de la estructura es el módulo 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.

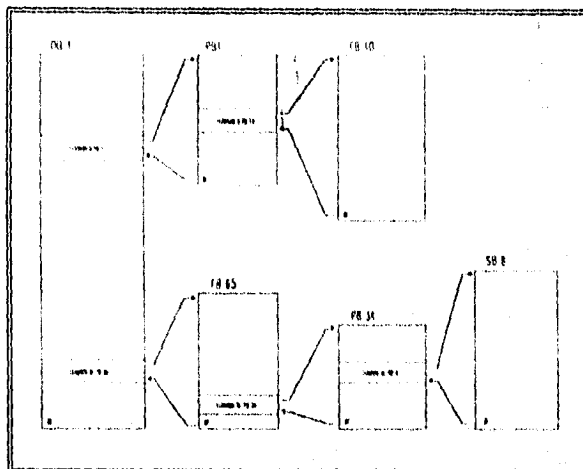


Fig. 2.15  
Programación estructurada

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 de trabajo de cada uno.

#### II.4.2. Tipos de módulos de programa

Es posible programar instrucciones en los siguientes diferentes módulos:

- Módulos de organización (OBs)
- Módulos de programa (PBs)
- Módulos de paso (SBs)
- Módulos funcionales (FBs)

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

#### Módulos de organización "OB"

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

Se disponen de OBs para las siguientes aplicaciones:

- Ejecución cíclica del programa: OB 1
- Rutina de arranque: OB 21 y OB 22
- Fallo de batería: OB 34
- Perro Guardlán: OB 31

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

El sistema operativo llama cíclicamente el OB 1. La primera instrucción del OB 1 marca el principio del programa de mando. La instrucción "BE" (fin de módulo) del OB 1 es la última instrucción del programa de mando.

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

Con ambos OBs es posible fijar el comportamiento del autómeta 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 autómeta estuviera en "RUN" antes de desconectar y no se presente ninguna causa de STOP al conectar, es decir que el OB 22 organiza el comportamiento de re arranque 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 autómeta al ocurrir el evento de falla de batería. Si se da cuenta que ha fallado (porque se ha desconectado o ha caído la tensión más allá de cierto punto), se procesará el OB 34 antes de cada ejecución cíclica del programa (OB 1).

#### Perro guardián usando el OB 31

Un "perro guardián" es un temporizador interno a 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, la 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 reinicializar el conteo del perro guardián en cualquier punto del programa, esto es, re 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 3 representaciones posibles dependiendo del programador que se use. Toda la lógica combinatorial se suele programar en estos bloques.

Ejemplo:

Se desea controlar el arranque de un motor por medio de dos botones, uno de arranque y uno de parada. Ambos botones son normalmente abiertos, y se ubican en el puesto 0, estando conectado el botón de arranque en el canal 0 y el de paro en el canal 7. El relevador que acciona el motor se encuentra conectado en la salida 4.3.

Solución:

Se deben programar los módulos OB1 y PB1

Programa contenido en el OB1

OB1  
JU PB 1  
BE

Programa contenido en el PB1

PB1  
O I 0.0  
O Q 4.3  
A I 0.7  
= Q 4.3  
BE

#### 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. Estos contienen las instrucciones y las condiciones de prosecución.

Es posible programar 256 módulos de paso (SB 0...255), con lo que se dispone de 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.

Características:

- tienen un encabezamiento más largo
- es posible utilizar las operaciones complementarias y de sistema
- solo son programables en STL (lista de instrucciones)
- a partir de la CPU 103 es posible asignar parámetros a los FBs
- para ser llamados desde otro módulo es imprescindible que existan previamente, si no se emite un aviso de error en el programador.

### II.4.3 Módulos Funcionales Integrados

A partir de la CPU 102 se tienen grabados en la ROM 6 bloques de funciones, los cuales ejecutan diversas tareas. Estos son los 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 de valor analógico FB 250
- Salida de un valor analógico FB 251

#### Conversor de código B4

FB 240: Este módulo de programa permite transformar un número de 4 tétradas en BCD con signo en un número binario de punto fijo (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	IW	0...9999
SBCD	Signo del número BCD	I (bit)	"1" para "-" "0" para "+"
DUAL	Número de punto fijo	QW	16 bits

Tabla 2.6

Ejemplo:

Se desea tomar el valor del contador 7 y transformarlo en un número de punto fijo que se escribirá en la palabra de datos 23. El signo del contador será almacenado en el bit 0 de la bandera 2. La bandera 0 se usa como registro intermedio.

```
:LC C 7
:T FW 0
```

```

:JU FB 240
NAME: COD :B4
BCD :FW 0
SBCD:F 2.0
DUAL :DW 23
    
```

### Conversor de código 16

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

Parámetro	Significado	Tipo	Asignación
DUAL	número binario	IW	-32768.. ..+32767
SBCD	signo del número BCD	O BI	"1" para "-" "0" para "+"
BCD2	Número BCD tétradas 4 y 5	Q BY	2 tétradas
BCD1	Número BCD tétradas 0...3	Q W	4 tétradas

Tabla 2.7

#### Ejemplo:

Se desea mostrar en una pantalla de 5 dígitos el valor de una variable de punto fijo que se obtiene como resultado de una operación de 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.

#### Solución:

Se programa el PB1 como se muestra a continuación.

```

PB1
JU FB 241
NAME: COD : 16
DUAL: FW 17
SBCD: F 0.0
BCD2: QY 1
BCD1: QW 2
:BE
    
```

**Multiplicador : 16**

FB 242: Este módulo funcional permite la multiplicación de dos números de punto fijo de 16 bits, y el resultado, representado con un número de punto fijo, se almacena en dos palabras, es decir, 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 BI	"0" cuando el producto es cero
Z32	Producto palabra alta	Q W	16 bits
Z31	Producto palabra baja	Q W	16 bits

Tabla 2.8

**Ejemplo:**

Se desea la multiplicación de dos variables analógicas, provenientes de sensores de peso y velocidad, para obtener el gasto másico de una banda transportadora. Estos datos están depositados en las banderas 1 y 2. El resultado debe almacenarse en las palabras de datos 26 y 27 del bloque de datos 7. Se tiene un piloto que se debe encender en el caso de que el gasto sea cero, y está en el puesto 4, canal 3.

**Solución:**

El FB 242 se parametriza en el PB1 como se muestra a continuación.

```

PB1
:C DB 7
:JU FB 242
NAME: MUL: 16
Z1 :FW 1
Z2 :FW 2
Z3=0 :Q 4,3
Z32 :DW 26
Z31 :DW 27
:BE
    
```

**Divisor : 16**

FB 243: Este modulo permite la división de dos números de punto fijo. El cociente se expresa como un número de punto fijo, así como también el residuo.

Parámetro	Significado	Tipo	Ocupación
Z1	Dividendo	I W	-32768.. +32767
Z2	Divisor	I W	-32768.. +32767
OV	Overflow	Q BI	"1" hubo desbordamiento
FEH	Indicador	Q BI	"1" intentó dividir por cero
Z3=0	Consulta de cero	Q BI	"0": cociente es cero
Z4=0	Consulta de cero	Q BI	"0": residuo es cero
Z3	Cociente	Q W	16 bits
Z4	Residuo	Q W	16 bits

Tabla 2.9

**Lectura y normalización de 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.

Parámetro	Significado	Tipo	Asignación
BG	Puesto de enchufe	D KF	0.. ..7
KNKT	Número y tipo de canal	D KY	KY = x,y
OGR	Límite superior	D KF	-32768.. ..+32767
UGR	Límite inferior	D KF	-32768.. ..+32767
EINZ	-	-	-
XA	Valor de salida	Q W	Valor analógico normalizado
FB	Bit de error	Q BI	"0" cuando no hay error
BU	Margen desbordado	Q BI	"0" cuando no hay error

Tabla 2.10

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 20 mA; tipo 4, es una entrada unipolar; tipo 5 es una entrada de valor absoluto, bipolar; tipo 6, un número de punto fijo, bipolar).



En el Simatic 100-U, el parámetro EINZ, que se utiliza para la activación selectiva de un canal, no se utiliza, puesto que no es posible seleccionar el modo de operación (entre cíclico o selectivo) de un módulo de entrada analógico. Sin embargo, es necesario designar aquí un valor cualquiera.

Los parámetros OGR y UGR determinan la interpretación del valor analógico y fijan los márgenes para la conversión, como se ilustra a continuación:

Ejemplo:

Se tiene un tacogenerador acoplado a la flecha de un motor. La salida del tacogenerador es de 50 mV para una velocidad de 4395 R.P.M., la cual es la máxima velocidad del motor. 0 mV corresponden a 0 R.P.M. Se tiene para esto un módulo de entrada analógica de 4 canales x +/- 50 mV (6 ES5 464-8MA11), enchufado en el puesto 2. El tacogenerador se encuentra conectado en el canal 0.

Dicho valor se depositará en la bandera 2, para su procesamiento en alguna otra sección del programa.

