

**ESCUELA POLITÉCNICA DEL EJÉRCITO
SEDE – LATACUNGA**



**CARRERA DE INGENIERÍA
ELECTRÓNICA E INSTRUMENTACIÓN**

**PROYECTO DE GRADO PARA LA OBTENCIÓN
DEL TÍTULO DE INGENIERO DE EJECUCIÓN EN
ELECTRÓNICA E INSTRUMENTACIÓN**

**“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA
CONTADOR PARA DIFERENTES PROCESOS DE
PRODUCCION DE LA EMPRESA TEXTILES RIO
BLANCO”**

JAVIER ENRIQUE MOLINA MOLINA

LATACUNGA, OCTUBRE DEL 2006

CERTIFICACIÓN

Certificamos que el presente documento fue elaborado por el señor:

Javier Enrique Molina Molina

Bajo nuestra dirección, como requisito para la obtención del título de Ingeniero de Ejecución en Electrónica Especialidad Instrumentación.

Ing. Franklin Silva
DIRECTOR DEL PROYECTO

Ing. Galo Ávila
CODIRECTOR DEL PROYECTO

DEDICATORIA

Dedico este trabajo
a mis Padres, a mis Hermanos
y a la memoria de mi Abuelita Dolores.

AGRADECIMIENTO

Agradezco a Dios, a mis Padres por todo su Sacrificio, por su apoyo moral y económico, también debo agradecer a mis familiares y amigos que de una u otra manera me apoyaron para dar un paso más en el largo camino del vivir.

Un agradecimiento especial a la Empresa TEXTILES RIO BLANCO, representada por el Señor Ingeniero Mauricio Pérez de Anda, quien ha sabido brindar su sincero apoyo a toda la juventud estudiosa del centro del país.

Un sincero y especial agradecimiento al Señor Ingeniero Edwin Molina, quién con su asesoría y desinteresada colaboración fue posible la realización de este proyecto.

CONTENIDO

INTRODUCCIÓN	X
CAPÍTULO I	
ESTUDIO DEL PLC SIEMENS S7-200.	1
1.1. Características del PLC.....	1
1.1.1. Introducción al PLC SIEMENS Simatic S7-200.....	1
1.1.2. Equipos necesarios.....	2
1.2. Estudio del hardware.....	3
1.2.1. Características y capacidad del CPU utilizado en el proyecto.....	3
1.2.1.1. Memoria.....	3
1.2.1.2. Entradas y salidas físicas.....	3
1.2.1.3. Entradas y salidas totales.....	4
1.2.1.4. Operaciones.....	4
1.2.1.5. Funciones adicionales.....	4
1.2.1.6. Comunicación.....	4
1.2.2. Principales componentes de un PLC S7-200.....	5
1.2.2.1. CPU S7-200.....	5
1.2.2.2. Componentes de un CPU S7-200.....	6
1.2.2.3. Configuraciones máximas de E/S.....	7
1.2.2.4. Instalación y configuración del sistema S7-200.....	8
1.2.3. Nociones básicas para programar un CPU S7-200.....	9
1.2.3.1. Especificar las unidades funcionales.....	9
1.2.3.2. Diseñar los circuitos de seguridad de cableados.....	9
1.2.3.3. Definir la estaciones de operación.....	10
1.2.3.4. Realizar planos de configuración.....	11
1.2.3.5. Elaborar una lista de nombres simbólicos.....	11
1.2.3.6. Referencias a las entradas y salidas en el programa.....	11
1.2.4. Visualizador de textos TD-200.....	12
1.3. Estudio de software.....	14
1.3.1. Software recomendado.....	14
1.3.1.1. Instalación STEP 7-Micro/WIN 32.....	15

1.3.1.2. Software STEP 7-Micro-WIN.....	17
1.3.1.2.1. Visualizar el estado del programa en AWL.....	19
1.3.1.2.2. Visualizar el estado del programa en KOP.....	19
1.3.1.2.3. Visualizar el estado del programa en FUP.....	19
1.3.1.3. Configurar la comunicación utilizando el cable PC/PPI.....	20
1.3.1.4. Conectar el PC al CPU.....	20
1.3.1.5. Verificación de parámetros estándar del interfase de comunicación.....	21
1.3.1.6. Establecer comunicación con el CPU.....	23
1.3.1.7. Fallas de comunicación.....	24
1.3.1.8. Cambio de parámetros de comunicación del CPU.....	25
1.3.2. Operaciones del PLC S7-200.....	26
1.3.2.1. Operaciones lógicas con bits (SIMATIC).....	26
1.3.2.2. Operaciones de comparación (SIMATIC).....	32
1.3.2.3. Operaciones de temporización (SIMATIC).....	35
1.3.2.4. Operaciones con contadores.....	39
1.3.2.4.1. Contadores rápidos (alta velocidad).....	41
1.3.2.5. Operaciones aritméticas con enteros (SIMATIC).....	57
1.3.2.6. Operaciones aritméticas con números reales (SIMATIC).....	65
1.3.2.7. Operaciones con funciones numéricas (SIMATIC).....	67
1.3.2.8. Operaciones de transferencia (SIMATIC).....	70
1.3.2.9. Operaciones de tabla (SIMATIC).....	73
1.3.2.10. Operaciones lógicas (SIMATIC).....	76
1.3.2.11. Operaciones de desplazamiento y rotación (SIMATIC).....	79
1.3.2.12. Operaciones de conversión (SIMATIC).....	85
1.3.2.13. Operaciones de control del programa (SIMATIC).....	93
1.3.2.14. Operaciones de interrupción y comunicación (SIMATIC).....	96

CAPÍTULO II

ESTUDIO DE LAS MÁQUINAS Y DETERMINACIÓN DE PARÁMETROS 101

2.1. Estudio de la Máquina Carda C4 Rieter 84.....	101
2.1.1. Descripción de la Carda.....	101
2.2. Estudio de la máquina peinadora E75, E76.....	103
2.2.1. Descripción de la peinadora.....	103
2.3. Estudio de la máquina Hila G51.....	106
2.3.1. Descripción de la Hila.....	106

2.4.	Determinación de parámetros de las máquinas.....	108
------	--	-----

CAPÍTULO III

DESARROLLO DEL SISTEMA **109**

3.1.	Diseño del software a implementarse.....	109
3.1.1.	Diseño de tacómetro, visualización de parámetros de entrega.....	109
3.1.2.	Selección de turno y operaciones de conteo de material entregado.....	116
3.1.3.	Configuración de subrutinas.....	124
3.1.4.	Configuración de visualizador TD-200.....	130
3.2.	Diseño de hardware.....	131
3.2.1.	Fuente de alimentación.....	132
3.2.2.	Conexiones de entradas y salidas del PLC.....	132
3.2.3.	Diseño de tarjeta electrónica.....	133
3.3.	Pruebas de funcionamiento.....	136
3.3.1.	Tacómetro.....	136
3.3.2.	Selección de turno.....	137
3.4.	Manual de usuario.....	139
3.4.1.	Identificación de partes.....	139
3.4.2.	Selección de turno.....	140
3.4.3.	Visualización de RPM y velocidad de entrega.....	141
3.4.4.	Programación para el metraje de bote.....	142
3.4.5.	Reiniciación de conteo de bote.....	144
3.4.6.	Puesta a cero de los contadores totalizadores.....	145
3.4.7.	Estado de fuente.....	146

CAPÍTULO IV

VENTAJAS Y DESVENTAJAS DEL NUEVO SISTEMA **147**

4.1.	Ventajas.....	147
4.2.	Desventajas.....	148

CAPITULO V	
CONCLUSIONES Y RECOMENDACIONES	149
5.1 Conclusiones.....	149
5.2 Recomendaciones.....	151
BIBLIOGRAFÍA	153
GLOSARIO	154
ANEXOS	155
A. Conexiones de PLC y Tarjeta electrónica.....	155
B. Conexión de terminales de contador EZA-1.....	156
C. Diagramas del programa realizado.....	157

INTRODUCCIÓN

En la actualidad la electrónica avanza en gigantes pasos, lo que ha permitido que el hombre minimice espacio físico y obtenga mayor funcionalidad de nuevos equipos con lo que se obtiene un mejor desempeño y rendimiento en maquinarias de producción, en consecuencia a este avance, surge la necesidad de innovar equipos con mejor funcionalidad de los que actualmente se dispone dentro de la empresa Textiles Río Blanco, a lo que se refiere a sus contadores de producción que se utilizan en distintas máquinas en cada una de las fases de hilatura.

La empresa Textiles Río Blanco produce hilo de fibra de algodón, para obtener este producto final, el algodón pasa por diferentes procesos, empezando por la apertura, en donde se libera el algodón de toda clase de impurezas, después pasa por las cardas en donde se obtiene mechas de algodón, para obtener una mejor calidad de las fibras se las somete a un proceso de peinado, regresa a las cardas, pasa a las mecheras en donde se realiza la torsión y templado de las mechas para obtener unas de menor diámetro, seguido a la fase de hilatura en donde se aplica otro proceso de torsión y templado de las mechas de menor diámetro, pasa a la fase de enconado, para obtener un producto final listo para ser comercializado.

La empresa Textiles Río Blanco en sus diferentes procesos de producción, utiliza contadores electrónicos originales de la firma textil RIETER, los cuales dan la posibilidad de apreciar diferentes parámetros de producción en todo el proceso de hilatura tales como velocidad de entrada, velocidad de salida, torsión, metros producidos por turnos, sean estos de 1 a 4 turnos diarios.

Es importante tener en cuenta el costo de estos módulos contadores, la tecnología electrónica con la que fueron construidos está discontinuada y en poco tiempo no existirían repuestos, de existir el costo de mano de obra por reparación y repuestos serían excesivamente costosos.

Es por eso que se ha diseñado y construido un contador de producción para las máquinas de hilatura con las mismas funciones y datos de entrega que el contador de la marca RIETER serie EZA1 y a las necesidades de operación. Además este contador se podrá adaptar a cualquier tipo de maquinaria donde se requiera tener datos de producción, sea en unidades m/min, y de estas unidades tener un consolidado, mediante la aplicación de factores textiles de cada máquina. Este contador de producción (Rieter) es utilizado en la empresa Textiles Río Blanco en las siguientes máquinas: Cardas RIETER C4, peinadoras E7/5, Peinadoras E7/6, hilas G5/1a, para el correcto funcionamiento del contador en las máquinas indicadas se debe cambiar los parámetros de programación en base a las condiciones de operación de cada una de ellas.

En el capítulo I se encuentra la descripción del PLC S7-200, la misma que explica el hardware y componentes, también se describe el software y operaciones que puede realizar el PLC.

El capítulo II contiene la descripción y funcionalidad de las máquinas como son la Carda, Peinadora e Hila, en las cuales se va utilizar este proyecto.

El capítulo III contiene los diseños realizados en hardware y software, además de la explicación de cada parte del programa cargado en el PLC, también se encuentra el manual de usuario.

El capítulo IV consta de las comparativas entre el nuevo sistema contador y el sistema contador original, deduciendo ventajas y desventajas del nuevo sistema.

En el capítulo V se encuentran las conclusiones y recomendaciones obtenidas durante la ejecución de éste proyecto.

CAPITULO I

ESTUDIO DEL PLC SIEMENS S7-200

1.1 CARACTERÍSTICAS DEL PLC.

1.1.1 Introducción al PLC SIEMENS Simatic S7-200

La serie S7-200 comprende diversos sistemas pequeños de automatización (Micro-PLC's) que se pueden utilizar para numerosas tareas, de acuerdo a la aplicación o el campo que se requiere utilizarlo, ya que tiene una gran variedad de operaciones, facilidad de comunicación, conexión con módulos de expansión, módulos de señales analógicas que facilita un proceso de automatización, en la figura 1.1 se muestra un PLC S7-200, los mismos son diseñados para numerosas aplicaciones de control, además los diversos tamaños y fuentes de alimentación de las CPU's, se ajustan a la necesidad de un sistema de control, estas ventajas son esenciales en los actuales diseños, ya que hoy en día se busca reducir espacio, optimizar aplicaciones y más que todo sistemas de confiabilidad y fácil operación para el usuario.

Entre las varias operaciones que dispone el PLC, podemos decir que es capaz de realizar operaciones matemáticas en coma fija, en coma flotante, operaciones de comparación, dispone además de temporizadores, contadores ascendentes, descendentes, también puede operar a nivel de byte, palabra, doble palabra, entero o en número real, entre otras aplicaciones podemos programar contadores rápidos (Alta Velocidad), sistemas PID.

