

Capítulo 4

Controlador Lógico

Programable PLC

La ingeniería de control se ha desarrollado a través del tiempo. En el pasado los humanos eran el método para controlar los sistemas. En tiempos recientes la electricidad se ha utilizado para el control, los primeros controles eléctricos fueron los relevadores. Los relevadores permiten encender o apagar un circuito eléctrico sin necesidad de un interruptor mecánico. Fue inventado por Joseph Henry en 1835[6].

4.1 Historia

Los Controladores Lógico Programables (PLC) fueron inventados como respuesta a las necesidades de la industria automotriz. Inicialmente fueron adoptados por las empresas para sustituir la lógica cableada. En 1968 GM Hydramatic (la división de transmisiones automáticas de General Motors) emitió una solicitud para realizar una propuesta que sustituyera la lógica cableada. La propuesta ganadora fue realizada por Bedford Associates. El primer PLC fue el MODICON 084 [6]

4.2 Definición y principios de operación

Un controlador lógico programable (Programmable Logic Controller PLC) figura 4.1 es un dispositivo operado digitalmente, que usa una memoria para el almacenamiento interno de instrucciones con el fin de implementar funciones específicas, tales como lógica, secuenciación, registro y control de tiempos, conteo y operaciones aritméticas, para controlar a través de entradas/salidas digitales o analógicas, varios tipos de máquinas o procesos [1].



Figura 4.1 PLC Siemens

Los PLC's operan de manera secuencial y cíclica, es decir, una vez finalizado el recorrido completo de un programa, comienza a ejecutar su primera instrucción.

Los elementos que contiene un PLC son:

- Unidad Central de proceso
- Módulos de entrada
- Módulos de salida
- Fuente de Alimentación
- Dispositivos periféricos
- Interfaces

La unidad central es el “cerebro” del PLC. Este toma las decisiones relacionadas al control de la máquina o proceso. Durante su operación, el CPU recibe entradas de diferentes dispositivos de sensado, ejecuta decisiones lógicas, basadas en un programa almacenado en la memoria, y controla los dispositivos de salida de acuerdo al resultado de la lógica programada.

Los módulos de entradas y salidas son la sección del PLC en donde sensores y actuadores son conectados y a través de los cuales el PLC monitorea y controla el proceso.

La fuente de alimentación convierte altos voltajes de corriente de línea (115V 230V CA) a bajos voltajes (5V, 15V, 24V CD) requeridos por el CPU y los módulos de entradas y salidas.

El funcionamiento del PLC es un continuo ciclo cerrado, primero el sistema operativo inicia la vigilancia de tiempo de ciclo, después el CPU escribe los valores de imagen de proceso de las salidas en los módulos de salida, a continuación la CPU lee el estado de las entradas en los módulos de entrada y actualiza la imagen de proceso de las entradas, el CPU procesa el programa del usuario en segmentos de tiempo y ejecuta las operaciones indicadas en el programa, al final de un ciclo el sistema realiza las tareas pendientes por ejemplo carga y borrado de bloques.

Los PLC's han ganado popularidad en las industrias y probablemente continuarán predominando por algún tiempo, debido a las ventajas que ofrecen [6]:

- Son un gasto efectivo para controlar sistemas complejos
- Son flexibles y pueden ser aplicados para controlar otros sistemas de manera rápida y fácil.
- Su capacidad computacional permite diseñar controles mas complejos
- La ayuda para resolver problemas permite programar fácilmente y reduce el tiempo de inactividad del proceso.
- Sus componentes confiables hacen posible que pueda operar varios años sin fallas.
- Capacidad de entradas y salidas
- Monitoreo
- Velocidad de operación

- Están diseñados para trabajar en condiciones severas como: vibraciones, campos magnéticos, humedad, temperaturas extremas.