Solución:

Se llama al FB 250 en el PB1 con los parámetros que se muestran a continuación.

```
PB 1  
JU FB 250  
NAME: RLG: AE  
BG :KF +2  
KNKT: KY 0,6  
OGR : KF +4395  
UGR : KF -4395  
EINZ : F 0.0  
XA : FW 2  
FB : F 5.0  
BU : F 5.1  
:BE
```

En este ejemplo se designa el UGR como -4395 debido a que el límite inferior de entrada del módulo es -50 mV (rotación en sentido opuesto). En el caso de que el tacogenerador produjese una salida de corriente en el rango 4... 20 mA, en el cual correspondiesen 4mA = 0 R.P.M. y 20 mA = 4395 R.P.M., se tendría que utilizar una tarjeta de entrada de corriente y la parametrización sería distinta:

```
PB 1  
JU FB 250  
NAME: RLG: AE  
BG : KF +2  
KNKT: KY 0,3  
OGR : KF +4395  
UGR : KF +0  
EINZ : F 0.0  
XA : FW 2  
FB : F 5.0  
BU : F 5.1  
:BE
```

**Salida de un valor analógico**

FB 25t: 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	1W	Cualquier número entre OGR y 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 2.11

El parámetro XE es interpretado 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) o 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 puesto 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 :

Se ha dispuesto de una tarjeta de salida analógica de 1 a 5 Volts en el puesto de enchufe 1, y se desea linealizar un voltaje de 5 Volts para una salida de 30000 y de 1 Volt para el 0 (representación de punto fijo de 16 bits). Los errores se mostrarán en una tarjeta de salida conectada en el puesto de enchufe 0.

```

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 "DB"**

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 el DB a crear depende del espacio disponible en la memoria RAM. En caso de no haber suficiente memoria, el PLC pasa a STOP.

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

En la 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, es requisito la previa 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 el programa lo cierra por causa de un salto a otro bloque de programa, cuando este regresa al módulo desde el que se hizo la llamada. No existe ninguna instrucción capaz de cerrar los módulos.

Ejemplo:

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

Solución:

Se programa el PB 0 como sigue:

```
L KF 90
G DB 10
C DB 10
```

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

```
L KF 0
G DB 10
```

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

## **II.5 DOCUMENTACIÓN DE PROGRAMAS.**

La documentación de un programa consiste en agregar información que ayude a un mejor entendimiento de la forma en que se ha resuelto la tarea de control.

Con los programadores AT y la versión de DOS, es posible introducir líneas de texto al lado de cada instrucción, en un campo de 28 caracteres, para esto se presiona la tecla de tabulación para que el cursor se coloque del lado derecho de la pantalla, o también con la tecla de shift y la flecha derecha.

También es posible colocar un título a cada bloque, con la tecla de shift y el número 5 del bloque de números en el teclado.

Una de las funciones de documentación más útiles 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 podría referenciarse en el programa como la variable "botón de arranque", ó como "X1". De una forma similar, se pueden poner 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, se tiene la generación de la tabla de referencias cruzadas, que nos indica dónde 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 provistas por el PG710.

## CAPITULO III

### Aplicaciones del Simatic S5 100-U



## APLICACIONES

A manera de ilustración de los temas expuestos hasta aquí, se muestran a continuación un grupo de ejemplos de software y en la parte del hardware, se ha provisto de un sistema Simatic S5 102 completo, armado y cableado en un gabinete, con 6 módulos de entrada y salida, que pueden ser reemplazados.

En la parte de software se principia desde un grupo aplicaciones del tipo más básico, para luego exponer un ejemplo de control secuencial, y finalmente un ejemplo que muestra el manejo de números en el PLC.

A través de los ejemplos que se muestran, se facilitará un conocimiento de:

- Estructura de los sistemas numéricos
- Procedimientos para la ejecución del programa
- Aprendizaje de los grupos funcionales de un PLC y su método de operación
- Conexiones lógicas en un PLC
- Funciones de temporizado, conteo, comparación y autoenclavamiento de banderas y salidas en controladores lógicos.
- Procesamiento de valores analógicos
- Procesamiento de Palabras

Para la elaboración de los ejemplos se establece la siguiente metodología de trabajo:

Objetivo  
Descripción del problema  
Aplicación  
Planteamiento del problema  
Configuración del PLC  
Modo de operación  
Listados.

En general recomendamos la lectura de la bibliografía citada ( 6 y 7), que trata más extensamente de la metodología sugerida, y que nosotros hemos adaptado aquí para el desarrollo de los proyectos.

### III.1 TABLERO DE SIMULACIÓN SIMATIC :

Tomando en cuenta que el equipo debe estar a la disposición de cualquier interesado, se ha dispuesto una construcción que es móvil, y flexible, para que pueda ser cambiada de lugar con facilidad, y se puedan usar en ella todos los módulos de que dispone el Laboratorio de Control Digital. El sistema se compone de los siguientes elementos:

1.- Gabinete. Su objetivo es el de proporcionar protección al polvo y golpes al equipo y además provee de los medios de sujeción mecánica para el PLC, sus módulos de entrada y salida, y los bomes tipo banana, de manera que facilite su traslado.

2.- Equipo Simatic S5-100U. El equipo ya instalado en su gabinete y a disposición del Laboratorio de Control Digital consta de:

Un riel de montaje DIN,  
Una fuente de alimentación de 0.8A,  
Una CPU 102 ,  
Tres elementos de bus,  
Un módulo de entradas digitales,

- Un módulo de salidas digitales,
- Un módulo de entradas analógicas,
- Un módulo de salidas analógicas,
- Un módulo de comparadores,
- Un módulo de contadores.

Adicionalmente se dispone en el laboratorio de 2 CPU 102, 2 Memorias de EEPROM 2Kb, 2 Programadores PG 605, 1 Programador PG 710, 2 Cables 1BD20, 2 Paquetes de Software Step 5, 9 Elementos de Bus, 5 Módulos 788.

3.- Borne de conexiones. Se tiene un conjunto de 42 bornes tipo banana, organizados como muestra la siguiente tabla:

Grupo	Cant. de Bornes	Configuración original
1	8	4 canales de entrada analógica
2	4	2 canales de salida analógica
3	4	2 canales de un módulo de comparadores
4	8	2 canales de un módulo de contadores
5	8	8 canales de entrada digital (24 V)
6	8	4 canales de salida digital (24 V)

Tabla 3.1  
Configuración original de módulos del simulador

### III.1.1 Construcción y Módulos Originales

En la figura a continuación se ilustra gráficamente el tablero de simulación tal y como está contenido en su gabinete. Algunas de las ventajas de disponer de los bornes de conexiones es que todo el sistema se encuentra pre-alambrado, con lo cual se ahorra tiempo en la realización de las prácticas.

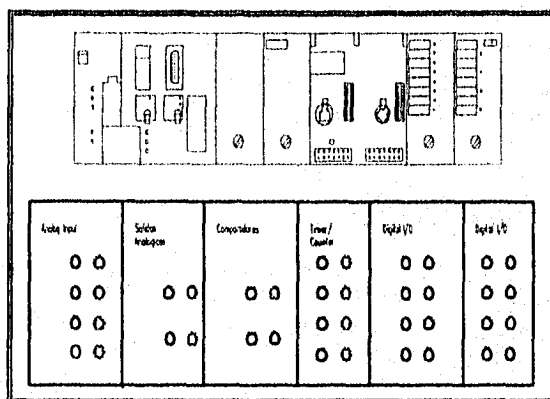


Fig. 3.1  
Configuración del Hardware para Laboratorio

La agrupación de los contactos corresponde a la de los puestos de enchufe de los elementos de bus, de modo que se tienen los puestos de enchufe desde la dirección 0 hasta la dirección 5, de acuerdo a lo observado en la tabla 3.1. De esta forma, el primer grupo corresponde a la dirección Byte 0, y el sexto grupo de bornes, al Byte 5.

### III.1.2 Módulos Sustitutos

El conjunto de bornes de conexiones tipo banana están conectados a las terminales de los elementos de bus como corresponde al diseño de cada módulo para la configuración inicial del



simulador universal, con excepción de los puntos 1 y 2 que en la mayor parte de los módulos corresponde a la alimentación de energía.

Debido a esta circunstancia, los módulos originales sólo pueden ser reemplazados de la manera que se indica en la siguiente tabla.

Puesto de enchufe	Módulo Original	Módulos sustitutos
0	Entrada Analógica (AI)	Simuladores de I/O, Temporizadores
1	Salida Analógica (AO)	Simuladores de I/O, Temporizadores
2	Comparadores (COMP)	Simuladores de I/O, Temporizadores
3	Contadores 500Hz (C)	Simuladores de I/O, DI, DO, COMP, AO, T
4	Entradas Digitales (DI)	Simuladores de I/O, DO, COMP, AO, C, T
5	Salidas Digitales (DO)	Simuladores de I/O, DI, COMP, AO, C, T

Tabla 3.2

Módulos que pueden usarse en sustitución de los originales

Como se puede observar los únicos módulos "universales" son los simuladores de I/O y los temporizadores (T). Esta particularidad se debe a que no poseen conexiones externas, es decir, no dependen de señales de campo para su operación, sino que se toman su alimentación directamente del bus, y realizan su función independientemente, ya sea a través las perillas frontales y diodos emisores de luz en el caso de los simuladores de entrada y salida, ó a por medio de circuitería especial en el caso de los módulos de Temporizador.

Si fuera necesaria una configuración de PLC con 8 entradas y 8 salidas digitales se podría reemplazar los módulos originales por aquellos indicados en la tabla 4.1, quedando entonces los tres últimos puestos de enchufe como muestra la Tabla 4.3.

Puesto	0	1	2	3	4	5
Tipo de tarjeta	Vacío	Vacío	Vacío	8DI x 24V	4DOx24V	4DOx24V
Dirección				I 3.0 hasta I3.7	Q 4.0 hasta Q 4.3	Q 5.0 hasta Q 5.3

Tabla 3.3

Módulos sustitutos para una configuración de 8 DI + 8 DO

En los puestos 0, 1 y 2 pueden dejarse los módulos originales, ó no conectar ninguno.

El diagrama del cableado del panel de conexiones a los elementos de bus se especifica en la figura a continuación. El número que aparece dentro de cada círculo es el tornillo correspondiente del elemento de bus de dicho puesto de enchufe. La alimentación se provee a través del tornillo 1 (para el potencial positivo de 24 VDC) y el tornillo 2 (para el potencial de referencia de 0 VDC), para todos los elementos de bus.

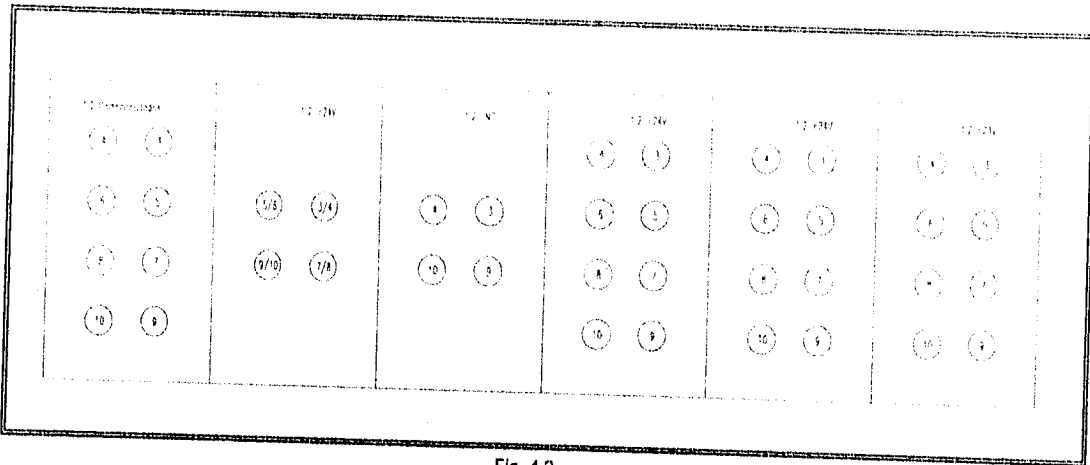


Fig. 4.2  
Correspondencia de bornes del PLC y Bananas

## III.2 PROGRAMAS DE APLICACIÓN DE LÓGICA COMBINACIONAL

### III.2.1 Generador de pulsos.

#### OBJETIVO:

Elaborar un programa generador de pulsos que pueda utilizarse como generador de base de tiempo, para apoyar otros programas.

#### DESCRIPCIÓN:

Cuando se realizan prácticas de laboratorio con un PLC, y se desea probar un programa, o simular una secuencia, se hace necesario un reloj. Este se puede proporcionar a través de una entrada digital de 24 VDC, y un generador de señales, pero implica el calibrar la amplitud, la frecuencia, el offset del aparato en cuestión.

Una forma alternativa consiste en generar el reloj internamente, y en un PLC esto es una tarea común y relativamente fácil de realizar.

#### APLICACIÓN:

El reloj se utiliza a menudo para probar salidas, generar situaciones de acuerdo a una secuencia definida y controlable con facilidad, o para limitar la duración de eventos, para probar enlaces de comunicación, y usualmente en el laboratorio, para ejecutar saltos a otros bloques de programa en forma controlada, y a la velocidad que sea más conveniente.

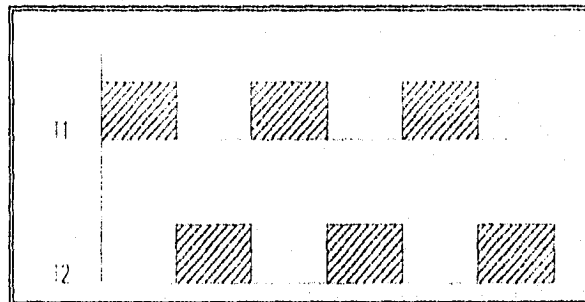


Fig. 3. 3

Diagrama de tiempos para el generador de reloj

#### PLANTEAMIENTO DEL PROBLEMA:

El método que utilizaremos para lograr un pulso de reloj utiliza dos temporizadores, que tienen una diferencia de fase de 180 grados, el direccionamiento de ellos se hace alternadamente como entrada y luego como salida, de modo que se tienen dos entradas y dos salidas (T1 y T2).

Por lo general las instrucciones que componen este reloj se colocan en el OB1, o en un bloque que es llamado incondicionalmente por aquel. Recordemos que todo depende de la complejidad de la tarea que se desea solucionar, y el uso de diferentes bloques de programa proveerá de claridad al programa terminado, de manera que será más fácil la etapa de comisión.

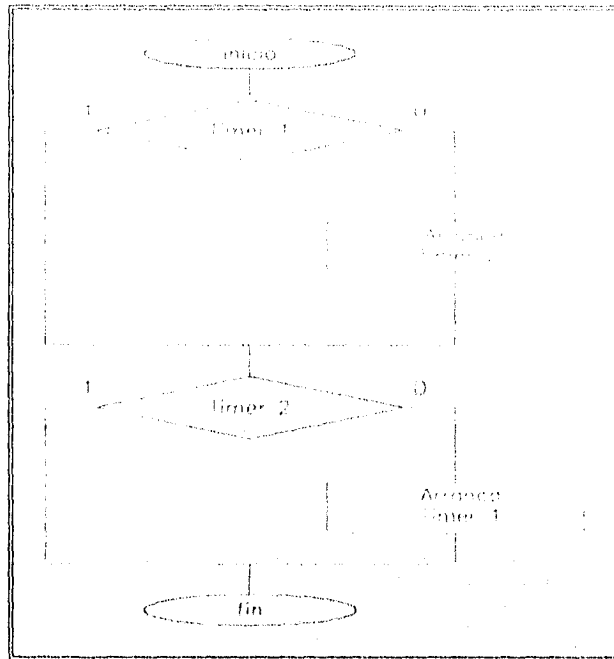


Fig. 3.4

Diagrama de flujo para el generador de tren de pulsos

### CONFIGURACIÓN DEL PLC

En el PLC no se requiere de ningún módulo de entrada o salida, sin embargo, sería conveniente colocar un simulador de salidas en el puesto de enchufe 0.

### MODO DE OPERACIÓN:

Este generador no tiene entradas externas, por lo que la señal de reloj estará presente mientras el PLC esté en modo RUN.

Con la función STATUS LIST puede observarse en el programador PG 710 la sucesión de estados lógicos de T0 y T1; así como su valor actual. El período viene a ser el doble de la constante cargada, puesto que se está generando un tren de pulsos con un ciclo de trabajo del 50%. Para modificar el valor del período, basta con cambiar la Instrucción que está en la primera línea.

Si se asigna una salida a uno de los temporizadores, se puede observar el tic en ella.

### LISTADO

El programa en STEP 5 es lo suficientemente pequeño como para ser programado en el OB 1. Este se muestra a continuación:

OB 1                                    A:TEMPO@ST.S5D                                    LEN=12

Page 1