Hoy en día en el sector industrial del centro de nuestro país se está innovando conforme la tecnología avanza en grandes pasos, por lo que es motivo de investigación y desarrollo de varios sistemas que facilitan el proceder humano y técnico.

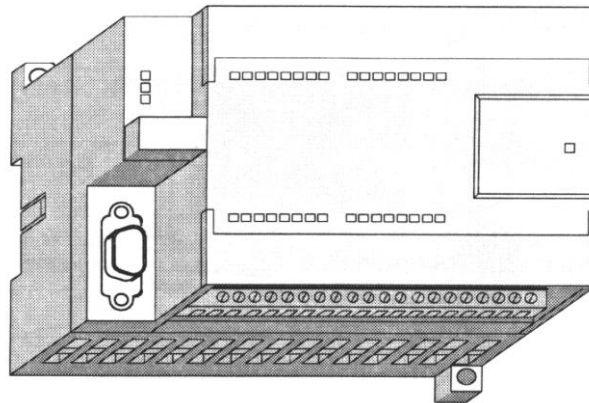


Figura 1.1 PLC Simatic S7-200

1.1.2 Equipos necesarios.

La figura 1.2 muestra los componentes básicos de un sistema para el PLC S7-200, incluye un CPU S7-200, un computador PC, el software de programación STEP 7 Micro/WIN 32 (versión 3.1) y un cable de comunicación (PC/PPI).

Si desea utilizar un PC, debe disponer de uno de los siguientes equipos adicionales:

- Un cable PC/PPI.
- Un procesador de comunicaciones (CP) y un cable de interfase multipunto (MPI).
- Una tarjeta de interfase multipunto (MPI). El cable de comunicación se suministra junto con la tarjeta MPI.

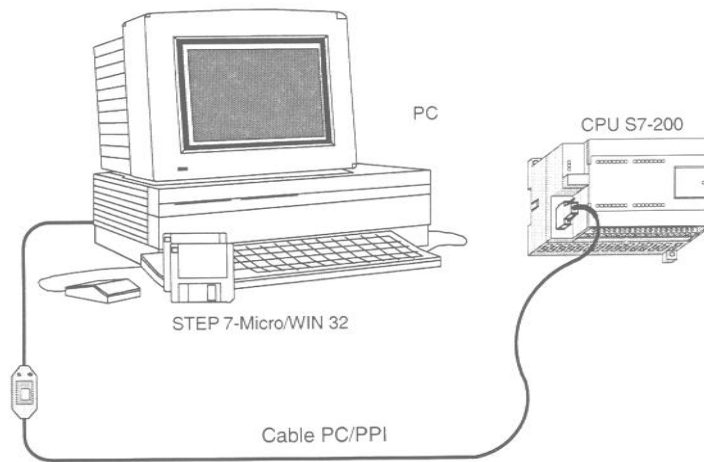


Figura 1.2 Componentes básicos de un PLC S7-200

1.2 ESTUDIO DEL HARDWARE.

1.2.1 Características y capacidad del CPU utilizado en el proyecto.

Para realizar este proyecto se utilizó un CPU 224, que pertenece a la familia de los Micro PLC S7-200, la misma que nos ofrece las funciones que se describe a continuación.

1.2.1.1 Memoria.

Tiene una capacidad de 4096 palabras para programa, en datos de usuario se dispone de 2560 palabras, todos estos datos se manejan a través de una memoria de tipo EEPROM, además dispone de un respaldo (condensador de alto rendimiento) de 190h.

1.2.1.2 Entradas y Salidas Físicas

El micro PLC S7-200, CPU 224 dispone de 14 entradas y 10 salidas, además se puede agregar siete módulos de ampliación.

1.2.1.3 Entradas y Salidas Totales.

La cantidad real de entradas y salidas que se puede contar con el CPU se puede ver limitada por el tamaño de la imagen del proceso, la cantidad de módulos de ampliación, la corriente de 5V y la cantidad de entradas/salidas físicas de cada componente.

Para el caso del CPU 224, el tamaño de la imagen de entradas/salidas digitales es de 256 (128E / 128S), el tamaño de la imagen de entradas/salidas analógicas es de 32E / 32S.

1.2.1.4 Operaciones.

La velocidad de ejecución booleana a 33MHz es de 0.37 μ S/operación, posee una imagen de proceso de 128 entradas y de 128 salidas, tiene 256 relés internos, además dispone de 256 contadores, 256 temporizadores; la capacidad de palabra de entrada como palabra de salida son de 32/32 respectivamente; también dispone de 256 relés de control secuencial; tiene la capacidad de trabajar con bucles FOR/NEXT, aritmética en coma fija y aritmética en coma flotante (suma, resta, multiplicación y división).

1.2.1.5 Funciones Adicionales.

Dentro de las funciones adicionales se tiene contadores rápidos 6 H/W (20 KHz), 2 potenciómetros analógicos; además 2 salidas de impulsos (20 KHz, sólo DC), dispone de interrupciones para comunicación, 1 para transmisión y 2 para recepción, también tenemos 2 interrupciones temporizadas (1mS a 250mS), dispone además entradas de interrupción de hardware (4 filtros de entrada), tiene un reloj de tiempo real y además posee protección con contraseña.

1.2.1.6 Comunicación.

Para comunicaciones el CPU 224 dispone de un puerto RS-485, los protocolos que soporta son PPI, DP/T, Freeport.

Además soporta PROFIBUS punto a punto (NETR/NETW).

1.2.2 Principales Componentes de un PLC S7-200.

Un PLC S7-200 puede comprender un CPU S7-200 solo o conectado a diversos módulos de ampliación opcionales.

1.2.2.1 CPU S7-200.

El CPU S7-200 es un equipo autónomo compacto que incorpora una unidad central de procesamiento (CPU), una fuente de alimentación, así como entradas y salidas digitales.

- El CPU ejecuta el programa y almacena los datos para la tarea de Automatización o el proceso.

- El sistema se controla mediante entradas y salidas digitales (E/S). Las entradas vigilan las señales de los dispositivos de campo, por ejemplo sensores e interruptores, mientras que las salidas supervisan las bombas, motores u otros aparatos del proceso.
- La fuente de alimentación suministra corriente a la CPU y a los módulos de ampliación conectados.
- Los puertos de comunicación permiten conectar la CPU a una unidad de programación o a otros dispositivos que intervengan en el proceso.
- Los diodos luminosos indican el modo de operación del CPU (RUN o STOP), el estado de las entradas y salidas físicas, así como los posibles fallos del sistema que se hayan detectado.
- Utilizando módulos de ampliación se pueden agregar entradas y salidas (E/S) adicionales a la CPU.
- El rendimiento de la comunicación se puede incrementar utilizando módulos de ampliación.
- Algunos CPU's tienen un reloj de tiempo real incorporado, en tanto que otras pueden disponer de un cartucho (opcional) de reloj de tiempo real.
- Un cartucho enchufable EEPROM en serie (opcional) sirve para almacenar programas del CPU y transferir programas de una CPU a otra.
- Un cartucho enchufable de pila (opcional) permite prolongar el respaldo de los datos en la RAM.

1.2.2.2 Componentes de un CPU S7-200.

En la siguiente figura se indica la identificación de los componentes de alimentación, señalización, comunicación, etc. de un CPU S7-200.

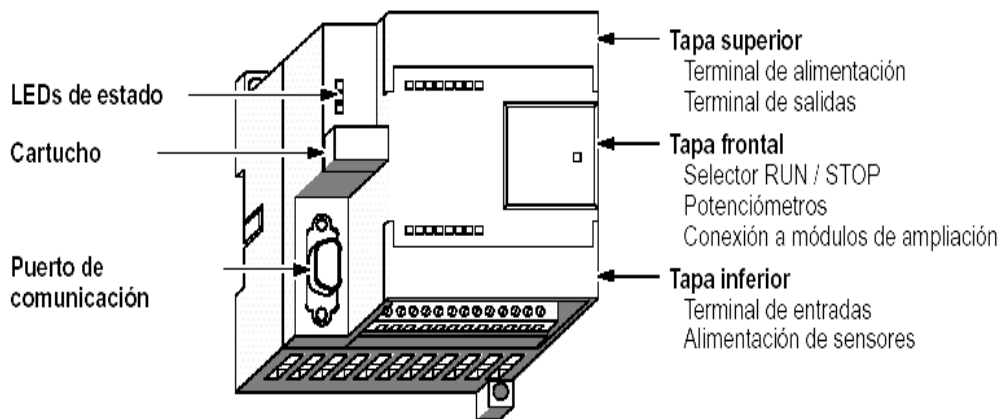


Figura 1.3 Identificación de CPU S7-200.

En la figura 1.4 se muestra un CPU S7-200 conectado con un módulo de ampliación para incrementar su número de entradas y salidas.

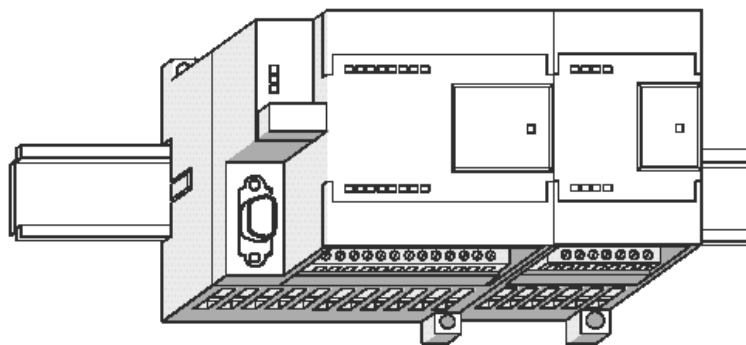


Figura 1.4. CPU S7-200 conectado con un modulo de ampliación.

1.2.2.3 Configuraciones máximas de E/S.

La configuración máxima de E/S para cada CPU está sujeta a los siguientes límites:

- Cantidad de módulos de ampliación:

El CPU 224 soporta 7 módulos de ampliación como máximo, de estos 7 módulos, sólo 2 pueden ser módulos de ampliación inteligentes (EM 277 PROFIBUS-DP).

- Tamaño de la imagen de las E/S digitales: La memoria imagen de todas las CPU's S7-200 permite manejar 128 entradas digitales y 128 salidas digitales. Algunas E/S físicas no se pueden gestionar en la memoria imagen, puesto que el espacio está dispuesto en bloques de 8 E/S. Puede suceder que un módulo determinado no pueda gestionar por completo un bloque de 8 E/S. Por ejemplo, la CPU 224 tiene 10

salidas físicas, pero necesita 16 salidas de la memoria imagen. Un módulo de 4 entradas y 4 salidas físicas requiere 8 entradas y 8 salidas de la memoria imagen.

- Tamaño de la imagen de las E/S analógicas: A continuación se indica la memoria imagen prevista para las E/S analógicas:

El CPU 224 dispone de 32 entradas y 32 salidas.

- **Corriente de 5 V:** La tabla 1.1 muestra la corriente de 5 V máxima que puede suministrar cada una de las CPU's. La corriente total de todos los módulos de ampliación integrados en el sistema no puede exceder el límite mencionado.

CPU 22X		Módulo de ampliación	
Corriente DC 5V para los módulos de ampliación (mA)		Consumo de corriente DC 5V (mA)	
CPU 222 CPU 224 CPU 226	340mA 660mA 1000 mA	EM 221, 8 entradas digitales x DC 24 V	30
		EM 222, 8 salidas digitales x DC 24 V	50
		EM 222, 8 salidas digitales x relé	40
		EM 223, 4 entradas digitales/4 salidas digitales x DC 24 V	40
		EM 223, 4 entradas digitales/4 salidas de relé x DC 24 V	40
		EM 223, 8 entradas digitales/8 salidas digitales x DC 24 V	80
		EM 223, 8 entradas digitales/8 salidas de relé x DC 24 V	80
		EM 223, 16 entradas digitales/16 salidas digitales x DC 24 V	160
		EM 223, 16 entradas digitales/16 salidas de relé x DC 24 V	150
		EM 231, 4 entradas analógicas x 12 bits	20
		EM 231, 4 entradas analógicas x termopar	60
		EM 231, 4 entradas analógicas x RTD	60
		EM 232, 2 salidas analógicas x 12 bits	20
		EM 235, 4 entradas analógicas/4 salidas analógicas x 12 bits	30
		EM 277 PROFIBUS-DP	150

Tabla 1.1 Corriente suministrada por la CPU S7-200

1.2.2.4 Instalación y configuración del sistema S7-200.

Para la instalación se deben tener en cuenta los siguientes criterios:

- El sistema operativo utilizado (Windows 95, Windows 98, o Windows NT 4.0).
- El tipo de hardware utilizado, por ejemplo:

- PC con cable PC/PPI.
 - PC o unidad de programación SIMATIC con procesador de comunicaciones (CP).
 - CPU 221, CPU 222, CPU 224, CPU 226.
 - MODEM.
- La velocidad de transferencia utilizada.