4.3 Programación STEP 7

Step 7 es el software estándar de SIMATIC para crear programas PLC en lenguaje KOP (Esquema de contactos), FUP (Diagrama de funciones) o AWL (Lista de instrucciones) para los equipos SIMATIC S7 [2].

El control difuso que se diseñó para controlar la máquina de enderezado de chasis se programó en lenguaje KOP y a continuación se explicará algunos de los conceptos básicos de este lenguaje de programación, para tener una mayor comprensión del programa desarrollado.

La representación del lenguaje de programación gráfico KOP (esquema de contactos) es similar a la de los esquemas de circuitos. Los elementos de un esquema de circuitos, tales como los contactos normalmente cerrados y normalmente abiertos, se agrupan en segmentos. Uno o varios segmentos constituyen el área de instrucciones de un bloque lógico. Las operaciones que realiza el PLC pueden clasificarse en operaciones con bits, comparación, conversión, aritméticas, de transferencia, temporización, contaje, etc.

Cada segmento de un esquema de contactos (KOP) puede contener instrucciones de entrada y salida figura. 4.2. Las instrucciones de entrada realizan una comparación o prueba

y coloca el estado del segmento dependiendo del resultado. Las instrucciones de salida examinan el estado del segmento y ejecuta alguna operación o función.

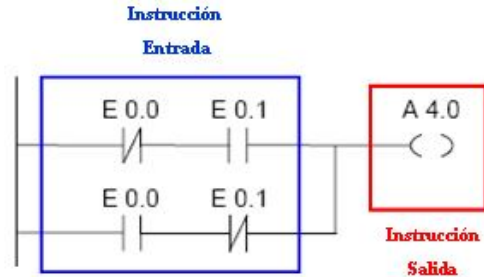


Figura 4.2 Ejemplo de segmento

A continuación se describirán algunos de los elementos mas utilizados en el lenguaje KOP.

4.3.1 Contacto normalmente abierto ---| |---

Se cierra si el valor del bit consultado, que se almacena en el operando indicado es "1". Si el contacto está cerrado, la corriente fluye a través del contacto, el resultado lógico es "1". De lo contrario si el estado de la señal del operando es "0" el contacto está abierto, no hay flujo de corriente y el resultado lógico de la operación es "0".

4.3.2 Contacto normalmente cerrado ---| / |---

Se abre si el valor del bit consultado, que se almacena en el operando indicado, es "0". Si el contacto está cerrado, la corriente fluye a través del contacto y el resultado lógico (RLO) es "1". De lo contrario, si el estado de señal en el operando indicado es "1", el contacto está abierto. Si el contacto está abierto no hay flujo de corriente y el resultado lógico de la operación es "0".

4.3.3 Bobina de relé ---()

Opera como una bobina en un esquema de circuitos. Si la corriente fluye hasta la bobina, el bit en el operando se pone a "1". Si no fluye corriente hasta la bobina, el bit en el operando se pone a "0". Una bobina de salida sólo puede colocarse dentro de un esquema de contactos en el extremo derecho de un circuito.

4.3.4 Contadores

Los contadores usados en un PLC tienen la misma función que un contador mecánico (figura 4.3). Los contadores comparan un valor acumulado y un valor preestablecido para la función de circuito de control. Los contadores pueden ser utilizados para inicializar una operación cuando se alcanza una cuenta o esperar la realización de una operación hasta que se alcanza la cuenta.

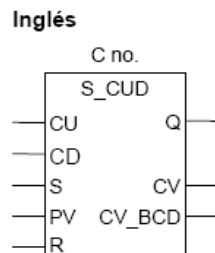


Figura 4.3 Bloque contador en inglés

Existen tres diferentes tipos de contadores: los contadores que incrementan, los contadores que decrementan y los que pueden incrementar y decrementar.