```

Segment 1
:L KT 100.0
:AN T 1
:SE T 2
:AN T 2
:SE T 1
:BE
  
```

### III.2.2 Control de cruceo.

**OBJETIVO:**

Se busca generar la secuencia rojo, ámbar (intermitente), y verde, de manera continua de acuerdo a lo ilustrado en el siguiente diagrama de tiempos:

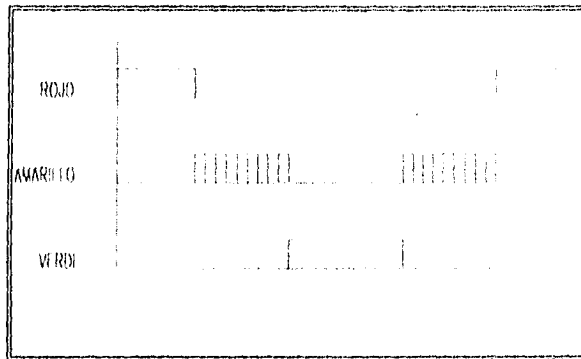


Fig. 3.5

Diagrama de tiempos en el control de cruceo

**DESCRIPCIÓN:**

Este es un ejemplo de un tipo de control en el que no hay entradas, solamente salidas, y los únicos ajustes posibles son la duración de las diferentes etapas, y del modo en que las luces se encienden y apagan. Este se haya en cruceos de una vía, y se puede visualizar rápidamente por medio del diagrama de flujo que aparece en la figura 3.5.

**APLICACIÓN:**

Este programa muestra el uso de banderas, saltos condicionados a diversos bloques de programa, el set, reset y la asignación de valores a banderas y salidas, transferencia de valores tipo bit, y en general el uso de temporizadores y de constantes para temporizador.

**PLANTEAMIENTO DEL PROBLEMA**

Primeramente se debe utilizar una base de tiempo para generar los pulsos de la luz amarilla, cuando esté activa esa etapa. Esta estará generada por dos relojes desfasados, como en el ejemplo anterior. Después, el paso de una etapa a la siguiente deberá tener principio cuando se alcance el valor prefijado de duración de dicha etapa. El final de cada etapa estará marcado por el principio de la siguiente.

Para esta solución se utiliza la combinación de 5 temporizadores, de modo que la secuencia se lleve a cabo de la manera prevista; luz roja, luz ámbar intermitente, luz verde.

Utilizando más temporizadores es posible agregar nuevas características al programa; por ejemplo, que una señal intermitente aparezca un poco antes de terminar el tiempo rojo y de la misma forma para el verde. Recordemos que el número máximo de temporizadores en una CPU 102 es de 32.

Otra manera de efectuar esta tarea consiste en implementar un Drum Sequencer, para lo cual se requiere de un apuntador y un conjunto de datos escritos en un bloque de datos. Sin embargo, esto solo es posible a partir de una CPU 103, puesto que requiere de funciones de indexación.

Veamos a continuación el diagrama de flujo de la solución propuesta:

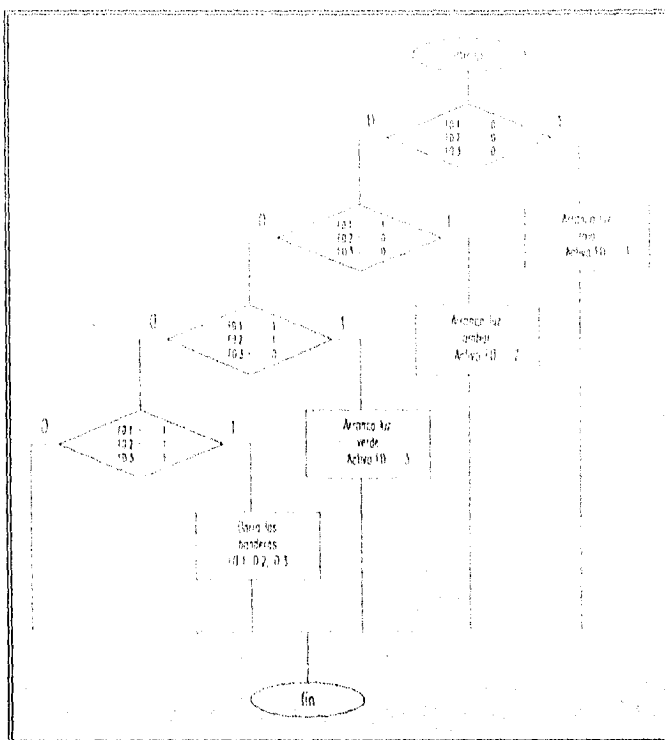


Fig. 3.6  
Diagrama de flujo de un Control de Cruce

**CONFIGURACIÓN DEL PLC**

Configurar un simulador de Entrada/Salida 788 como salida, y conectarlo en el slot 5, es decir, el más alejado de la CPU

**MODO DE OPERACIÓN:**

Este programa no tiene entradas, por lo que una vez arrancado el PLC la secuencia se sigue en el orden previsto, y sólo se puede detener pasando la CPU a STOP.

**LISTADO**

Bits de control internos

- F 0.1      Luz roja está activa
- F 0.2      Luz ámbar está activa
- F 0.3      Luz verde está activa

Salidas

- Q 5.0      Luz roja
- Q 5.1      Luz ámbar
- Q 5.2      Luz verde

Temporizadores

- T1      Temporizado de luz roja
- T2      Temporizado de luz ámbar
- T3      Temporizado de luz verde
- T4      Reloj
- T5      Reloj

Bloques de programa

OB1: Controla los saltos a subrutinas  
 PB1, PB2, PB3 arrancan las temporizaciones de las luces  
 PB4 Borra todas las banderas para así permitir el reinicio automático del ciclo  
 PB5 Transfiere los valores de los temporizadores a las salidas  
 PB6 Generador de reloj para las luces intermitentes

C:SEMAF@ST.S5D PB 1 0003 :R F 0.3  
 LEN=12 0004 :BE

PAGE 1

SEGMENT 1 0000 Semáforo  
 0000 :AN T 2  
 0001 :AN T 3  
 0002 :S F 0.1  
 0003 :L KT 300.0  
 0005 :SE T 1  
 0006 :BE

C:SEMAF@ST.S5D PB 5  
 LEN=13

PAGE 1

SEGMENT 1 0000  
 0000 :A T 3  
 0001 := Q 5.0  
 0002 :A T 1  
 0003 :A T 5  
 0004 := Q 5.1  
 0005 :A T 2  
 0006 := Q 5.2  
 0007 :BE

C:SEMAF@ST.S5D PB 2  
 LEN=12

PAGE 1

SEGMENT 1 0000  
 0000 :AN T 3  
 0001 :AN T 1  
 0002 :S F 0.2  
 0003 :L KT 100.0  
 0005 :SE T 2  
 0006 :BE

C:SEMAF@ST.S5D PB 6  
 LEN=14

PAGE 1

SEGMENT 1 0000  
 0000 :AN T 5  
 0001 :L KT 050.0  
 0003 :SE T 6  
 0004 :AN T 6  
 0005 :L KT 050.0  
 0007 :SE T 5  
 0008 :BE

C:SEMAF@ST.S5D PB 3  
 LEN=12

PAGE 1

SEGMENT 1 0000  
 0000 :AN T 1  
 0001 :AN T 2  
 0002 :S F 0.3  
 0003 :L KT 100.0  
 0005 :SE T 3  
 0006 :BE

C:SEMAF@ST.S5D OB 1  
 LEN=25

PAGE 1

SEGMENT 1 0000  
 0000 :AN F 0.1  
 0001 :AN F 0.2  
 0002 :AN F 0.3  
 0003 :JC PB 1  
 0004 :A F 0.1  
 0005 :AN F 0.2  
 0006 :AN F 0.3  
 0007 :JC PB 2  
 0008 :A F 0.1

C:SEMAF@ST.S5D PB 4  
 LEN=10

PAGE 1

SEGMENT 1 0000  
 0000 :A F 0.1  
 0001 :R F 0.1  
 0002 :R F 0.2

0009	:A F 0.2	0010	:A F 0.2
000A	:AN F 0.3	0011	:A F 0.3
000B	:JC PB 3	0012	:JC PB 4
000C	:JU PB 5	0013	:BE
000D	:JU PB 6		
000E	:AN T 2		
000F	:A F 0.1		

### III.3 CONTROL SECUENCIAL

#### III.3.1 Lavado de automóviles

##### OBJETIVO:

Este programa simula el control automático de una máquina de lavado de autos, que lavará, enjuagará, aplicará cera y secará el auto por medio de un túnel de lavado, toberas, válvulas, y un bastidor en el cual se fija el vehículo.

Para dar principio al ciclo, el operador solamente deberá colocar el auto en el bastidor, y oprimir el botón de arranque.

##### DESCRIPCIÓN:

El control de un lavado de autos es una tarea secuencial, y se pueden tener cuantos pasos sean necesarios, entre ellos el arranque normal, el re arranque tras un corte de alimentación, y el paro de emergencia.

##### APLICACIÓN

Con el presente programa se ilustra un control secuencial. El paso de una etapa a otra ocurre cuando se completan una ciertas condiciones.

##### PLANTEAMIENTO DEL PROBLEMA

Diagrama de Flujo

Etapa 1. posición inicial: el bastidor debe estar colocado en la entrada, o atrás, las puertas se abren y se despliega COLOCAR AUTO

Etapa 2. Cuando se tiene un auto en el bastidor, se espera a que se presione el botón de start.

Etapa 3. Se cierran las puertas y se apaga el letrero. Se mueve el bastidor hacia adelante, con los cepillos rotando, mientras se aplica el jabón.

Etapa 4. Se mueve el bastidor hacia atrás, con los cepillos parados, y se enjuaga con agua para remover el jabón.

Etapa 5. Se aplica la cera, moviendo el bastidor hacia adelante.

Etapa 6. Se mueve hacia atrás, todas las válvulas (toberas) estarán cerradas.

Etapa 7. En esta etapa se arranca un tiempo de espera, para la distribución de la cera.

Etapa 8. Se abre la válvula de aire, para secar el auto.

Etapa 9. Se abre la puerta, y se despliega el aviso de REMOVER AUTO

##### CONFIGURACIÓN DEL PLC

Se requieren en el PLC de 8 entradas y 8 salidas, que serán conectadas de la forma indicada en el inciso III.1.2

##### LISTADO

FB 10	C:CARWASST.S5D	LIB=2	LEN=174
		PAGE	1
SEGMENT 1	0000	"Preparar marcha programa"	



Antes de poder procesar el programa de túnel de lavado, que esta depositado en el modulo de funciones FB 10, se debe abrir el modulo de datos que se llama en el FB 10.

NAME .CAR-WASH

0005 :C DB 10 Llamada DB 10 (temp./v.contador)  
 0006 :\*\*\*

SEGMENT 2 0007 "Definir estado funcionamiento"

El programa activa al arrancar o rearmar para un ciclo la marca de impulso F10.1, que se evalúa en el seg. 4 y que puede causar el arranque de la posición inicial. El estado de servicio es representado por la marca de flanco F10.0 (flanco pos.) para "interruptor principal ON" o "nuevo rearmar". No se puede realizar un rearmar de la instalacion hasta que no se haya realizado el reset de F10.0 via "interruptor principal OFF".