1.2.3 Nociones básicas para programar un CPU S7-200.

1.2.3.1 Especificar las unidades funcionales.

Hay que tomar en cuenta las funciones de cada sección del proceso o de la instalación. Analizando los siguientes aspectos:

- Entradas y salidas (E/S).
- Descripción del funcionamiento de la operación.
- Condiciones de habilitación es decir, los estados que se deben alcanzar antes de ejecutar una función de cada actuador (electroválvulas, motores, accionamientos, etc).
- Descripción del interfase de operador.
- Interfaces con otras secciones del proceso o de la instalación.

1.2.3.2 Diseñar los circuitos de seguridad cableados.

Hay que determinar si los aparatos requieren un cableado permanente por motivos de seguridad. Si fallan los sistemas de automatización, puede ocurrir un arranque inesperado o un cambio en el funcionamiento de las máquinas. En este caso, pueden producirse heridas graves o daños materiales, por lo tanto, es preciso utilizar dispositivos de protección contra sobrecargas electromecánicas que funcionen independientemente del CPU, a fin de evitar las condiciones inseguras.

Para diseñar los circuitos de seguridad cableados se deben tomar en cuenta los siguientes aspectos:

- Definir el funcionamiento erróneo o inesperado de los actuadores que pudieran causar peligros.
- Determinar las condiciones que garanticen un funcionamiento seguro y determinar cómo detectar dichas condiciones, independientemente del CPU.

- Definir cómo el CPU y los módulos de ampliación deberán influir en el proceso cuando se conecte y desconecte la alimentación, así como al detectarse errores. Estas informaciones se deberán utilizar únicamente para diseñar el funcionamiento normal y el funcionamiento anormal esperado, sin poderse aplicar para fines de seguridad.
- Se debe disponer de dispositivos de parada de emergencia manual o de protección contra sobrecargas electromagnéticas que impidan un funcionamiento peligroso, independientemente del CPU.
- Desde los circuitos independientes, se debe dar informaciones de estado apropiados al CPU, para que el programa y los interfaces de operador dispongan de los datos necesarios.
- Definir otros requisitos adicionales de seguridad para que el proceso se lleve a cabo de forma segura y fiable.

1.2.3.3 Definir las estaciones de operación.

Conforme a las funciones exigidas, genere planos de las estaciones de operación incorporando los siguientes puntos:

- Panorámica de la ubicación de todas las estaciones de operador con respecto al proceso o a la instalación.
- Disposición mecánica de los aparatos (pantalla, interruptores, lámparas, etc.) de la estación de operador.
- Esquemas eléctricos con las correspondientes entradas y salidas del CPU o de los módulos de ampliación.

1.2.3.4 Realizar planos de configuración.

Conforme a las funciones exigidas, realice planos de configuración del sistema de automatización incorporando los siguientes puntos:

- Ubicación de todas las CPU's y de todos los módulos de ampliación con respecto al proceso o a la instalación.
- Disposición mecánica de las CPU's y de los módulos de ampliación (incluyendo armarios, etc.).
- Esquemas eléctricos de todas las CPU's y de los módulos de ampliación (incluyendo los números de referencia, las direcciones de comunicación y las direcciones de las entradas y salidas).

1.2.3.5 Elaborar una lista de nombres simbólicos.

Si desea utilizar nombres simbólicos para el direccionamiento, elabore una lista de nombres simbólicos para las direcciones absolutas. Incluya no sólo las entradas y salidas físicas, sino también todos los demás elementos que utilizará en su programa.

1.2.3.6 Referencias a las entradas y salidas en el programa.

El funcionamiento básico del CPU S7-200 es muy sencillo:

- El CPU lee el estado de las entradas.
- El programa almacenado en el CPU utiliza dichas entradas para evaluar la lógica. Durante la ejecución del programa, el CPU actualiza los datos.
- El CPU escribe los datos en las salidas.

La figura 1.5 muestra cómo se procesa un esquema de circuitos simple en una CPU S7-200. En este ejemplo, el estado del interruptor de la estación de operador para abrir la electroválvula de vaciado se suma a los estados de otras entradas. El resultado obtenido establece el estado de la salida que corresponde a dicha electroválvula.

La CPU procesa el programa cíclicamente, leyendo y escribiendo los datos.

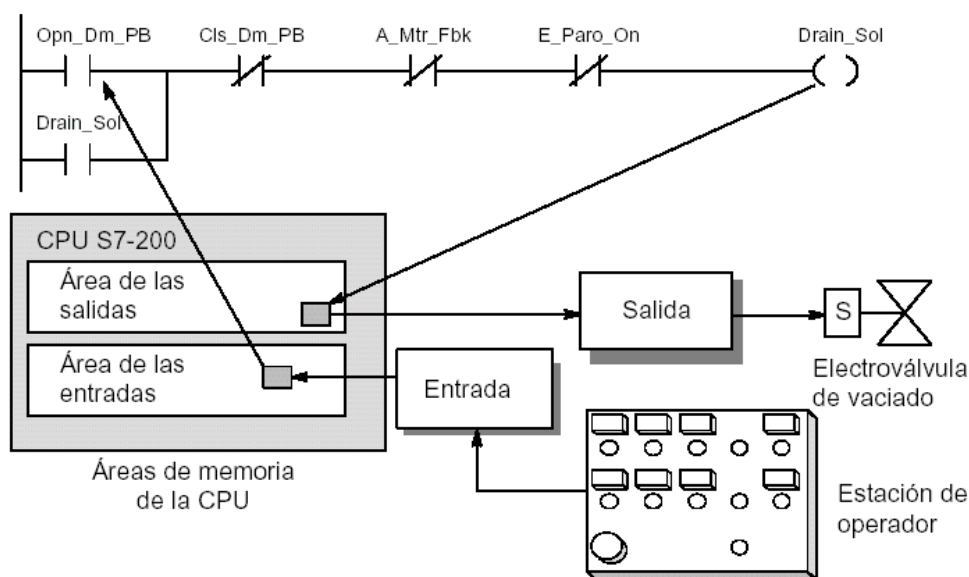


Figura 1.5 Referencias de entradas y salidas según un programa.

1.2.4 Visualizador de textos TD-200.

El TD 200 es un visualizador de textos y un interfase de operador para la gama de sistemas de automatización S7-200.

Con el TD 200 se pueden ejecutar las siguientes funciones:

- Visualizar mensajes leídos del CPU S7-200.
- Ajustar determinadas variables de programa.
- Forzar/desforzar entradas y salidas (E/S).
- Ajustar la hora y la fecha de los CPU's que incorporen un reloj de tiempo real.
- Utilizar menús e indicadores en seis idiomas (inglés, alemán, francés, español, italiano y chino).
- Utilizar diversos juegos de caracteres que soportan los idiomas de Europa Occidental, Eslavos y chinos.

El TD 200 es un pequeño equipo compacto que incorpora todos los componentes necesarios para manejar y observar el CPU S7-200. La figura 1.6 muestra los principales componentes del TD 200 que se describen en la tabla 1.2

El TD 200 es alimentado desde el CPU S7-200 a través del cable TD/CPU o desde una fuente de alimentación independiente, funciona como un maestro de red cuando se conecta a una o más CPU's S7-200, también se puede utilizar con otros maestros en una red. Es posible utilizar diversos TD 200s con una o varias CPU's S7-200 conectadas a una misma red.

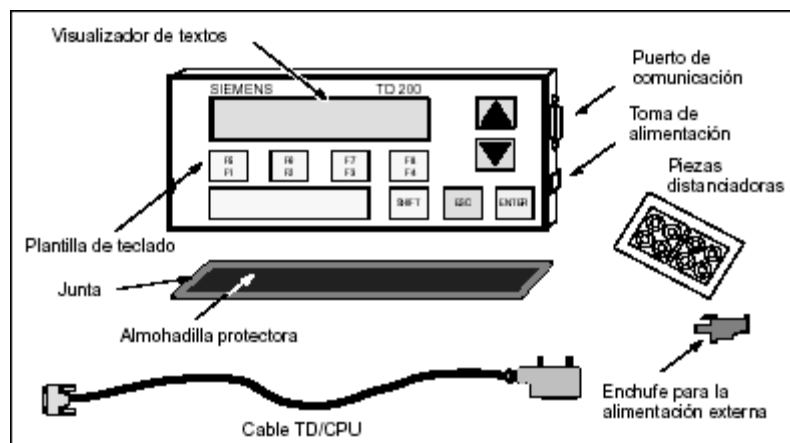


Figura 1.6 Componentes del TD-200.

Componente	Descripción
Visualizador de textos	El visualizador de textos consiste en un display de cristal líquido (LCD) con retroiluminación y una resolución de 33 x 181 píxeles para visualizar los mensajes recibidos de la CPU S7-200.
Junta	El TD 200 incluye una almohadilla protectora y una junta para facilitar el montaje en entornos desfavorables.
Puerto de comunicación	El puerto de comunicación es un conector D subminiatura de 9 pines que permite conectar el TD 200 a una CPU S7-200 mediante el cable TD/CPU adjunto.
Toma de alimentación	El TD 200 se puede conectar a una fuente de alimentación externa a través de la toma situada en el lado derecho del TD 200. Dicha conexión no se requiere si se utiliza el cable TD/CPU.
Cable TD/CPU	El cable TD/CPU se utiliza para la comunicación y alimentación del TD 200. Se trata de un cable de conexión de 9 pines que se suministra junto con el TD 200.
Plantilla de teclado	La plantilla de teclado es un rótulo extraíble que se puede utilizar para personalizar las teclas de función conforme a la aplicación que se les dé.

Tabla 1.2 Componentes del TD-200.

1.3 ESTUDIO DE SOFTWARE.

1.3.1 Software recomendado.

Para el correcto funcionamiento de un sistema S7-200 es necesario tener en cuenta los siguientes aspectos de software.

STEP 7-Micro/WIN 32, versión 3.1 y STEP 7-Micro/WIN 32 Toolbox son aplicaciones de software que soportan los entornos Windows 95 (de 32 bits), Windows 98 y Windows NT.

Para poder utilizar el software se recomiendan los siguientes componentes:

- Un panel táctil TP070 para su utilización con STEP 7-Micro/WIN 32 Toolbox.
- Un ordenador personal (PC) con un procesador 80586 o superior y 16 MB de RAM, o bien una unidad de programación Siemens con STEP 7-Micro/WIN 32 instalado (p.ej. una PG 740). Como mínimo se necesita un procesador 80486 con 8 MB de RAM.
- Uno de los componentes siguientes:
 - Un cable PC/PPI conectado al puerto de comunicación.
 - Una tarjeta de procesador de comunicaciones (CP).
- Una pantalla VGA o cualquier otra pantalla soportada por Microsoft Windows que tenga una resolución de 1024 X 768.

- 50 MB libres en el disco duro (como mínimo).
- Windows 95, Windows 98 o Windows NT 4.0.
- Opcional pero recomendable: un ratón asistido por Microsoft Windows.

STEP 7-Micro/WIN 32 incorpora Ayuda en pantalla y una Guía de iniciación. Elija el comando de menú **Ayuda** o pulse la tecla F1 para obtener las informaciones más recientes.

STEP 7-Micro/WIN 32 Toolbox incluye además el software de configuración del panel táctil TP070, así como las operaciones del protocolo USS para los accionamientos MicroMaster.

1.3.1.1 Instalación STEP 7-Micro/WIN 32.

Para la instalación del software STEP 7-microWIN 32, se recomienda tomar en cuenta los siguientes pasos:

- Si ya está instalada una versión anterior de STEP 7-Micro/WIN 32, cree una copia de seguridad de todos los proyectos de STEP 7-Micro/WIN.
- Verifique que todas las demás aplicaciones estén cerradas, incluyendo la barra de herramientas de Microsoft Office.
- Verifique que esté conectado el cable entre el PC y la CPU.