A continuación se muestra (tabla 4.1) los parámetros que contienen los contadores y su descripción [2]:

Tabla 4.1 Parámetros de los contadores

Parámetro Inglés	Parámetro Alemán	Tipo de datos	Área de memoria	Descripción
N.º de C	N.º de Z	COUNTER	Z	Número de identificación del contador, el área varía según CPU utilizada
CU	ZV	BOOL	E,A,M,L,D	Entrada de conteo adelante
CD	ZR	BOOL	E,A,M,L,D	Entrada de conteo atrás
S	S	BOOL	E,A,M,L,D	Entrada para predeterminedar el contador
PV	ZW	WORD	E,A,M,L,D o constante	Valor numérico introducido en forma de C#<valor> en el margen comprendido entre 0 y 999
PV	ZW	WORD	E,A,M,L,D	Valor para iniciar el contador
R	R	BOOL	E,A,M,L,D	Entrada de puesta a 0
CV	DUAL	WORD	E,A,M,L,D	Valor actual del contador, número hexadecimal
CV_BCD	DEZ	WORD	E,A,M,L,D	Valor actual del contador, número BCD
Q	Q	BOOL	E,A,M,L,D	Estado del contador

4.3.5 Temporizadores

Un temporizador (figura 4.4) es un aparato mediante el cual, podemos regular la conexión o desconexión de un circuito eléctrico pasado un tiempo desde que se le dio dicha orden.

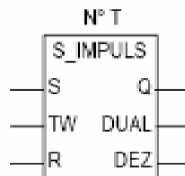


Figura 4.4 Bloque temporizador

Existen diferentes tipos de temporizadores, su funcionamiento se describe en la tabla 4.2 [2]:

Tabla 4.2 Tipos de temporizador

Temporizadores	Descripción
S_IMPULS Temporizador de impulso	El tiempo máximo que la señal de salida permanece a 1 corresponde al valor de temporización t programado. La señal de salida permanece a 1 durante un tiempo inferior si la señal de entrada cambia a 0.
S_VIMP Temporizador de impulso prolongado	La señal de salida permanece a 1 durante el tiempo programado, independientemente del tiempo en que la señal de entrada esté a 1.
S_EVERZ Temporizador de retardo a la conexión	La señal de salida es 1 solamente si ha finalizado el tiempo programado y la señal de entrada sigue siendo 1.
S_SEVERZ Temporizador de retardo a la conexión con memoria	La señal de salida cambia de 0 a 1 solamente si ha finalizado el tiempo programado, independientemente del tiempo en que la señal de salida esté a 1.
S_AVERZ Temporizador de retardo a la desconexión	La señal de salida es 1 cuando la señal de entrada es 1 o cuando el temporizador está en marcha. El temporizador arranca cuando la señal de entrada cambia de 1 a 0.

4.3.6 Funciones (FC)

Las funciones son bloques programables. Una función es un bloque lógico "sin memoria". Las variables temporales de las FC se memorizan en la pila de datos locales. Estos datos se pierden tras el tratamiento de las FC. Para fines de memorización de datos, las funciones pueden utilizar bloques de datos globales. Como una FC no tiene asignada ninguna memoria, se han de indicar siempre parámetros actuales. A los datos locales de una FC no se pueden asignar valores iniciales.

4.3.7 Bloques de Función (FB)

Los bloques de función son bloques programables. Un FB es un bloque "con memoria". Dispone de un bloque de datos asignado como memoria (bloque de datos de instancia DB). Los parámetros que se transfieren al FB, así como las variables estáticas, se memorizan en el bloque de datos de instancia. Las variables temporales se memorizan en la pila de datos locales. Los datos memorizados en el DB de instancia no se pierden al concluir el tratamiento del FB.

4.4 Programa sistema de enderezado de chasis en frío

El programa diseñado puede dividirse en 3 secciones (figura 4.5), la primera realiza una comparación de las entradas de los sensores para poder obtener las variables de entrada antes mencionadas AB, CD, MN-OP, la segunda parte implementa el controlador difuso diseñado en el capítulo 2 y la última parte realiza todas las acciones de los actuadores del sistema hidráulico.