0007 :O I 32.0 Interruptor "instalacion ON"  
 0008 :O F 10.7 Ident. arranque de OB 20/21/22  
 0009 :AN F 10.0 Marca de flanco positivo  
 000A := F 10.1 Marca impulso(solo 1 ciclo!)  
 000B :R F 10.7 Rearmar ident. arranque  
 000C :A F 10.1  
 000D :S F 10.0 Actualizar marca de flanco  
 000E :AN I 32.0 Ning. orden "instalacion ON"  
 000F :AN F 10.7 Ning. ident. arranque  
 0010 :R F 10.0 Rearmar marca de flanco  
 0011 :\*\*\*

SEGMENT 3 0012 "Desconectar instalacion"

Si se desactiva la instalacion o si se pulsa el interruptor de emergencia, se ponen la palabra de datos 0 y el byte de datos 2 en cero y se finaliza el programa.

0012 :A I 32.0 Interruptor "instalacion ON"  
 0013 :A I 32.1 No se pulso interruptor emerg.  
 0014 :JC =WEIT (Ramificacion programa!)  
 0015 :R C 2 Rearmar contador de paso  
 0016 :L KB 0  
 0017 :T QB 32 rearmar la sal. en el AB 32  
 0018 :T QB 33 rearmar la salida en el AB 33  
 0019 :BEU Fin de modulo  
 001A WEIT :\*\*\*

SEGMENT 4 001B "Posicion inicial"

El impulso creado durante el arranque o el re arranque en el segmento 2 puede causar el arranque de la posicion inicial. El bastidor es conducido a la posicion final trasera, se abre la puerta y se visualiza el letrero REMOVE...

001B	:AN F 10.1	"inst.ON/re arranque"
001C	:JC =WEIT	
001D	:R C 2	rearmar contador de paso
001E	:L KH 0000	
0020	:T QB 32	rearmar las salidas
0021	:T QB 33	" " "
0022	:AN I 32.5	Bastidor no en pos.final trasera
0023	:S Q 32.1	Retroceder bastidor
0024	:AN I 32.6	Puerta no esta abierta
0025	:S Q 32.2	Abrir puerta
0026	:A I 32.3	otro vehiculo en instalacion
0027	:S Q 32.5	mostrar: REMOVE AUTO
0028	WEIT :***	

SEGMENT 5 0029 "Situacion inicial"

Se comprueba si el estado de la instalacion es la "posicion inicial", en este caso aparecera el aviso "COLOCAR AUTO".

0029	:LD C 2	Estado contador paso a AKKU I
002A	:L KC 000	Requerimiento: paso 0
002C	:I=F	
002D	:AN I 32.3	Ning. vehiculo en posicion
002E	:A I 32.5	Bastidor en pos. final trasera
002F	:A I 32.6	Puerta esta abierta
0030	:S Q 32.4	mostrar: COLOCAR AUTO
0031	:R Q 32.5	desconectar: REMOVE AUTO
0032	:CU C 2	Contador de paso +1
0033	:***	

SEGMENT 6 0034 "Arrancar el lavado"

Se introduce el vehiculo que se desea lavar en la posicion correspondiente y se pulsa el boton para arrancar el procedimiento de lavado. Despues de comprobar "vehiculo en posicion" y "boton de arranque activado", se cierra la puerta y desaparece el aviso "COLOCAR AUTO".

0034	:LD C 2	Estado contador paso a AKKU I
0035	:L KC 001	Requerimiento: paso 1
0037	:I=F	
0038	:A I 33.0	Se pulso "tecla de arranque"

0039	:A I 32.3	Vehículo en posición de lavado
003A	:CU C 20	Contador procedimiento +1
003B	:LD C 20	Cantidad proc. hacia AKKU 1
003C	:T DW 2	Depositar estado contador (KF)
003D	:S Q 32.3	Cerrar puerta
003E	:R Q 32.4	desconectar: COLOCAR AUTO
003F	:CU C 2	Contador paso +1
0040	:***	

SEGMENT 7 0041 "Enjabonar"

Tras comprobar la entrada "puerta cerrada", se mueve el bastidor con los cepillos giratorios y las toberas de espuma abiertas hacia adelante.

0041	:LD C 2	Estado contador paso a AKKU 1
0042	:L KC 002	Requerimiento: paso 2
0044	:!=F	
0045	:A I 32.7	Puerta esta cerrada
0046	:R Q 32.3	desactivar "Cerrar puerta"
0047	:S Q 32.7	Enjabonar
0048	:S Q 32.6	Activar cepillos giratorios
0049	:S Q 32.0	Mover bastidor hacia adelante
004A	:CU C 2	Contador paso +1
004B	:***	

SEGMENT 8 004C "Lavar, enjuagar"

Tras comprobar la posición final delantera del bastidor: se desactiva el motor del bastidor, se cierran las toberas de espuma y vuelve a pasar el bastidor con el flujo de agua abierto hacia atrás.

004C	:LD C 2	Estado contador paso a AKKU 1
004D	:L KC 003	Requerimiento: paso 3
004F	:!=F	
0050	:A I 32.4	Bastidor esta alante
0051	:R Q 32.7	desactivar "enjabonado"
0052	:R Q 32.0	desc. "mover bast.hacia adelante"
0053	:S Q 33.0	conectar "aflujo de agua"
0054	:S Q 32.1	Bastidor hacia atrás
0055	:CU C 2	Contador paso +1
0056	:***	

SEGMENT 9 0057 "Encerar"

En la posición final trasera se desactivan el motor de bastidor y de los cepillos y se cierra el flujo de agua. A continuación se conduce el bastidor con las toberas de rociado abiertas para el lavado hacia adelante.

0057	:LD C 2	Estado contador paso a AKKU 1
0058	:L KC 004	Requerimiento: paso 4
005A	:=F	
005B	:A I 32.5	Bastidor esta atras
005C	:R Q 33.0	cerrar "aflujo de agua"
005D	:R Q 32.6	desactivar "cepillos giratorios"
005E	:R Q 32.1	Desc."mover bast. hacia atras"
005F	:S Q 33.1	Activar encerado
0060	:S Q 32.0	Bastidor hacia adelante
0061	:CU C 2	Contador paso +1
0062	:***	

SEGMENT 10 0063 "Encerado"

En la posición final delantera se cierran las toberas para el encerado y se conduce el bastidor hacia atras.

0063	:LD C 2	Estado contador paso a AKKU 1
0064	:L KC 005	Requerimiento: paso 5
0066	:=F	
0067	:A I 32.4	Bastidor esta alante
0068	:R Q 33.1	Fin "encerar"
0069	:R Q 32.0	desc."mover bast.hacia adelante"
006A	:S Q 32.1	Bastidor hacia atras
006B	:CU C 2	Contador paso +1
006C	:***	

SEGMENT 11 006D Otro "encerado"

En la posición final trasera se desactiva el motor del bastidor y se arranca el temporizador para el encerado VT. Una vez finalizado el VT se libera el próximo paso del ciclo de proceso.

006D	:LD C 2	Estado contador paso a AKKU 1
006E	:L KC 006	Requerimiento: paso 6
0070	:=F	
0071	:A I 32.5	Bastidor esta atras
0072	:L DW 4	Cargar consigna tiempo VT de DB5
0073	:SD T 20	Arrancar retard. de conexion
0074	:LD T 20	
0075	:T DW 6	Depositar tiempo real (KF)
0076	:R Q 32.1	desc. "bastidor hacia atras"
0077	:	
0078	:LD C 2	Estado contador paso a AKKU 1
0079	:L KC 006	Requerimiento: paso 6
007B	:=F	

007C :A T 20 Transcurrido tiempo VT ?  
 007D :CU C 2 Contador paso +1  
 007E :\*\*\*

SEGMENT 12 007F "Secado"

El procedimiento de secado es iniciado via arranque del tiempo de secado y apertura simultánea de la admision de aire. Una vez finalizado el tiempo de secado TT se cierra la valvula de aire y se abre la puerta.

007F :LD C 2 Estado contador paso a AKKU 1  
 0080 :L KC 007 Requerimiento: paso 7  
 0082 :!=F  
 0083 :S Q 33.2 Abrir admision de aire  
 0084 :L DW 8 Cargar consigna tiempo TT de DB5  
 0085 :SD T 22 Arrancar retard. de conexion  
 0086 :LD T 22  
 0087 :T DW 10 Depositar tiempo real (KF)  
 0088 :A Q 33.2 "Admision de aire" abierta  
 0089 :A T 22 Transcurrido tiempo TT ?  
 008A :R Q 33.2 cerrar "admision de aire"  
 008B :S Q 32.2 Abrir puerta  
 008C :CU C 2 Contador paso +1  
 008D :\*\*\*

SEGMENT 13 008E "Sacar vehiculo del túnel"

Tras abrir la puerta, se desactiva el motor de la puerta y aparece el aviso REMOVER AUTO.

008E :LD C 2 Estado contador paso a AKKU 1  
 008F :L KC 008 Requerimiento: paso 8  
 0091 :!=F  
 0092 :A I 32.6 Puerta esta abierta  
 0093 :S Q 32.5 mostrar: REMOVER AUTO  
 0094 :CU C 2 Contador paso +1  
 0095 :\*\*\*

SEGMENT 14 0096 "Túnel vacio"

Una vez que se haya sacado el vehiculo del túnel, desaparece el aviso REMOVER AUTO y se pone nuevamente el contador de paso interno en 0.

0096 :LD C 2 Estado contador paso a AKKU 1  
 0097 :L KC 009 Requerimiento: paso 9  
 0099 :!=F  
 009A :AN I 32.3 Ning.vehiculo en pos.lavado