Para la instalación del software se debe seguir los siguientes pasos:

1. Inserte el CD o el disquete en la correspondiente unidad del PC.
2. Haga clic en el botón “Inicio” para abrir el menú de Windows.
3. Haga clic en Ejecutar.
4. Si la instalación se efectúa desde un disquete: En el cuadro de diálogo “Ejecutar”, digite **a:\setup** y haga clic en el botón “Aceptar” o pulse la tecla ENTER. Así se inicia la instalación.

Si la instalación se efectúa desde un CD: En el cuadro de diálogo “Ejecutar”, digite **e:\setup** (donde “e” es la letra correspondiente a la unidad de CD-ROM) y haga clic en el botón “Aceptar” o pulse la tecla ENTER, de esta manera se inicia la instalación.

5. Siga las instrucciones que van apareciendo en pantalla hasta finalizar la instalación.

6. Al final de la instalación aparecerá automáticamente el cuadro de diálogo “Ajustar interfase PG/PC”. Haga clic en el botón “Cancelar” para continuar.
7. Aparecerá el cuadro de diálogo “Fin de la instalación”, con una de las siguientes opciones:

Primera. Sí, deseo reiniciar mi ordenador en este momento. (Ajuste estándar) No, reiniciaré mi ordenador más adelante.

Si se visualiza la primera opción, es recomendable que acepte los ajustes estándar y haga clic en el botón ”Finalizar” para concluir la instalación y leer el archivo README que contiene las informaciones más recientes acerca de STEP 7-Micro/WIN 32.

Segunda. Sí, deseo leer el archivo LÉAME ahora mismo. (Ajuste estándar) Sí, deseo arrancar STEP 7-Micro/WIN 32 ahora. Si se visualiza la segunda opción, es recomendable que acepte los ajustes estándar y haga clic en el botón ”Finalizar” para concluir la instalación y leer el archivo README (LÉAME) que contiene las informaciones más recientes acerca de STEP 7-Micro/WIN 32.

En la segunda opción se pueden seleccionar las dos casillas de verificación. Haciendo clic en el botón “Finalizar” al estar seleccionadas las dos casillas, se concluirá la instalación, se visualizará el archivo LÉAME y se arrancará STEP 7-Micro/WIN 32.

1.3.1.2 Software STEP 7 micro-WIN.

Este software nos permite la creación edición y modificación de proyectos de automatización o control en base a un PLC S7-200, para realizar un proyecto se lo puede realizar mediante tres formas: AWL, KOP y FUP, en la figura 1,7 observamos el programa STEP 7 y sus diferentes componentes.

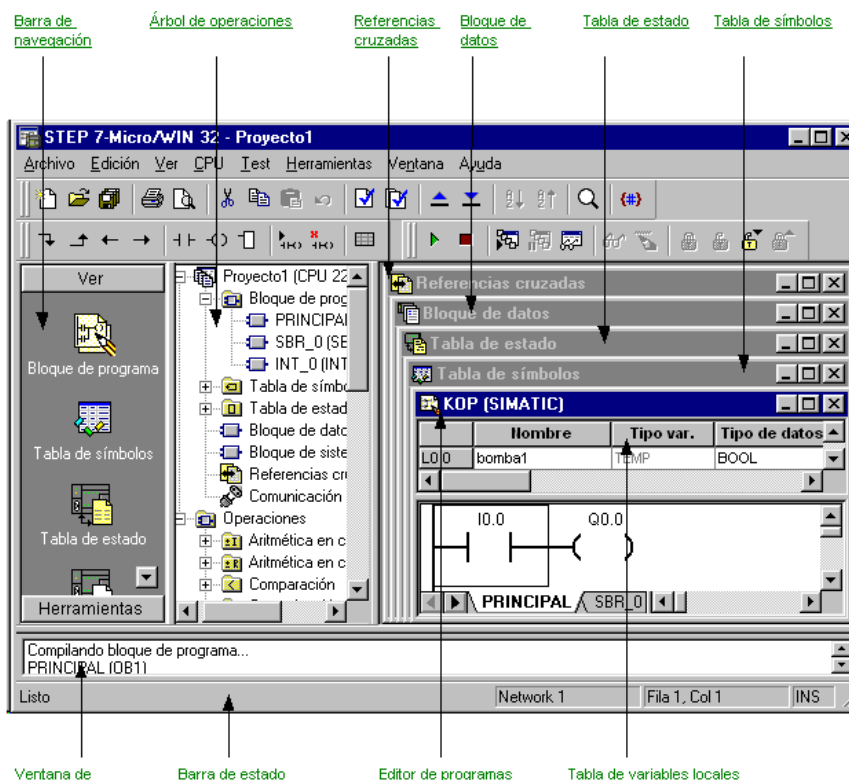


Figura 1,7 Programa STEP 7 y sus diferentes componentes

Barra de menú. Permite ejecutar funciones utilizando el ratón o combinaciones de teclas. El menú Herramientas se puede personalizar añadiendo aplicaciones propias.



Barras de herramientas. Permiten acceder fácilmente con el ratón a las funciones de STEP 7-Micro/WIN 32 utilizadas con frecuencia. El contenido y el aspecto de cada una de las barras de herramientas se pueden personalizar.



Barra de navegación. Presenta grupos de botones para facilitar la programación:

“**Ver**” Seleccione esta categoría para visualizar los botones Bloque de programa, Tabla de símbolos, Tabla de estado, Bloque de datos, Bloque de sistema, Referencias cruzadas y Comunicación.

“**Herramientas**” Seleccione esta categoría para visualizar los botones del Asistente de operaciones y del Asistente TD 200.

Árbol de operaciones. Ofrece una vista en árbol de todos los objetos del proyecto y de todas las operaciones disponibles en el editor de programas actual (KOP, FUP o AWL).

Para insertar unidades de organización del programa adicionales (UOP's) en el área de proyectos del árbol haga clic con el botón derecho del ratón en la carpeta en cuestión.

De igual manera, puede pulsar el botón derecho del ratón en una UOP individual para abrirla, borrarla, editar su hoja de propiedades o cambiar el nombre de subrutinas y/o rutinas de interrupción. Estando en el área de operaciones del árbol, puede hacer clic con el botón derecho del ratón en una carpeta o en una operación individual, con objeto de ocultar el árbol entero. Tras abrir una carpeta de operaciones puede insertar operaciones en la posición de cursor en la ventana del editor de

programas, haciendo doble clic en la operación en cuestión o utilizando el método de arrastrar y soltar. (En programas AWL, el árbol de operaciones se utiliza sólo de referencia, siendo preciso teclear la operación, puesto que no se puede adoptar del árbol).

1.3.1.2.1 Visualizar el estado del programa en AWL.

Si el programa se visualiza utilizando el editor AWL, STEP 7-Micro/WIN 32 ofrece un método para observar el estado de ejecución del programa tras evaluarse cada una de las operaciones, a este método de observación del estado se denomina “estado AWL”, la sección del programa para la que se ha habilitado el estado AWL se denomina “ventana de estado AWL”. El tamaño de esta ventana es aproximadamente igual al de la pantalla de STEP 7-Micro/WIN 32, el método AWL es el lenguaje NEMONICO para programar en los sistemas S7-200.

1.3.1.2.2 Visualizar el estado del programa en KOP.

El estado KOP muestra el estado de todos los valores de los operandos de las operaciones, todas las informaciones de estado se basan en los valores leídos al final de un ciclo del CPU.

STEP 7-Micro/WIN 32 adquiere los valores para visualizar el estado durante varios ciclos del CPU, actualizando luego la ventana de estado KOP. Por consiguiente, el estado KOP visualizado no refleja el estado real de ejecución de cada elemento KOP.

El método KOP es el lenguaje LEADER del sistema S7-200, que no es más que realizar el programa a manera de escalera con todos los componentes y operaciones asociadas en una línea de instrucciones.

1.3.1.2.3 Visualizar el estado del programa en FUP.

El estado FUP muestra el estado de todos los valores de los operandos de las operaciones. Todas las informaciones de estado se basan en los valores leídos al final de un ciclo del CPU. STEP 7-Micro/WIN 32 adquiere los valores para visualizar el estado durante varios ciclos del CPU, actualizando luego la ventana de estado FUP, por lo tanto, el estado FUP visualizado no refleja el estado real de ejecución de cada elemento FUP.

1.3.1.3 Configurar la comunicación utilizando el cable PC/PPI.

Aquí se explica cómo configurar la comunicación entre la CPU S7-200 y el PC utilizando el cable PC/PPI. Esta es una configuración con un solo maestro y sin ningún otro equipo de hardware instalado (como p.ej. un módem o una unidad de programación).

1.3.1.4 Conectar el PC al CPU.

La figura 1.8 muestra una configuración típica para conectar el PC a la CPU utilizando el cable PC/PPI. Para establecer un enlace correcto entre los componentes:

1. Ajuste los interruptores DIP del cable PC/PPI a la velocidad de transferencia asistida por su PC. Seleccione también las opciones "11 bits" y "DCE" si su cable PC/PPI las asiste.
2. Conecte el extremo RS-232 ("PC") del cable PC/PPI al puerto de comunicaciones de su PC (COM1 ó COM2) y apriete los tornillos.
3. Conecte el extremo RS-485 ("PPI") del cable PC/PPI al puerto de comunicaciones del CPU y apriete los tornillos.

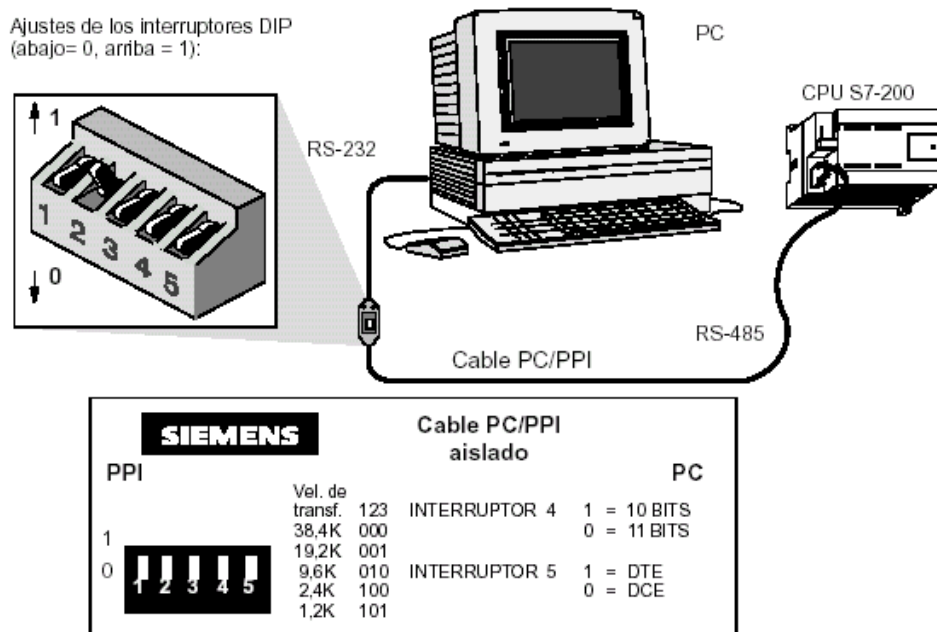


Figura 1.8 Comunicación de un CPU en modo PPI

1.3.1.5 Verificación de parámetros estándar del interfase de comunicación.

Para verificar los parámetros estándar de su interfase, siga los siguientes pasos:

1. En la ventana de STEP 7-Micro/WIN 32, haga clic en el icono "Comunicación" o elija el comando de menú **Ver > Comunicación**. Aparecerá el cuadro de diálogo "Enlaces de comunicación".
2. En el cuadro de diálogo "Enlaces de comunicación", haga doble clic en el icono del cable PC/PPI. Aparecerá el cuadro de diálogo "Ajustar interfase PG/PC" (figura 1.9).

3. Haga clic en el botón “Propiedades” para acceder al cuadro de diálogo donde se visualizan las propiedades del interfase (figura 1.10). Verifique las propiedades. La velocidad de transferencia estándar es de 9,6 kbit/s.