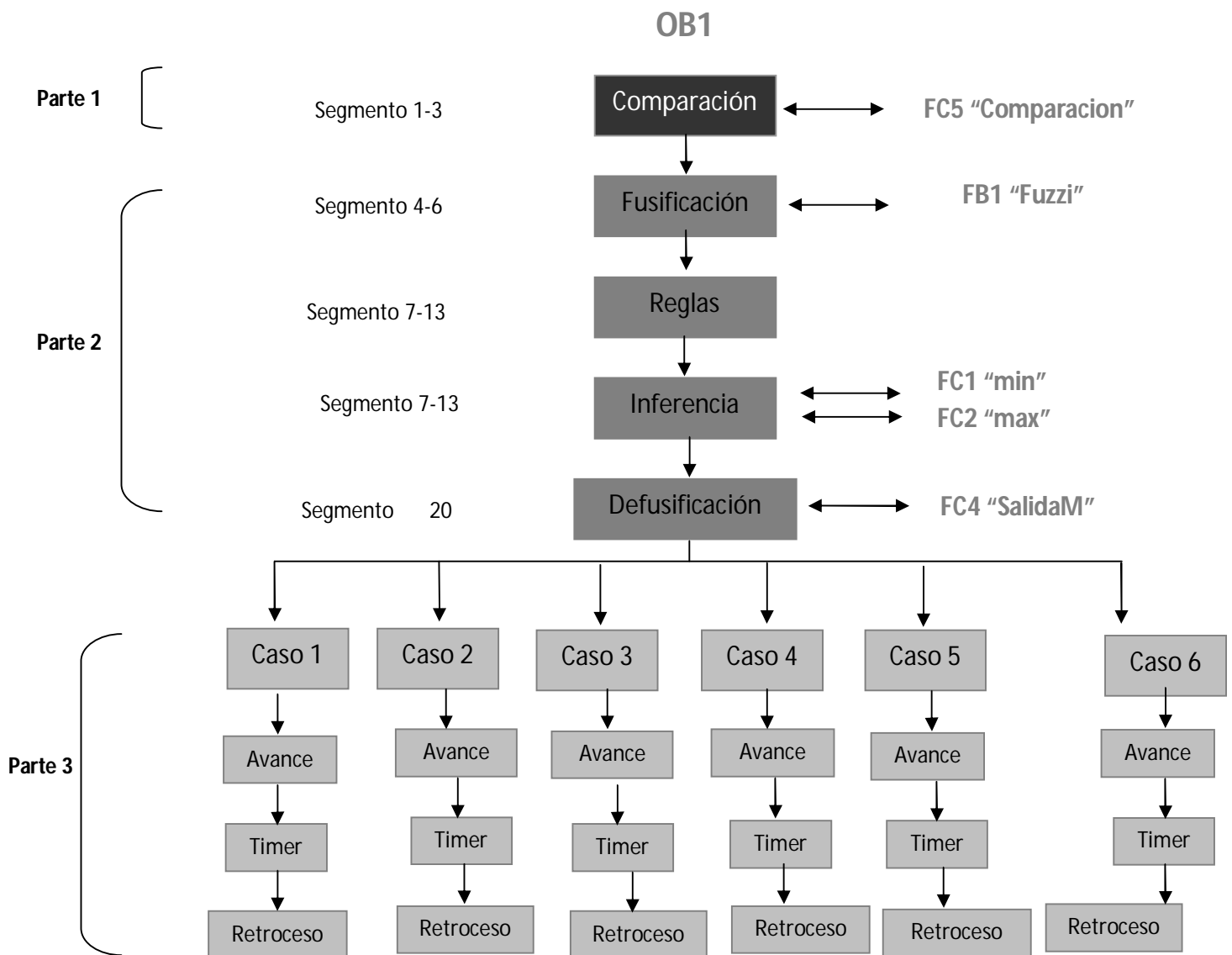


Figura 4.5 Diagrama a bloques programa sistema de enderezado de chasis en frío

El programa esta formado de un programa principal OB1 que contiene 40 segmentos, como se mencionó anteriormente los primeros segmentos 1-3 se utilizaron para realizar la comparación de los valores de entrada de los sensores para obtener las variables de entrada que utiliza el controlador difuso utilizando una función “Comparación” la cual recibe dos valores de entrada y da como salida la resta de estos valores estos valores son guardados en las marcas MD 97 para AB, MD 101 para CD y MD105 para MN-OP

De los segmentos 4-6 se llama al bloque de función “Fuzzi” que es el encargado de calcular los valores de pertenencia de las entradas en cada uno de los conjuntos. Recibe como entrada la variable (AB, CD, MN-OP) y entrega como salida el grado de pertenencia de la variable y la parte del trapecio de los conjuntos en donde se encuentra la variable de entrada ya que esta información nos servirá para poder activar las reglas que tengan relación con ese conjunto, sin importar en que parte del trapecio se encuentren.

Si tomamos como ejemplo el segmento 4 (figura 4.5) donde se calcula el grado de pertenencia de la variable AB se tienen tres bloques en donde se llama a la función “Fuzzi” el primer bloque calcula el valor de pertenencia en el conjunto $A > B$, tiene como entrada la marca M97 esta marca contiene el valor de la resta entre el sensor A y B, después de realizar las operaciones la función entrega en MD4 el grado de pertenencia y así lo calcula también para $A = B$ y $A < B$. Es importante mencionar que cada bloque de función “Fuzzi” tiene asignado un bloque de datos DB en estos se encuentran definidos los límites de cada conjunto, el programa completo se puede observar en el Apéndice A.