```

009B :R Q 32.5 desconectar: REMOVE AUTO
009C :L KC 000
009E :S C 2 Cargar paso inicial 0
009F :***

```

```

SEGMENT 15 00A0 "Evaluar Retroaviso"
00A0 :A I 32.4 Bastidor está adelante
00A1 :R Q 32.0 rearmar"bastidor hacia adelante"
00A2 :A I 32.5 Bastidor esta atras
00A3 :R Q 32.1 rearmar"mover bast.hacia atras"
00A4 :A I 32.6 Puerta esta abierta
00A5 :R Q 32.2 rearmar "abrir puerta"
00A6 :A I 32.7 Puerta esta cerrada
00A7 :R Q 32.3 rearmar "cerrar puerta"
00A8 :BE

```

```

OB 1 C:CARWASST.S5D LEN=8
PAGE 1

```

```

SEGMENT 1 0000 Elaboracion ciclica de programa
0000 :JU FB 10
0001 NAME :CAR-WASH
0002 :BE

```

```

OB 20 C:CARWASST.S5D LEN=11
PAGE 1

```

```

SEGMENT 1 0000 Rearranque manual/automatico
0000 :AN F 10.7
0001 :S F 10.7 Poner ident. arranque
0002 :L KC 000
0004 :S C 2 Prefijar contador paso con 0
0005 :BE

```

```

OB 21 C:CARWASST.S5D LEN=11
PAGE 1

```

```

SEGMENT 1 0000 Rearranque manual
0000 :AN F 10.7
0001 :S F 10.7 Poner ident. arranque
0002 :L KC 000
0004 :S C 2 Prefijar contador paso con 0
0005 :BE

```