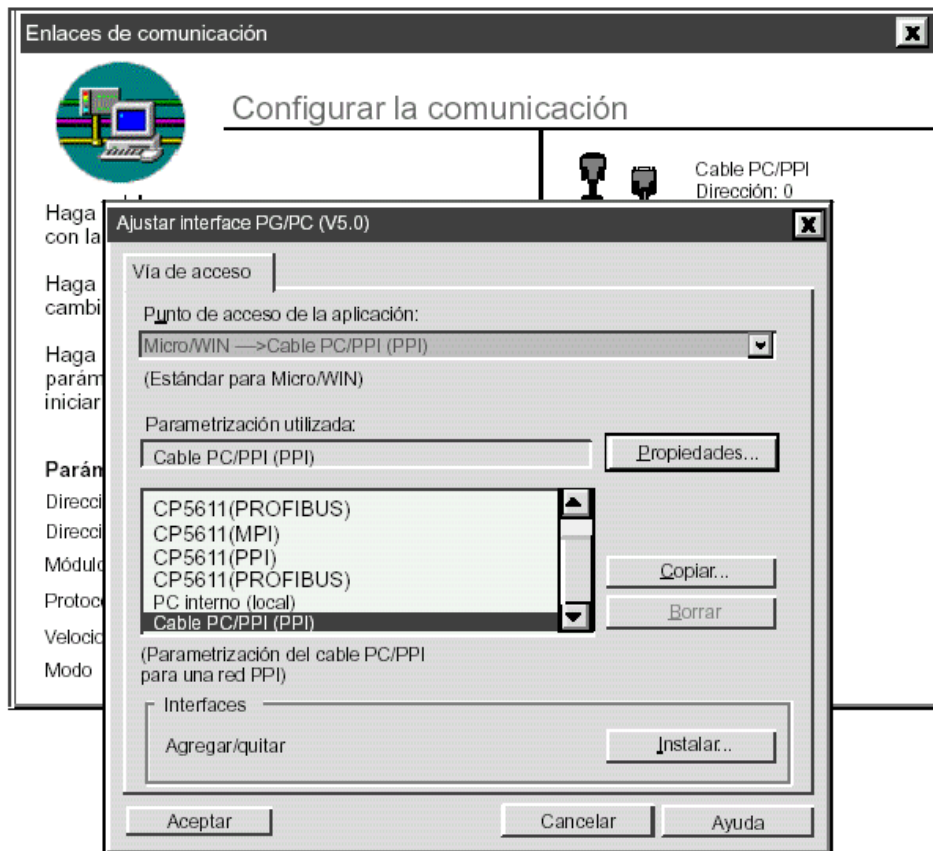


Figura 1.9 Ajustes para el interfase PG/PC.

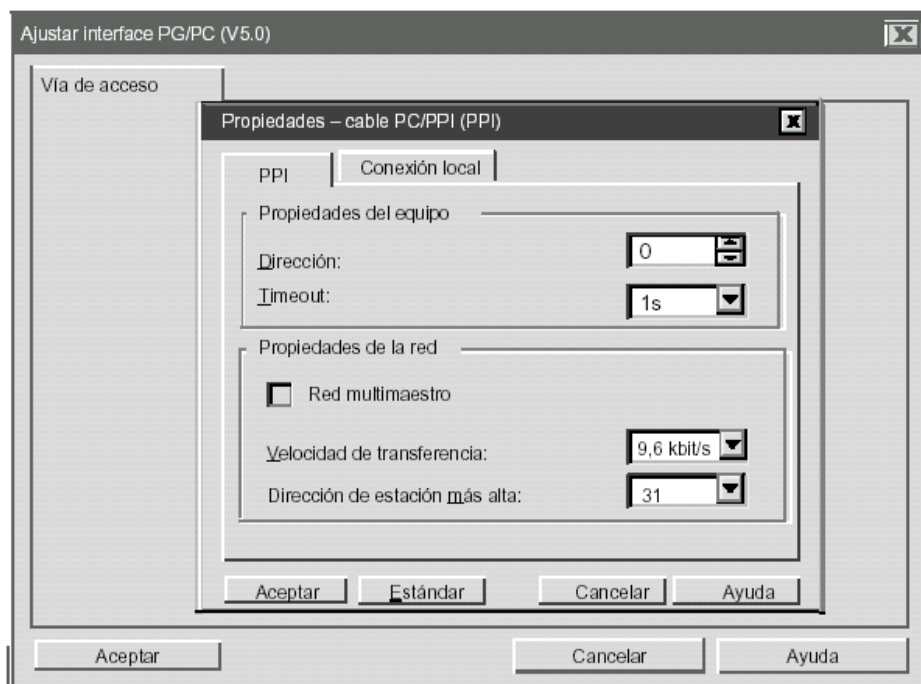


Figura 1.10 Propiedades de interfase PG/PC.

1.3.1.6 Establecer comunicación con el CPU.

Después de haber instalado el software STEP 7-Micro/WIN 32 en el PC y determinado los ajustes de comunicación del cable PC/PPI, podrá establecer un enlace con el CPU S7-200. (Si está utilizando una unidad de programación, STEP 7-Micro/WIN 32 ya estará instalado).

Para establecer la comunicación con el CPU S7-200, siga los siguientes pasos:

1. En la pantalla de STEP 7-Micro/WIN 32, haga clic en el icono “Comunicación” o elija el comando de menú **Ver > Comunicación**. Aparecerá el cuadro de diálogo “Enlaces de comunicación” donde se indica que no hay ninguna CPU conectada.
2. En el cuadro de diálogo “Enlaces de comunicación”, haga doble clic en el icono “Actualizar”. STEP 7-Micro/WIN 32 verifica si hay CPU's S7-200 (estaciones conectadas), hasta la dirección de estación más alta indicada en la configuración de los parámetros de comunicación. Por cada estación conectada aparecerá un icono de CPU en el cuadro de diálogo “Enlaces de comunicación” (figura 1.11).
3. Haga doble clic en el icono del CPU con la que desea establecer la comunicación. Como podrá apreciar, los parámetros de comunicación visualizados en el cuadro de diálogo corresponden a la estación seleccionada.
4. Así queda establecido el enlace con el CPU S7-200.

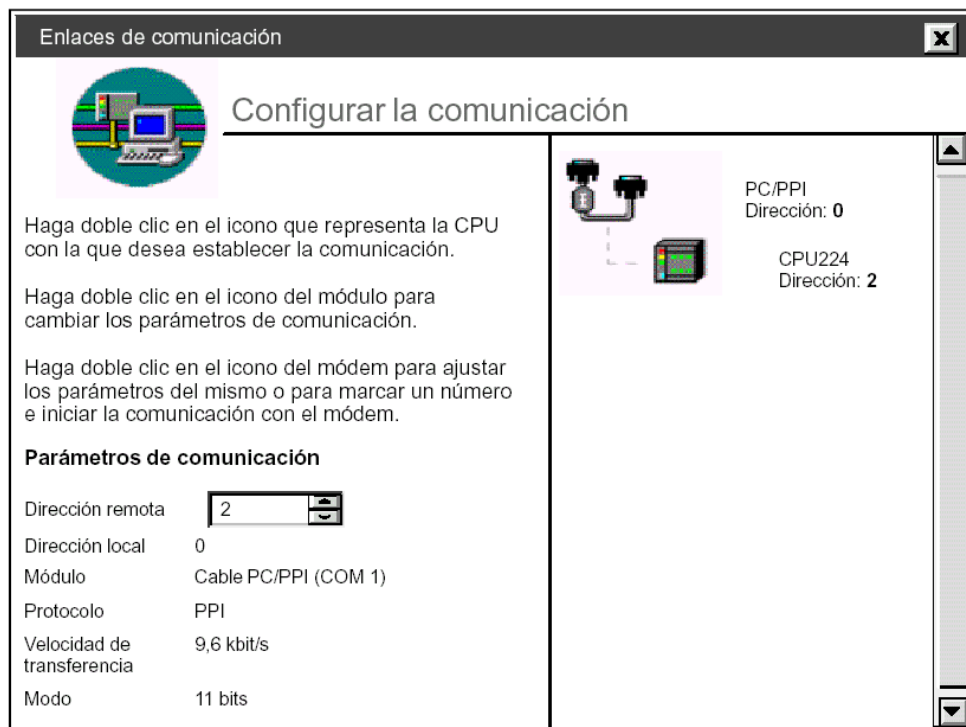


Figura 1.11 Enlaces de comunicación.

1.3.1.7 Fallas de comunicación.

Las siguientes situaciones pueden causar fallas de comunicación:

- Velocidad de transferencia incorrecta: corrija la velocidad de transferencia.
- Dirección de estación incorrecta: corrija la dirección de estación.
- Cable PC/PPI ajustado incorrectamente: verifique los ajustes de los interruptores DIP del cable PC/PPI.
- Puerto de comunicaciones incorrecto en el PC: verifique el puerto COM.
- CPU en modo Freeport (puerto de comunicaciones bajo control del programa de usuario): cambie el CPU a modo STOP.
- Conflicto con otros maestros: desconecte el CPU de la red.

1.3.1.8 Cambio de parámetros de comunicación del CPU.

Tras haber establecido un enlace con el CPU S7-200, puede verificar o cambiar los parámetros de comunicación del CPU.

Para cambiar los parámetros de comunicación, siga los siguientes pasos:

1. En la barra de navegación, haga clic en el icono “Bloque de sistema” o elija el comando de menú **Ver > Bloque de sistema**.
2. Aparecerá el cuadro de diálogo “Bloque de sistema”. Haga clic en la ficha “Puertos” (figura 1.12). El ajuste estándar de la dirección de estación es 2 y el de la velocidad de transferencia es de 9,6 kbit/s.
3. Haga clic en “Aceptar” para conservar esos parámetros. Si desea modificar los parámetros, efectúe los cambios deseados y haga clic en el botón “Aceptar”.
4. En la barra de herramientas, haga clic en el botón “Cargar en CPU” para cargar los cambios en el CPU.

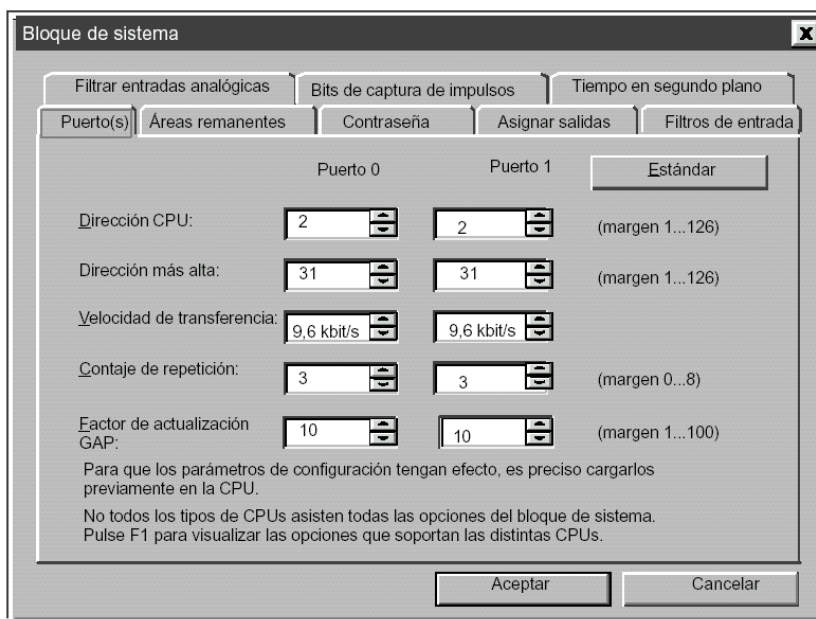


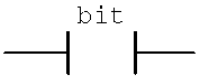
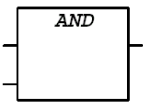
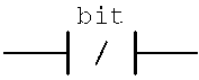
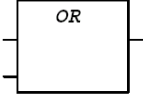
Figura 1.12 Cambio de parámetros de comunicación.

1.3.2 Operaciones del PLC S7-200.

Las operaciones del PLC Simatic S7-200 son variadas, lo que ayuda a que los proyectos tengan más facilidad de realizarse de acuerdo a la necesidad y aplicación que se requiera, entre ellas tenemos.

1.3.2.1 Operaciones lógicas con bits (SIMATIC).

- **Contactos estándar**

KOP	FUP	AWL
		LD bit A bit O bit
		LDN bit AN bit ON bit

Estas operaciones leen el valor direccionado de la memoria o de la imagen del proceso si el tipo de datos es I o Q. Para los cuadros AND y OR se pueden utilizar siete entradas como máximo.

El **Contacto normalmente abierto** se cierra (ON) si el bit es igual a 1.

El **Contacto normalmente cerrado** se cierra (ON) si el bit es igual a 0.