La base de conocimiento esta definido en los segmentos 7-13, en estos se definió las reglas del control, en el segmento 7 del Apéndice A se muestra la regla 1, cada conjunto de marcas (M0.1, M0.2, M0.3) representa el conjunto de la variable $A < B$, y estas son conectadas en paralelo con las demás variables (M1.5, M1.6, M1.7) para $C = D$ y (M2.3, M2.4, M2.5) para $MN < OP$, cada marca representa un segmento del trapecoide de cada conjunto, se encuentran conectadas de esta manera para que sin importar en que parte del conjunto se encuentre la variable de entrada, la regla se active si se encuentra en ese conjunto. Además de las reglas en cada segmento se comienza con la inferencia se hace una llamada a la función “min”, esta función tiene como entradas los valores de pertenencia correspondientes a las variables que conforman las reglas y da como salida el valor mínimo de activación de la regla.

En los segmentos 14-19 se continúa con la inferencia y se obtiene el máximo de los mínimos que activan cada regla. En el segmento 20 se realiza de defusificación, para esto se programó la función “Salida M” en la cual se calcula el valor máximo de las salida, el resultado de la defusificación es guardado en la memoria MD113.

Teniendo la salida del control, el siguiente paso es realizar las acciones necesarias para reparar el chasis dependiendo del caso en el que haya clasificado. Primero se realiza una comparación de MD113 la salida, para identificar en que caso se encuentra, se enciende la bomba y se manda la señal a las electroválvulas para que avancen los cilindros, cuando llega la señal de los sensores de presencia indicando que la estructura se encuentra alineada en cada uno de los lados de la estructura se desactiva la señal de la válvula y esta

regresa a la posición inicial, cuando todos las partes se encuentran alineadas, se apaga la bomba y se manda la señal a la electroválvula para hacer retroceder los cilindros.

En este capítulo se presentó introducción a los Controladores Lógico Programables PLC, sus características y se presentaron los elementos básicos para poder programarlos. También se explicó como funciona y como esta compuesto el programa del control difuso que se diseñó en el Capítulo 3. El programa completo se puede ver en el Apéndice A.