```

OB 22 C:CARWASST.S5D LEN=11
PAGE 1

```

```

SEGMENT 1 0000 Rearranque automatico

```

0000	:AN F 10.7	
0001	:S F 10.7	Poner ident. arranque
0002	:L KC 000	
0004	:S C 2	Prefijar contador paso con 0
0005	:BE	

## **III.4 MANIPULACIÓN DE DATOS NUMÉRICOS**

### **III.4.1 Simulador de Elevadores**

#### **OBJETIVO**

Este programa simula el control automático de dos elevadores, los cuales pueden atender llamadas provenientes de 8 distintos pisos.

La asignación de uno de los dos elevadores a una llamada en particular se lleva a cabo de acuerdo a los siguientes criterios:

Un elevador que se dirige hacia arriba no atenderá ninguna llamada abajo de su posición actual y viceversa.

El elevador más cercano deberá atender la llamada.

En caso de equidistancia, la prioridad es fija.

Los elevadores se quedan donde estén después de la última llamada.

Atendiendo a estos puntos, se observa que el control es dinámico puesto que depende no solo de la posición de un elevador, sino también de la tarea que esté realizando (en espera o moviéndose).

#### **DESCRIPCIÓN:**

El control de elevadores es una tarea a la cual rara vez se le presta atención. Sin embargo, el control es complejo debido a que las variables involucradas son muchas (Falla de alimentación, iluminación, ventilación, exceso de carga, apertura de puertas, asignación de destinos, atención a llamadas, acceso restringido a pisos, velocidad, etc.). Usualmente el control de elevadores se realiza por medio de controladores dedicados, y este ejemplo fue elegido sólo por la facilidad de comprensión del proceso. La característica que nos decidió a utilizar este ejemplo es la comprensión intuitiva de lo que debe hacer el programa.

#### **APLICACIONES**

El presente programa ilustra cómo una aplicación de tipo numérica (en este caso el cálculo de distancias basado en las señales provenientes de sensores digitales) puede llevarse a cabo en el PLC Simatic.

#### **PLANTEAMIENTO DEL PROBLEMA**

##### **DISEÑO MECÁNICO:**

El simulador consta de 16 diodos emisores de luz de color verde cuya función es indicar la posición de cada elevador, siendo independientes uno del otro, salvo por el control. Existen ocho interruptores normalmente abiertos que sirven para simular las llamadas desde los pisos. Estos no están conectados a ninguno de los dos elevadores, sino que se cablean directamente a las entradas digitales del PLC. Se tienen también conectores para suplir la alimentación y para proveer del cableado hacia los módulos de entrada y salida del PLC.

A continuación se muestra un esquemático de los diodos emisores de luz y de los interruptores.



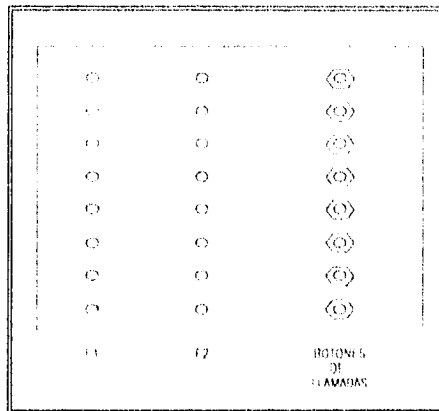


Fig. 3.7  
Esquema mecánico del simulador para elevador

Este programa sirve de ejemplo de una programación estructurada. A continuación se presenta el diagrama de flujo del programa.

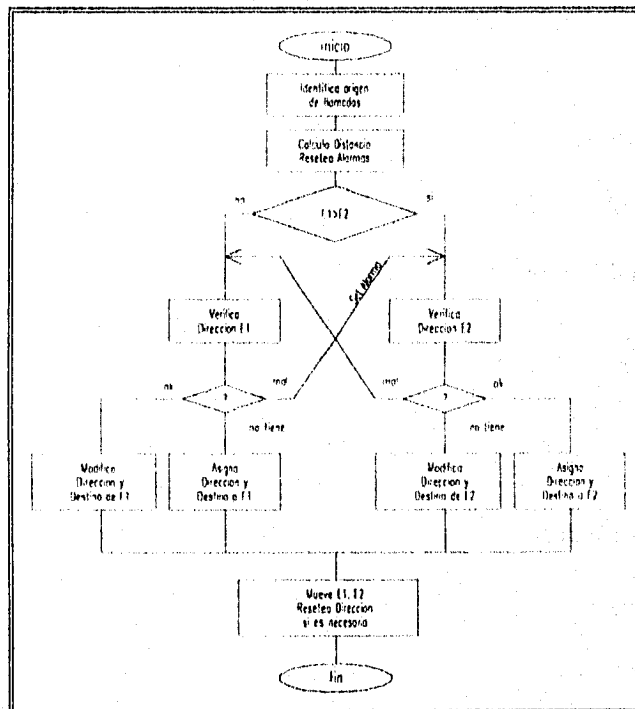


Fig. 3.8  
Diagrama de flujo para el elevador

### CONFIGURACIÓN DEL PLC

Los módulos necesarios en el PLC son los mismos que se ilustran en la sección III.1.2, y que considera que las entradas están en el Byte 4, y que las salidas se ubican en los Bytes 4 y 5. Adicionalmente se tiene una tarjeta 788 en el Byte 2.

### MODO DE OPERACIÓN

Una vez dispuesto del tablero de simulación Simatic y del simulador del elevador, simplemente se efectúan las llamadas por medio de los interruptores. La posición del elevador será indicada por los diodos emisores de luz.

### LISTADOS

El siguiente listado es generado por el programa Step 5 y muestra las sucesivas llamadas de los bloques de programa.

Program overview with DB

Page 2

```
+OB 1-+FB 1-
|
| +=FB 10-+FB 2-
| |
| | +=FB 3-+FB 4-+FB 6-
| | | |
| | | +=FB 7-
| | | |
| | | +=FB 6-
| | | |
| | | +=FB 5-+FB 8-
| | | |
| | | +=FB 9-
| | | |
| | | +=FB 8-
| | | |
| | | +=FB 4-+!F114!!
| | |
| | +=FB 5-+FB 8-
| | |
| | +=FB 9-
| | |
| | +=FB 8-
| | |
| | +=FB 4-+FB 6-
| | |
| | +=FB 7-
| | |
| | +=FB 6-
| | |
| | +=FB 5-+!F114!!
|
| +=FB 11-
|
| +=FB 12-
|
| +=PB 1-
|
| +=PB 2-
```

+OB 21-

El programa contenido en el PLC es el siguiente:

```

PB 1          A:ELEV@@ST.S5D          :JC =M002
LEN=15        :T FY 1
              :BEU
              :
Page 1        M002:L KB 3
Segment 1    :AN 1 3.2
:L KF +0     :JC =M003
:T FW 0      :T FY 1
:T FW 2      :BEU
:T FW 4      :M003:L KB 4
:T FW 6      :AN 1 3.3
:T FW 8      :JC =M004
:T FW 10     :T FY 1
:T FW 12     :BEU
:BE          :
              M004:L KB 5
PB 2          A:ELEV@@ST.S5D          :AN 1 3.4
LEN=16        :JC =M005
              :T FY 1
              :BEU
              :
Page 1        M005:L KB 6
Segment 1    :AN 1 3.5
:L FW 2      :JC =M006
:L KF +1     :T FY 1
:-F          :BEU
:T QB 4      :
:L FW 4      :M006:L KB 7
:L KF +1     :AN 1 3.6
:-F          :JC =M007
:T QB 5      :T FY 1
:BE          :BEU
              :
FB 1          A:ELEV@@ST.S5D          :M007:L KB 8
LEN=60        :AN 1 3.7
              :JC =M008
              :T FY 1
              M008:BE
              :
Page 1        :
Segment 1    :
Name :SCANIA :
              :
:L KB 0      :
:T FY 1      :
              :
:L KB 1      :
:AN 1 3.0    :
:JC =M001    :
:T FY 1      :
:BEU        :
              :
M001:L KB 2  :
:AN 1 3.1    :
              :
              FB 2          A:ELEV@@ST.S5D
              LEN=31
              :
              Page 1
              Segment 1
              Name :DISTANCI
              :
              :L FW 2
              :L FW 0
              :-F
    
```

```
:T FW 6
:A F 6.7
:JC =M001
M003 :L FW 4
:L FW 0
:>F
:T FW 8
:A F 8.7
:JC =M002
:BEU
M001 :L FW 6
:CSW
:T FW 6
:JU =M003
M002 :L FW 8
:CSW
:T FW 8
:BE
```

FB 3  
LEN=25

Page 1  
Segment 1  
Name :ELECT

```
:R F 31.4
:R F 31.3
:R F 31.1
:R F 31.0
:L FW 6
:L FW 8
:>=F
:JC FB 4
Name :CHKE2
:L FW 6
:L FW 8
:<F
:JC FB 5
Name :CHKE1
:BE
```

FB 4  
LEN=38

Page 1  
Segment 1  
Name :CHKE2

```
:A F 31.1
:BEU
:L FW 0
:L FW 4
:>F
```

```
: = F 30.2
:A F 30.2
:A F 30.6
:JC FB 6
Name :MDE2
:AN F 30.6
:AN F 30.7
:JC FB 7
Name :ADDE2
:L FW 0
:L FW 4
:<F
: = F 30.3
:A F 30.3
:A F 30.7
:JC FB 6
Name :MDE2
:AN F 31.4
: = F 31.1
:AN F 31.4
:JC FB 5
Name :CHKE1
:BE
```

FB 5  
LEN=34

Page 1  
Segment 1  
Name :CHKE1

```
:L FW 0
:L FW 2
:>F
: = F 30.4
:A F 30.4
:A F 30.0
:JC FB 8
Name :MDE1
:AN F 30.0
:AN F 30.1
:JC FB 9
Name :ADDE1
:L FW 0
:L FW 2
:<F
: = F 30.5
:A F 30.5
:A F 30.1
:JC FB 8
Name :MDE1
:AN F 31.3
:JC FB 4
Name :CHKE2
```

A:ELEV@@ST.S5D

```

BE:
FB 6          A:ELEV@@@ST.S5D      :S F 31.3
LEN=30        :A F 30.1
              :JC =M001
              :A F 30.0
Page 1        :JC =M002
Segment 1     :BEU
Name :MDE2     M001:L FW 0
              :L FW 10
              :>F
              :L FW 0
              :T FW 10
              :BEU
              M002:L FW 0
              :L FW 10
              :<F
              :BEC
              :L FW 0
              :T FW 10
              :BEU
              :BE

:S F 31.4
:A F 30.6
:JC =M001
:A F 30.7
:JC =M002
:BEU
M001:L FW 0
:L FW 12
:<F
:BEC
:L FW 0
:T FW 12
:BEU
M002:L FW 0
:L FW 12
:>F
:BEC
:L FW 0
:T FW 12
:BE

FB 7          A:ELEV@@@ST.S5D      :S F 31.3
LEN=20        :L FW 0
              :T FW 10
              :L FW 2
Page 1        :L FW 0
Segment 1     :>F
Name :ADDE2   :S F 30.1
              :<F
              :S F 30.0
              :BE

:S F 31.4
:L FW 0
:T FW 12
:L FW 4
:L FW 0
:>F
:S F 30.7
:<F
:S F 30.6
:BE

FB 8          A:ELEV@@@ST.S5D      :JU FB 2
LEN=30        Name :DISTANCI
              :JU FB 3
Page 1        Name :ELECT
Segment 1     :BE
Name :MDE1