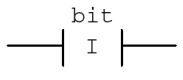
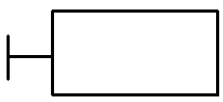
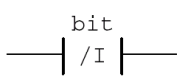
En KOP, las operaciones Contacto normalmente abierto y Contacto normalmente cerrado se representan mediante contactos.

En FUP, los contactos normalmente abiertos se representan mediante cuadros AND/OR. Estas operaciones sirven para manipular señales booleanas de la misma forma que los contactos KOP.

En AWL, el contacto normalmente abierto se representa con las operaciones **Cargar, Y u O**. Estas operaciones cargan el valor binario del bit de dirección en el nivel superior de la pila y lo combinan mediante Y u O.

En AWL, el contacto normalmente cerrado se representa con las operaciones **Cargar valor negado, Y-NO y O-NO**. Estas operaciones cargan el valor binario invertido del bit de la dirección en el nivel superior de la pila y lo combinan mediante Y u O.

- **Contactos directos**

KOP	FUP	AWL
		LDI bit AI bit OI bit
		LDNI bit ANI bit ONI bit

Estas operaciones leen el valor de la entrada física al ejecutarse la operación, pero la imagen del proceso no se actualiza.

El **Contacto abierto directo** se cierra (se activa) si la entrada física es 1.

El **Contacto cerrado directo** se cierra (se activa) si la entrada física es 0.

En KOP, las operaciones Contacto abierto directo y Contacto cerrado directo se representan mediante contactos.

En FUP, la operación Contacto abierto directo se representa mediante un corchete delante del operando. El corchete puede faltar si se usa la circulación de corriente.

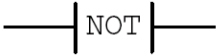

Esta operación sirve para manipular señales físicas de la misma forma que los contactos KOP.

En FUP, la operación Contacto cerrado directo se representa también mediante un corchete y el símbolo de negación delante del operando. La operación Contacto Normalmente Cerrado se realiza situando el símbolo de negación en la raíz de la señal de entrada.

En AWL, el contacto abierto directo se representa con las operaciones **Cargar directamente, Y directa y O directa**. Estas operaciones cargan directamente el valor de la entrada física en el nivel superior de la pila y lo combinan mediante Y u O.

En AWL, el contacto cerrado directo se representa con las operaciones **Cargar valor negado directamente, Y-NO directa u O-NO directa (ONI)**. Estas operaciones cargan directamente el valor binario negado de la entrada física en el nivel superior de la pila y lo combinan mediante Y u O.

- **Not.**

KOP	FUP	AWL
		<p data-bbox="1126 1637 1174 1659">NOT</p>

El contacto **NOT** cambia el estado de la entrada de circulación de corriente.

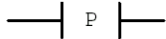
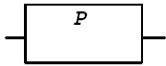

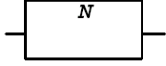
La corriente se detiene al alcanzar el contacto NOT. Si no logra alcanzar el contacto, entonces hace circular la corriente.

En KOP, la operación **NOT** se representa en forma de contacto.

En FUP, la operación **NOT** utiliza el símbolo gráfico de negación con entradas booleanas de cuadro.

En AWL, la operación **Invertir primer valor** (NOT) invierte el primer valor de la pila de 0 a 1, o bien de 1 a 0.

- **Detectar flanco positivo o negativo**

KOP	FUP	AWL
		EU
		ED

El contacto **Detectar flanco positivo** permite que la corriente circule durante un ciclo cada vez que se produce un cambio de 0 a 1 (de “off” a “on”).

El contacto **Detectar flanco negativo** permite que la corriente circule durante un ciclo cada vez que se produce un cambio de señal de 1 a 0 (de “on” a “off”).

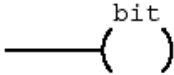
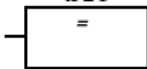
En KOP, las operaciones Detectar flanco positivo y negativo se representan mediante contactos.

En FUP, estas operaciones se representan mediante los cuadros POS y NEG.

En AWL, la transición positiva se representa con la operación **Detectar flanco positivo**. Cuando se detecta un cambio de señal de 0 a 1 en el primer valor de la pila, éste se pone a 1. En caso contrario, se pone a 0.

En AWL, la transición negativa se representa con la operación **Detectar flanco negativo**. Cuando se detecta un cambio de señal de 1 a 0 en el primer valor de la pila, éste se pone a 1. En caso contrario, se pone a 0.

- **Asignar.**

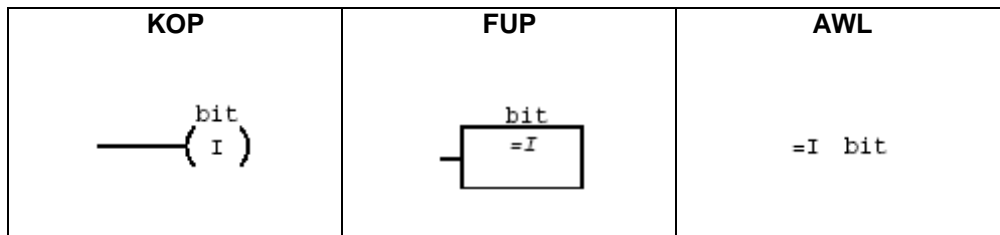
KOP	FUP	AWL
		= bit

Cuando se ejecuta la operación **Asignar**, el bit de salida se activa en la imagen del proceso.

Cuando la operación Asignar se ejecuta en KOP y FUP, el bit indicado se ajusta de forma equivalente a la circulación de la corriente.

En AWL, la operación Asignar copia el primer valor de la pila el bit indicado.

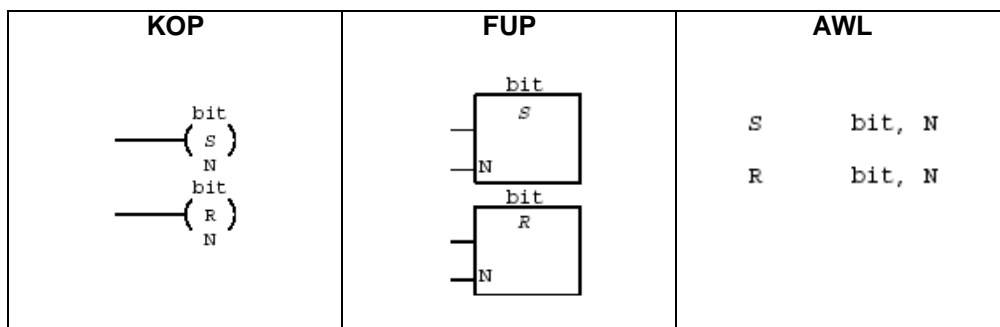
- **Asignar directamente.**



Cuando se ejecuta la operación **Asignar directamente**, la entrada física (bit u OUT) se ajusta de forma equivalente a la circulación de la corriente, la “I” indica que la operación se ejecuta directamente. El nuevo valor se escribe entonces tanto en la salida física como en la correspondiente dirección de la imagen del proceso. En cambio, en las operaciones no directas, el nuevo valor se escribe sólo en la imagen del proceso.

En AWL, la operación Asignar directamente copia el primer valor de la pila directamente en la salida física indicada (bit).

- **Poner a 1, Poner a 0 (N bits).**

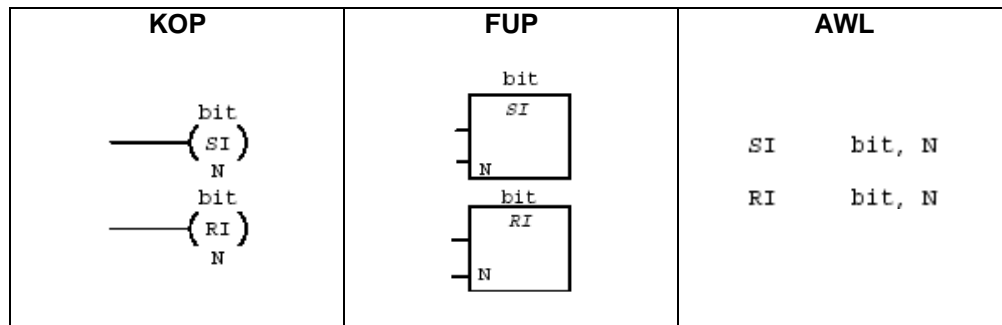


Cuando se ejecutan las operaciones **Poner a 1** y **Poner a 0**, se activa (se pone a 1) o se desactiva (se pone a 0) el número indicado de salidas (N) a partir del valor indicado por el bit o por el parámetro OUT.

El margen de entradas y/o salidas que se pueden activar o desactivar está comprendido entre 1 y 255. Con la operación Poner a 0, si el bit indicado es un bit T (bit de temporización) o un bit C (bit de conteo), se desactivará el bit de temporización/conteo y se borrará el valor actual del temporizador/contador.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área).

- **Poner a 1 directamente, Poner a 0 directamente (N bits).**

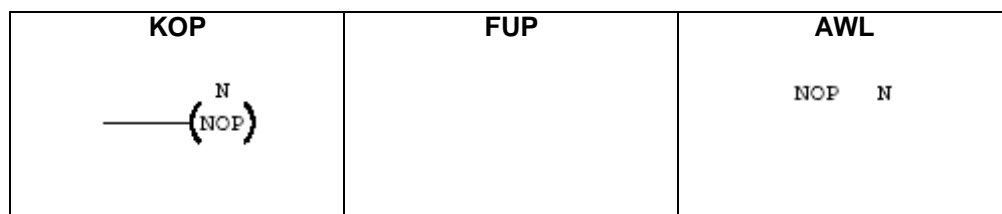


Cuando se ejecutan las operaciones **Poner a 1 directamente** y **Poner a 0 directamente** se activa (se pone a 1) o se desactiva (se pone a 0) directamente el número indicado de salidas físicas (N) a partir del bit o de OUT, el margen de entradas y/o salidas que se pueden activar o desactivar está comprendido entre 1 y 128.

La “I” indica que la operación se ejecuta directamente. Al ejecutarse ésta, el nuevo valor se escribe tanto en la salida física como en la correspondiente dirección de la imagen del proceso. En cambio, en las operaciones no directas, el nuevo valor se escribe sólo en la imagen del proceso.

Condiciones de error que ponen ENO a 0: SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto), 0091 (operando fuera de área).

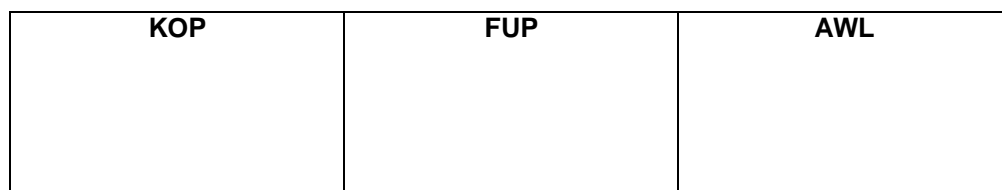
- **Operación nula.**

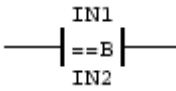
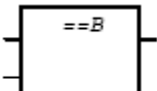


La **Operación nula** no tiene efecto alguno sobre la ejecución del programa. Esta operación no está disponible en FUP. El operando N es un número comprendido entre 0 y 255.

1.3.2.2 Operaciones de comparación (SIMATIC).

- **Comparar byte.**



		LDB= IN1 , IN2 AB= IN1 , IN2 OB= IN1 , IN2 LDB<> IN1 , IN2 AB<> IN1 , IN2 OB<> IN1 , IN2 LDB< IN1 , IN2 AB< IN1 , IN2 OB< IN1 , IN2 LDB<= IN1 , IN2 AB<= IN1 , IN2 OB<= IN1 , IN2 LDB> IN1 , IN2 AB> IN1 , IN2 OB> IN1 , IN2 LDB>= IN1 , IN2 AB>= IN1 , IN2 OB>= IN1 , IN2
---	---	---

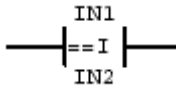
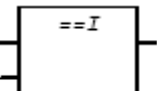
La operación **Comparar byte** se utiliza para comparar dos valores: IN1 e IN2. Las comparaciones incluyen: IN1 = IN2, IN1 >= IN2, IN1 <= IN2, IN1 > IN2, IN1 < IN2, o IN1 <> IN2. Las comparaciones de bytes no llevan signo.

En KOP, el contacto se activa si la comparación es verdadera.

En FUP, la salida se activa si la comparación es verdadera.