FB 9          A:ELEV@@@ST.S5D
LEN=20
Page 1
Segment 1
Name :ADDE1

:S F 31.3
:L FW 0
:T FW 10
:L FW 2
:L FW 0
:>F
:S F 30.1
:<F
:S F 30.0
:BE

FB 10         A:ELEV@@@ST.S5D
LEN=15
Page 1
Segment 1
Name :HUBO

```

FB 11                   A:ELEV@@ST.S5D  
LEN=40

:I=F  
:R F 30.7  
:R F 30.6  
:BE

Page 1  
Segment 1  
Name :MOVER

:S F 31.7  
:A F 30.0  
:IC =M001  
:A F 30.1  
:IC =M002  
M005 :A F 30.6  
:IC =M003  
:A F 30.7  
:IC =M004  
:BEU  
M001 :L FW 2  
:L KB 1  
:F  
:T FW 2  
:JU =M005  
M002 :L FW 2  
:L KB 1  
:F  
:T FW 2  
:JU =M005  
M003 :L FW 4  
:L KB 1  
:F  
:T FW 4  
:BEU  
M004 :L FW 4  
:L KB 1  
:F  
:T FW 4  
:BE

OBI

Page 1  
Segment 1  
:JU FB 1  
Name :SCANIA  
:L KH 0000  
:L FW 0  
:><F  
:IC FB 10  
Name :HUBO

:AN T 1  
:L KT 015.1  
:SE T 2  
:AN T 2  
:L KT 015.1  
:SE T 1  
:  
:A T 1  
:AN F 31.7  
:IC FB 11  
Name :MOVER  
:  
:  
:A T 2  
:R F 31.7  
:JU FB 12  
Name :RESET  
:  
:A I 2.0  
:IC PB 1  
:  
:JU PB 2  
:BE

FB 12                   A:ELEV@@ST.S5D  
LEN=21

Page 1  
Segment 1  
Name :RESET

:L FW 10  
:L FW 2  
:I=F  
:R F 30.0  
:R F 30.1  
:L FW 12  
:L FW 4

OB 21  
LEN=10

Page 1  
Segment 1  
:L KF +8  
:T FW 2  
:T FW 4  
:BE

A:ELEV@@ST.S5D

### Listado de variables

#### Direccionamiento de Banderas por palabras

FW 0	Origen de la llamada
FW 2	Posición de E1
FW 4	Posición de E2
FW 6	Distancia absoluta de E1
FW 8	Distancia absoluta de E2
FW 10	Destino de E1
FW 12	Destino de E2

#### Direccionamiento de banderas por Bits

F 30.0	E1up
F 30.1	E1 down
F 30.2	El origen de la llamada es mayor que la posición de E2
F 30.3	El origen de la llamada es menor que la posición de E2
F 30.4	Origen es mayor que pos E1
F 30.5	Origen es menor que pos E1
F 30.6	E2 up
F 30.7	E2 down
F 31.1	Alarma. Activa por default, se desactiva si pasa por FB6 ó FB7
F 31.4	Paso por E2
F 31.7	Temporizador

#### Temporizadores:

T1, T2: Generan la base de tiempo tal como se describe en el ejemplo de aplicación # 1 de este capítulo.

#### Entradas:

I 2.0:	Borra todas las palabras.
I 30....I 37:	Llamadas desde cada piso.

#### Salidas:

Q 4.1, 4.2, 4.3:	Salidas del elevador E1
Q 5.1, 5.2, 5.3:	Salidas del elevador E2

#### Programa:

OB 1:	Control principal de flujo del programa.
OB 21:	Inicialización de variables (ver capítulo 4)
FB1:	Determinación y filtrado de llamadas.
FB 2:	Calculo de distancia absoluta de E1 y E2.
FB 3:	Determinación de elevador con menor distancia.
FB 4:	Determina si E2 está en la dirección correcta.
FB 5:	Determina si E1 está en la dirección correcta.
FB 6:	Modifica el destino del E2.
FB 7:	Asignación de dirección y destino del E2.
FB 8:	Modifica el destino de E1.
FB 9:	Asignación de dirección y destino del E1.
FB 10:	Llama a FB 2 y a FB 3.
FB 11:	Mueve ambos elevadores.
FB 12:	Verifica si E1, E2 han llegado a su destino programado.
PB 1:	Borra todas las palabras.
PB 2:	Proporciona el offset de salida.

Apéndice

## APENDICE



## APÉNDICE A

### Primera sesión con un Simatic S5

Para su primer contacto con un controlador lógico programable, es preciso tener lo siguiente:

- Un Programador PG 710 con Software STEP 5 Versión 6.3
- Un Cable de alimentación y fuente de alimentación para PG 710
- Un Cable de conexión PG -AG con número de parte 6ES5 734 2BD20
- Un Disco de alta densidad de 1.2 MB, 3.5", ya formateado
- Un PLC Simatic que consta de lo siguiente:
  - Un riel DIN
  - Fuente de alimentación PS 930
  - Unidad Central CPU 102
  - Un Elemento de bus
  - Dos simuladores de entrada digital, uno de ellos configurado como entrada y conectado en el puesto de enchufe 0 (junto a la CPU), el otro como salida y conectado en el puesto 1.

Si se tiene acceso al tablero de simulación Simatic, debe remover los módulos de entrada y salida analógica de los puestos de enchufe 0 y 1.

#### Armado del PLC

1. Se coloca en el riel la fuente de alimentación, la CPU y el elemento de bus en ese orden, tal como está ilustrado en la figura a continuación. Para hacer esto incline cada pieza, para que encaje la uña en la parte superior del riel, y después empuje hacia abajo.

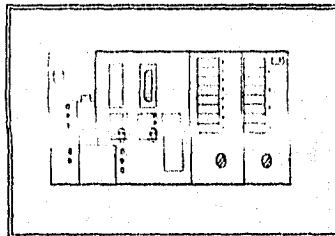


Fig. A 1

2. Se toma con cuidado el cable plano del elemento de bus y conéctelo en el receptáculo ubicado en el lado derecho de la CPU. La longitud del cable es exacta, y no debe retorcerse.

3. Se toma una tarjeta simuladora de entradas digitales. En la cara posterior, hacia la parte de arriba se encuentra un selector. Deslícelo hacia la posición I. Coloque el módulo en el elemento de bus justo al lado de la CPU. Para esto incline la tarjeta a un ángulo mayor de 15° y verifique que las protuberancias de la parte superior encajen en los huecos del elemento de bus. Empuje la parte inferior de la tarjeta hasta que quede en posición vertical. Asegure el tornillo frontal. Los tornillos de conexiones del elemento de bus pueden estar cableados o no. No hay ninguna influencia en la tarjeta, pues no tiene contacto eléctrico.

4. Se toma la otra tarjeta simuladora, y coloque el selector en la posición O. Esto configura el módulo como salida. Instálelo en el puesto libre del elemento de bus y asegure el tornillo.

5. Se conecta la fuente de alimentación a la CPU. La fuente en este punto no debe estar energizada. Vea la figura a continuación.

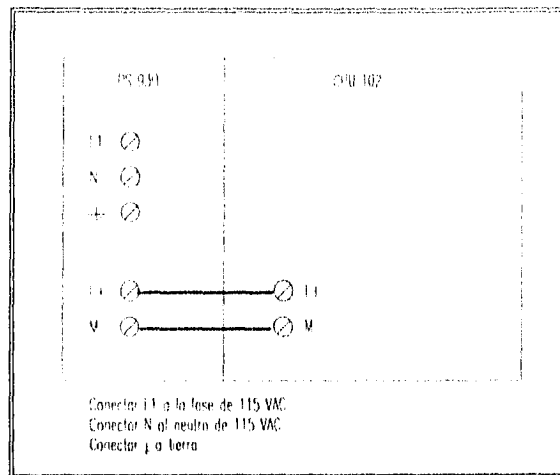


Fig. A 2

Conexiones de la fuente de alimentación al CPU y a la red

6. Se verifica que la perilla de modo de la CPU indica STOP y que la fuente tiene su perilla en la posición 0.
7. Se verifica las conexiones del PLC y la fuente.
8. Se conecta la alimentación de la fuente a la red de 115 VAC.
9. Se coloca la perilla de la fuente en la posición 1. Se debe iluminar en forma continua el diodo emisor de luz rojo de la CPU 102, lo que indica que la CPU está en STOP. Los diodos emisores de luz de los módulos simuladores deben indicar que el módulo junto a la CPU es una entrada, y que el siguiente está configurado como salida.

**Conexiones del Programador al CPU**

10. Se energiza el PG 710 siguiendo estrictamente los siguientes pasos

Se conecta la fuente al PG 710, usando el conector DIN

Se conecta el cable de alimentación de la fuente a la red de 115 VAC

Se abre el PG deslizando los broches frontales en sentidos opuestos, hacia afuera

Se enciende el PG 710 con la perilla ubicada en el costado izquierdo. No debe haber ningún disco en el drive.

El PG debe presentar una pantalla con tres opciones. Seleccione F1.

En el caso de que el PG entre directamente a DOS, arranque el programa tecleando "S5" y presionando RETURN. En la selección de lenguaje indique "E".

11. Se conecta ahora el PG con el CPU a través del cable 734 2BD20, de acuerdo a la siguiente figura.

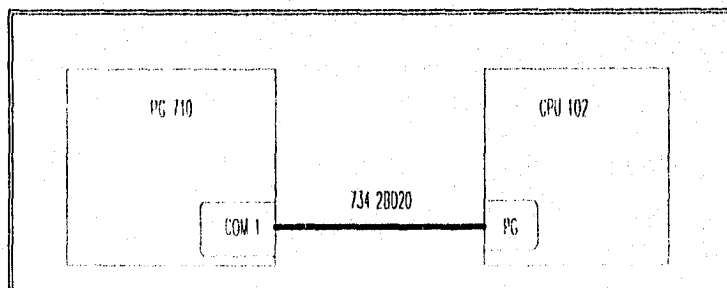


Fig. 4.6

Instalación del cable de comunicación PG-AG

El puerto *Com 1* se encuentra en la parte posterior del PG, y el puerto *Pg* está ubicado en la parte frontal de la CPU Simatic 102 U.

### Configuración del programa STEP 5

12. Se inserta el disquete en el drive que está ubicado del lado izquierdo del PG.

13. Se supone que el programador está presentado una pantalla de STEP 5.

Presione F4. En esta pantalla se da de alta el nombre y la unidad de disco en la que se guardarán los archivos de esta sesión. Con la ayuda de las flechas llegue al renglón que indica directorio. Seleccione la Página 1, presionando F1 en caso necesario. Presione F3. Presione F3 otra vez. Con la tecla de tabulación llegue hasta donde se indica la unidad A, y presione entonces la tecla 0 del bloque de números. Esta tecla se conoce como Aceptación. De aquí en adelante se supone que usted presiona esa tecla cada vez que su entrada es la correcta, en lugar del RETURN que se utiliza en MS-DOS. Con la ayuda de las flechas seleccione la unidad A: . Acepte.

14. Con la ayuda de las flechas ubíquese en la línea de archivo de trabajo. Presione F3. Teclee el nombre del archivo "prueba", luego Aceptación. El sistema automáticamente pone las terminaciones ST.S5D

15. En la línea de representación seleccione (con F3) el modo STL

16. En la barra de herramientas de la parte inferior de la hoja se halla la forma de pasar a la página 2. Esta es probablemente la tecla F1 ó F4. Presione la tecla correcta.

17. Vaya al PLC y coloque el selector de modo hacia la posición RUN. El diodo emisor de luz rojo debe apagar, y encender el diodo emisor de luz verde. Si no es así, pase a stop y repita la operación nuevamente.