En AWL, las operaciones cargan un “1” en el nivel superior de la pila si la comparación es verdadera.

- **Comparar entero.**

KOP	FUP	AWL
		LDW= IN1 , IN2 AW= IN1 , IN2 OB= IN1 , IN2 LDW<> IN1 , IN2 AW<> IN1 , IN2 OW<> IN1 , IN2 LDW< IN1 , IN2 AW< IN1 , IN2 OW< IN1 , IN2 LDW<= IN1 , IN2 AW<= IN1 , IN2 OW<= IN1 , IN2 LDW> IN1 , IN2 AW> IN1 , IN2 OW> IN1 , IN2 LDW>= IN1 , IN2 AW>= IN1 , IN2 OW>= IN1 , IN2

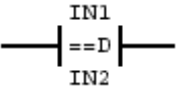
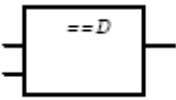
La operación **Comparar entero** se utiliza para comparar dos valores: IN1 e IN2. Las comparaciones incluyen: IN1 = IN2, IN1 >= IN2, IN1 <= IN2, IN1 > IN2, IN1 < IN2, o IN1 <> IN2.

Las comparaciones de enteros llevan signo (16#7FFF >16#8000), en KOP el contacto se activa si la comparación es verdadera.

En FUP la salida se activa si la comparación es verdadera.

En AWL las operaciones cargan un “1” en el nivel superior de la pila y combinan el valor “1” con el primer valor de la pila mediante Y u O cuando la comparación es verdadera.

- **Comparar palabra doble.**

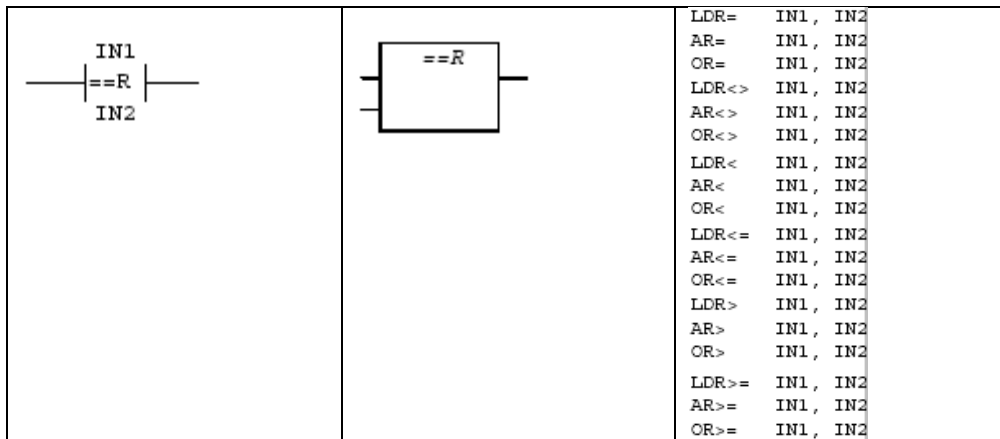
KOP	FUP	AWL
		<pre> LDD= IN1 , IN2 AD= IN1 , IN2 OD= IN1 , IN2 LDD<> IN1 , IN2 AD<> IN1 , IN2 OD<> IN1 , IN2 LDD< IN1 , IN2 AD< IN1 , IN2 OD< IN1 , IN2 LDD<= IN1 , IN2 AD<= IN1 , IN2 OD<= IN1 , IN2 LDD> IN1 , IN2 AD> IN1 , IN2 OD> IN1 , IN2 LDD>= IN1 , IN2 AD>= IN1 , IN2 OD>= IN1 , IN2 </pre>

La operación **Comparar palabra doble** se utiliza para comparar dos valores: IN1 e IN2. Las comparaciones incluyen: IN1 = IN2, IN1 >= IN2, IN1 <= IN2, IN1 > IN2, IN1 < IN2, o IN1 <> IN2.

Las comparaciones de palabras dobles llevan signo (16#7FFFFFFF > 16#80000000), en KOP, el contacto se activa si la comparación es verdadera, en FUP, la salida se activa si la comparación es verdadera, e AWL, las operaciones cargan un “1” en el nivel superior de la pila y combinan el valor “1” con el primer valor de la pila mediante Y u O cuando la comparación es verdadera.

- **Comparar real.**

KOP	FUP	AWL



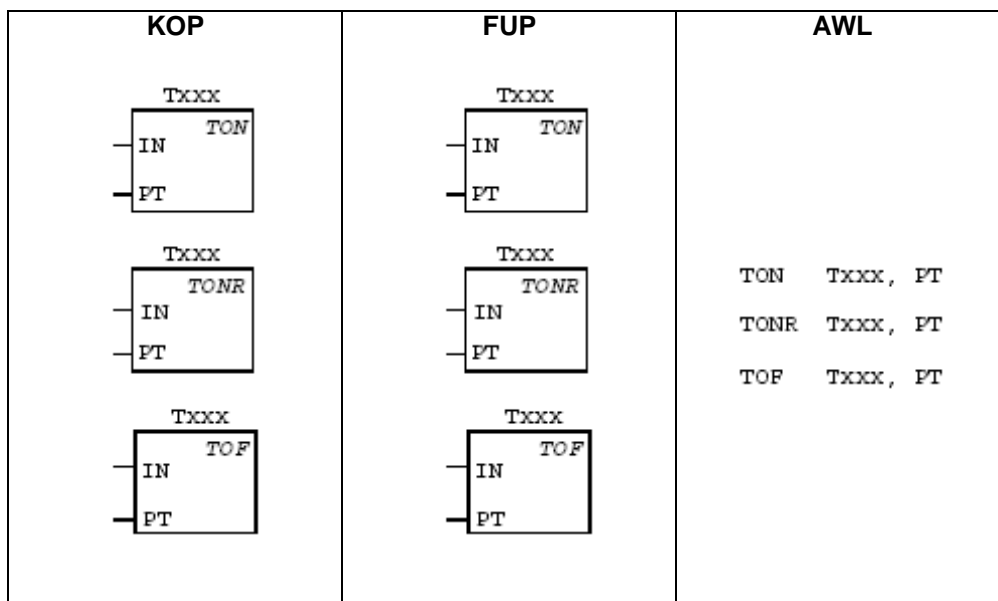
La operación **Comparar real** se utiliza para comparar dos valores: IN1 e IN2. Las comparaciones incluyen: IN1 = IN2, IN1 >= IN2, IN1 <= IN2, IN1 > IN2, IN1 < IN2, o IN1 <> IN2.

Las comparaciones de números reales llevan signo, e KOP, el contacto se activa si la comparación es verdadera, e FUP, la salida se activa si la comparación es verdadera.

En AWL, las operaciones cargan un "1" en el nivel superior de la pila y combinan el valor "1" con el primer valor de la pila mediante Y u O cuando la comparación es verdadera.

1.3.2.3 Operaciones de temporización (SIMATIC).

- **Temporizador de retardo a la conexión, Temporizador de retardo a la conexión memorizado, temporizador de retardo a la desconexión.**



Las operaciones **Temporizador de retardo a la conexión** y **Temporizador de retardo a la conexión memorizado** cuentan el tiempo al estar activada (ON) la entrada de habilitación. Si el valor actual (Txxx) es

mayor o igual al valor de preselección (PT), se activa el bit de temporización (bit T). Cuando la entrada de habilitación está desconectada (OFF), el valor actual se borra en el caso del temporizador de retardo a la conexión. En cambio, se conserva en el temporizador de retardo a la conexión memorizado. Éste último sirve para acumular varios períodos de tiempo de la entrada en ON. Para borrar el valor actual del temporizador de retardo a la conexión memorizado se utiliza la operación Poner a 0 (R). Tanto el temporizador de retardo a la conexión como el temporizador de retardo a la conexión memorizado continúan contando tras haberse alcanzado el valor de preselección y paran de contar al alcanzar el valor máximo de 32767.

El **Temporizador de retardo a la desconexión** se utiliza para retardar la puesta a 0 (OFF) de una salida durante un período determinado tras haberse desactivado (OFF) una entrada. Cuando la entrada de habilitación se activa (ON), el bit de temporización se activa (ON) inmediatamente y el valor actual se pone a 0. Cuando la entrada se desactiva (OFF), el temporizador cuenta hasta que el tiempo transcurrido alcance el valor de preselección. Una vez alcanzado éste, el bit de temporización se desactiva (OFF) y el valor actual detiene el conteo. Si la entrada está desactivada (OFF) durante un tiempo inferior al valor de preselección, el bit de temporización permanece activado (ON). Para que la operación TOF comience a contar se debe producir un cambio de ON a OFF, si un temporizador TOF se encuentra dentro de una sección SCR y ésta se encuentra desactivada, el valor actual se pone a 0, el bit de temporización se desactiva (OFF) y el valor actual no cuenta.

La resolución viene determinada por el número del temporizador (Tabla 1.3). El valor actual resulta del valor de conteo multiplicado por la base de tiempo. Por ejemplo, el valor de conteo 50 en un temporizador de 10 mS equivale a 500 mS.

Tipo de temporizador	Resolución en milisegundos (ms)	Valor máximo en segundos (s)	Nº de temporizador
TONR (memorizado)	1 mS	32,767 S (0,546 min.)	T0, T64
	10 mS	327,67 S (0,546 min.)	T1 a T4, T65 a T68
	100 mS	3276,7 S (0,546 min.)	T5 a T31, T69 a T95
TON, TOF (no memorizados)	1 mS	32,767 S (0,546 min.)	T32, T96
	10 mS	327,67 S (0,546 min.)	T33 a T36, T97 a T100
	100 mS	3276,7 S (0,546 min.)	T37 a T63, T101 a T255

Tabla 1.3 Temporizadores y sus resoluciones

Resolución de 1 milisegundo

Los temporizadores con resolución de 1mS cuentan el número de intervalos de 1mS que han transcurrido desde que se habilitó el temporizador activo de 1mS. La temporización arranca al comenzarse a ejecutar la operación. No obstante, los temporizadores de 1mS se actualizan (tanto el bit de temporización como el valor actual) cada milisegundo de forma asíncrona al ciclo. En otras palabras, el bit de temporización y el valor actual se actualizan varias veces en un ciclo que dure más de 1mS.

La operación de temporización se utiliza para activar e inicializar el temporizador o, en el caso del temporizador de retardo a la conexión memorizado (TONR), para desactivarlo.

Puesto que el temporizador puede arrancar en cualquier momento durante un milisegundo, el valor de preselección se debe ajustar a un intervalo de tiempo que exceda el intervalo mínimo deseado. Por ejemplo, para garantizar un intervalo mínimo de 56mS utilizando un temporizador de 1mS, es preciso ajustar el valor de preselección a 57.

Resolución de 10 milisegundos.

Los temporizadores con resolución de 10mS cuentan el número de intervalos de 10mS que han transcurrido desde que se habilitó el temporizador activo de 10mS.

La temporización arranca al comenzarse a ejecutar la operación. No obstante, los temporizadores de 10mS se actualizan al comienzo de cada ciclo (en otras palabras, el valor actual y el bit de temporización permanecen constantes durante el ciclo), sumando el número acumulado de intervalos de 10mS (desde el comienzo del ciclo anterior) al valor actual del temporizador activo, puesto que el temporizador puede arrancar en cualquier momento durante un intervalo de 10mS, el valor de preselección se debe ajustar a un intervalo de tiempo que exceda el intervalo mínimo deseado. Por ejemplo, para garantizar un intervalo mínimo de 140mS utilizando un temporizador de 10mS, es preciso ajustar el valor de preselección a 15.

Resolución de 100 milisegundos.

Los temporizadores con resolución de 100mS cuentan el número de intervalos de 100mS que han transcurrido desde la última vez que se actualizó el temporizador activo de 100mS. Estos temporizadores se actualizan sumando el valor acumulado de intervalos de 100mS (desde el ciclo anterior) al valor actual del temporizador cuando se ejecuta la operación del mismo.

El valor actual de un temporizador de 100mS se actualiza únicamente si se ha ejecutado la operación correspondiente. Por consiguiente, si un temporizador de 100mS está habilitado, pero la correspondiente operación no se ejecuta en cada ciclo, no se actualizará el valor actual de ese temporizador y disminuirá el tiempo.