18. Regrese al PG 710. En la pantalla debe seleccionar el modo de operación ONLINE. Haga esto con las flechas y la tecla F3. El programador emitirá un mensaje de confirmación y pasará a modo online.

19. Presione F8.

Con esto ha configurado el programa para que guarde el programa en el disco que ha sido colocado en la unidad A:

### Carga del programa en el PLC

20. Presione F1. Se despliega en la pantalla una caja de interacción. Teclee el nombre del segmento que se va a programar, en este caso "FB 1". Acepte.

21. Se despliega ahora en pantalla el formato para la programación de un bloque de funciones. El cursor pide un nombre. Teclee "prueba", RETURN, RETURN. En este ejemplo no se declaran variables. Escriba lo siguiente:

L IB 0

T QB 1

Para cambiar de línea utilice la tecla de RETURN.

Presione la tecla de aceptación cuatro veces. El programa regresará a la pantalla original.

22. Presione F1. Programaremos ahora el OB1, por lo cual escribirá en el renglón "OB 1".

23. La pantalla muestra el formato de un bloque de organización. En este no se da un nombre. Teclee lo siguiente:

JU FB 1

Al presionar RETURN el programa agregará la siguiente línea:

NAME: PRUEBA

Presione ahora aceptación 4 veces.

Ahora tiene usted en su disco un programa completo de STEP 5.

### Carga del programa en la CPU

24. Presione F5. Con la ayuda del tabulador vaya a la caja que indica el bloque o los bloques que desea escribir en el PLC. Seleccione con las flechas la caja que indica "All Blocks". Acepte. Si Ud. eligió que confirmara la sobrescritura, deberá confirmar cada bloque con la tecla de aceptación.

Presione ESC. Coloque ahora el programador como fuera de línea. Presione F4, y después F3 en la caja de modo de operación.

Al haber terminado esta secuencia de 24 pasos se tiene un programa corriendo en el PLC. Veamos que hace este programa de prueba.

**Prueba del programa**

25. Mueva todas las perillas del módulo simulador de entradas a la posición 0. Ahora active y desactive una a una las perillas del simulador de salidas. Verifique que esto no tiene ningún efecto en los diodos emisores de luz de salida. Haga lo mismo con las perillas del simulador de entradas. Verifique que los mismos diodos emisores de luz que se iluminan en el simulador de entradas lo hacen en el simulador de salidas. Esto se debe a que el programa está copiando el contenido del simulador de entradas en el de salidas.

26. Ponga el PLC en STOP. Active las perillas del simulador de entradas. Verifique que no se copia el estado en las salidas.

Con esto hemos finalizado su primera sesión con un PLC Simatic 100-U

## CONCLUSIONES

---

## CONCLUSIONES

### 1. Futuro de la lógica ó control por cableado.

La lógica Cableada como estrategia de control seguirá teniendo un porcentaje del mercado debido principalmente a su bajo costo y escaso servicio requerido, para las aplicaciones de la gama más baja (por ejemplo, en línea blanca y electrodomésticos). Además, siempre estará presente como medio para de proveer de medidas de seguridad como son el paro y bloqueo de dispositivos (motores), en la forma de botones de paro de emergencia, y candados de seguridad.

Sin embargo, para el campo de aplicaciones de 8 entradas / 6 salidas, cada vez será mayor la presencia de micro-plc , que hoy abundan en el mercado. Estos aparatos, con sus capacidades de control y visualización integradas ó posibles, marcarán la pauta para las aplicaciones de la gama más baja, en donde el papel más importante ó en ocasiones el factor decisivo. será las tunciones de comunicación de los micro PLC.

### 2. Influencia del lenguaje de programación

El disponer de un único lenguaje facilita la difusión comercial del producto Simatic, puesto que no es necesario aprender varios paquetes de programación para pasar desde un modelo pequeño hacia uno más grande.

Los mnemónicos utilizados son de fácil aprendizaje, y las reglas de utilización también lo son. En todo caso, cualquier error de sintaxis es inmediatamente advertido por el intérprete Step 5, el cual, sin embargo, no sugiere la solución. Básicamente se puede decir que solamente existe la *and*, la *or*, el *load* y el *transfer*, y con esas cuatro instrucciones se hace casi el 80 % de cualquier programa.

Es precisamente el leguaje de programación lo que suele ser una traba para PLC de otras marcas, en los cuales las familias micro, mini y grande poseen cada uno un lenguaje diferente, aún cuando los diferentes departamentos de ingeniería los proyectan bajo una misma filosofía de programación.

### 3. El control por medio de PLC requiere de personal más capacitado, lo cual es una desventaja para la modernización de máquinas y sistemas, puesto que existe una inercia para no aceptar la entrada de tecnologías diferentes por parte del personal que no está capacitado en ellas. De la misma forma, egresados de las facultades deberán tener conocimiento de las técnicas en uso por las empresas en las que desean emplearse.

### 4. Con el paso del tiempo el costo del hardware tiende a disminuir, mientras que el software ( p. ej. también el Know-How) va en aumento. En el futuro, el precio total de un proyecto de automatización se verá fuertemente influenciado por la ingeniería, en proporción al relativamente bajo costo de los elementos de control. Esta tendencia beneficiará a los integradores de sistemas, es decir, a los ingenieros.

A este respecto, la aplicación del método conlleva varias ganancias. Por ejemplo, la documentación del proyecto es mucho más fácil puesto que esta se genera en las primeras etapas. Otra ventaja es que en los protocolos generados en la primera etapa (análisis) se basará el desarrollo de las demás, limitando de esta forma las tareas y responsabilidades del programador y de cada uno de los participantes. El método permite conocer desde temprano la cantidad de personal requerido, y la repartición de las tareas en grupos, todo lo cual favorece el trabajo en equipo. Por ejemplo, se facilitará mucho la programación de un equipo PLC extenso cuando este se hace en equipo. En fin, a pesar de que pareciera más burocrático el estar emitiendo una gran cantidad de documentación, el progreso que se deriva de esto es mayor que aquel que puede suponerse de tener todo el hardware atornillado en su lugar.

### 5. Un aspecto que se debe tener siempre presente en las Universidades es la capacitación de sus alumnos en la tecnología de PLC's ya que con el paso del tiempo estos sistemas de están manejando más comúnmente en la industria y están apareciendo otros con el mismo principio pero

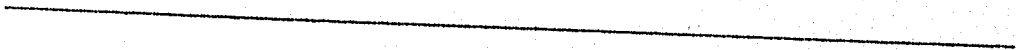
## Conclusiones

---

de mayores capacidades. El desarrollo tecnológico deberá tomarse en cuenta para el perfil de los egresados de carreras de ingenierías.

Actualmente se ha visto una mayor difusión de los PLC en los centros de estudios tecnológicos, tales como Cecatis, y Conaleps, y también en el Instituto Politécnico Nacional, en relación con la difusión que se le da en la UNAM, tendencia que nos pone en desventaja. La Facultad de Ingeniería no debe tener por objetivo crear expertos en automatización, pero sí brindar a sus alumnos medios para entender tecnologías aplicables (y extensamente aplicadas) en México. Es nuestra esperanza que este trabajo logre facilitar la difusión de los PLC en nuestra facultad.

## BIBLIOGRAFÍA





## **BIBLIOGRAFÍA**

1. Programmable Controlers, Concepts and Applications,  
C.T.Jones & L.A. Bryan  
Publicado por International Programmable Controls Inc.  
Estados Unidos, 1988

2. Automating With The Simatic S5-115U  
Hans Berger  
Publicado por Siemens Aktiengesellschaft  
Berlin/Munich, Alemania, 1992

3. Simatic S5-90/95U System Manual  
Publicado por Siemens Aktiengesellschaft  
Berlin/Munich, Alemania, 1994

4. Simatic S5-100U System Manual  
Publicado por Siemens Aktiengesellschaft  
Berlin/Munich, Alemania, 1989

5. Industrial Solid State Electronics, Devices and Systems  
Timothy J. Maloney  
Publicado por Prentice Hall Inc.  
New Jersey, Estados Unidos, 1986

6. Programmable Controlers, An engineer's Guide  
E. A. Parr  
Publicado por Newnes (Butterworth-Heinemann LTD)  
Oxford, Inglaterra, 1993

7. The anatomy of an automation Project  
Vance J. Vandoren  
Artículo aparecido en la revista Control Engineering.  
Estados Unidos, Diciembre de 1995,