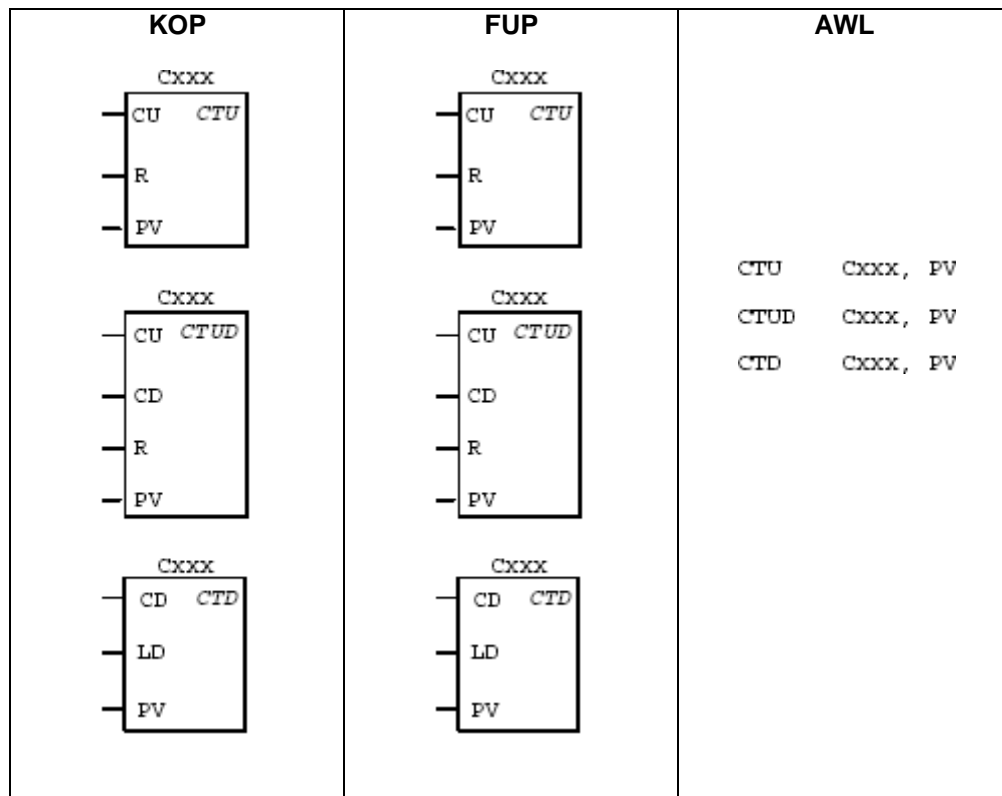
Por otra parte, si se ejecuta una misma operación con un temporizador de 100mS varias veces en un ciclo, el valor de 100mS acumulado se sumará también varias veces al valor actual del temporizador, con lo cual se prolonga el tiempo.

Debido a ello, es recomendable utilizar los temporizadores con una resolución de 100mS sólo cuando se ejecute exactamente una operación de temporización en cada ciclo, ya que el temporizador puede arrancar en cualquier momento durante un intervalo de 100mS, el valor de preselección se debe ajustar a un intervalo de

tiempo que exceda el intervalo mínimo deseado. Por ejemplo, para garantizar un intervalo mínimo de 2.100mS utilizando un temporizador de 100mS, es preciso ajustar el valor de preselección a 22.

1.3.2.4 Operaciones con contadores (SIMATIC).

- Contar adelante, Contar adelante/atrás, Contar atrás.



La operación **Contar adelante** empieza a contar hasta el valor máximo cuando se produce un flanco positivo en la entrada de conteo adelante (CU). Si el valor actual (Cxxx) es mayor o igual al valor de preselección (PV), se activa el bit de conteo (Cxxx). El contador se inicializa al activarse la entrada de desactivación (R) y para de contar cuando alcanza PV.

La operación **Contar adelante/atrás** empieza a contar adelante cuando se produce un flanco positivo en la entrada de conteo adelante (CU). Por el contrario, empieza a contar atrás cuando se produce un flanco

positivo en la entrada de conteo atrás (CD). Si el valor actual (Cxxx) es mayor o igual al valor de preselección (PV), se activa el bit de conteo (Cxxx). El contador se inicializa al activarse la entrada de desactivación (R).

La operación **Contar atrás** empieza a contar atrás desde el valor de preselección cuando se produce un flanco positivo en la entrada de conteo atrás (CD). Si el valor actual es igual a cero, se activa el bit de conteo (Cxxx). El contador desactiva el bit de conteo (Cxxx) y carga el valor actual con el valor de preselección (PV) cuando se activa la entrada de carga (LD). El contador atrás se detiene al alcanzar el valor cero.

Márgenes de conteo: Cxxx=C0 hasta C255, en AWL la entrada de desactivación CTU es el primer valor de la pila y la entrada de conteo adelante se carga en el segundo nivel de la pila, la entrada de desactivación CTUD es el primer valor de la pila, la entrada de conteo atrás se carga en el segundo nivel de la pila y la entrada de conteo adelante, en el tercer nivel, la entrada de carga CTD es el primer nivel de la pila y la entrada de conteo atrás es el valor cargado en el segundo nivel de la pila.

- **Descripción de las operaciones de conteo del S7-200.**

La operación Contar adelante (CTU) empieza a contar adelante a partir del valor actual cuando se produce un flanco positivo en la entrada de conteo adelante. El contador se inicializa cuando se activa la entrada de desactivación, el contador para de contar cuando se alcanza el valor máximo (32.767).

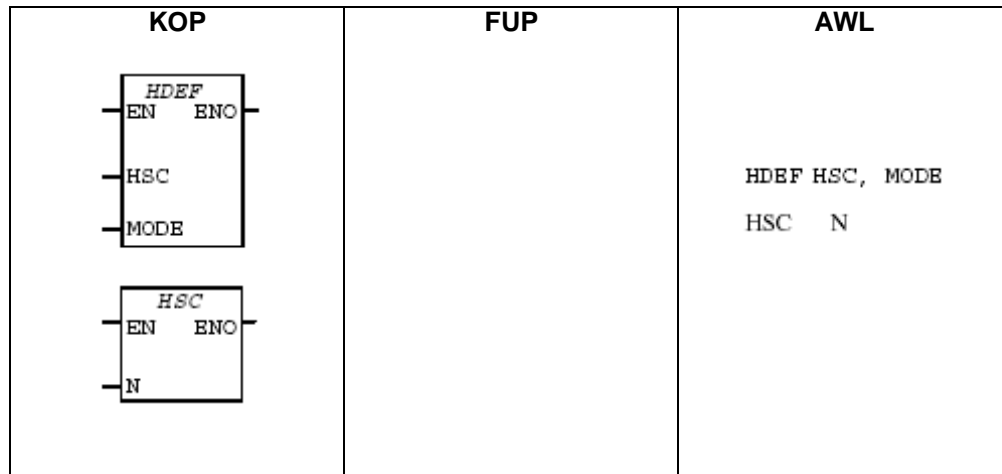
La operación Contar adelante/atrás (CTUD) empieza a contar adelante cuando se produce un flanco positivo en la entrada de conteo adelante, y empieza a contar atrás cuando se produce un flanco positivo en la entrada de conteo atrás. El contador se inicializa cuando se activa la entrada de desactivación o al ejecutarse la operación Poner a 0, cuando se alcanza el valor máximo (32.767), el siguiente flanco positivo en la entrada de conteo adelante invertirá el conteo hasta alcanzar el valor mínimo (-32.768). Igualmente, cuando se alcanza el valor mínimo (-32.768), el siguiente flanco positivo en la entrada de conteo atrás invertirá el conteo hasta alcanzar el valor máximo (32.767).

Los contadores Contar adelante y Contar adelante/atrás tienen un valor actual que almacena el valor de conteo actual. También disponen de un valor de preselección (PV) que se compara con el valor actual cuando se ejecuta la operación de conteo. Si el valor actual es mayor o igual al valor de preselección, se activa el bit de conteo (bit C). En caso contrario, el bit se desactiva.

La operación Contar atrás empieza a contar atrás a partir del valor actual cuando se produce un flanco negativo en la entrada de conteo adelante. El contador desactiva el bit de conteo y carga el valor actual con el valor de preselección cuando se activa la entrada de carga. El contador se detiene al alcanzar el valor cero y el bit de conteo (bit C) se activa, cuando se inicializa un contador con la operación Poner a 0, se desactivan tanto el bit de conteo como el valor actual del contador. El número del contador se debe utilizar para direccional tanto el valor actual como el bit C del contador.

1.3.2.4.1 Contadores rápidos (Alta velocidad).

- **Definir modo para contador rápido, Activar contador rápido.**



La operación **Definir modo para contador rápido** asigna un modo (MODE) al contador rápido direccionado (HSC). La operación **Activar contador rápido** configura y controla el funcionamiento del contador rápido direccionado, basándose en el estado de las marcas especiales del mismo. El parámetro N indica el número del contador rápido, los CPUs 221 y 222 no soportan los contadores HSC1 y HSC2.

- **Descripción de las operaciones con contadores rápidos.**

Los contadores rápidos cuentan eventos que se ejecutan mucho más rápido de lo que es posible controlarlos en los ciclos del CPU. Estos contadores se pueden configurar como máximo para 12 modos de operación diferentes. La frecuencia máxima de un contador rápido depende del tipo de CPU, todos los contadores disponen de entradas que soportan funciones tales como relojes, control del sentido, puesta a 0 y arranque, para los contadores de dos fases, ambos relojes pueden funcionar a máxima frecuencia. En el caso de los contadores A/B, se puede elegir entre una velocidad máxima de conteo simple (1x) o cuádruple (4x). Todos los contadores funcionan a velocidades máximas sin interferirse mutuamente.

- **Utilización de los contadores rápidos.**

Los contadores rápidos se utilizan habitualmente como accionamiento para temporizadores que funcionan impulsados por un árbol que gira a un régimen constante y provisto de un encoder incremental. Éste último provee un número determinado de valores de conteo por giro, así como un impulso de puesta a 0 una vez por giro. El reloj (o relojes) y el impulso de puesta a 0 del encoder suministran las entradas para el contador rápido. El primero de los valores predeterminados se carga en el contador y las salidas deseadas se activan para el intervalo de tiempo en que el valor actual del contador es menor que el valor predeterminado. El contador se ajusta para que una interrupción se active cuando el conteo actual sea igual al predeterminado o cuando el contador se ponga a 0, cuando el valor actual es igual al predeterminado y se presenta un evento de

interrupción, entonces se carga un nuevo valor predeterminado y se activa el siguiente estado de señal para las salidas. Si se produce un evento de interrupción porque el contador se ha inicializado, entonces se ajusta el primer valor predeterminado y los primeros estados de las salidas, repitiéndose el ciclo.

Puesto que las interrupciones se producen a una velocidad muy inferior a la de los contadores rápidos, es posible implementar un control preciso de las operaciones rápidas con un impacto relativamente bajo en el ciclo total del sistema.

- **Conectar el cableado de las entradas de los contadores rápidos.**

La tabla 1.4 muestra las entradas correspondientes al reloj, el control del sentido, la puesta a 0 y las funciones de arranque de los contadores rápidos. Estas funciones de entrada y los modos de operación de los contadores rápidos se describen desde las tablas 1.6 a 1.11.

Contador rápido	Entradas utilizadas
HSC0	I0.0, I0.1, 0.2
HSC1	I0.6, I0.7, I1.0, I1.1
HSC2	I1.2, I1.3, I1.4, I1.5
HSC3	I0.1
HSC4	I0.3, I0.4, I0.5
HSC5	I0.4

Tabla 1.4 Entradas para los contadores rápidos

Como muestra el área sombreada de la tabla 1.5, la asignación de entradas de algunos contadores rápidos se solapa con las interrupciones de flanco. Una misma entrada no se puede utilizar para dos funciones diferentes. No obstante, cualquier entrada que no se esté utilizando en el modo actual del contador rápido se puede usar para otro fin. Por ejemplo, si HSC0 se está utilizando en modo 2 (que utiliza las entradas I0.0 e I0.2), I0.1 se podrá utilizar para interrupciones de flanco o para HSC3, si se utiliza un modo de HSC0 que no use la entrada I0.1, ésta se podrá emplear para HSC3 o para interrupciones de flanco. De forma similar, si I0.2 no se utiliza en el modo de HSC0 seleccionado, la entrada estará disponible para interrupciones de flanco. Asimismo, si I0.4 no se usa en el modo de HSC4 seleccionado, la entrada se podrá utilizar para HSC5.

Es preciso tener en cuenta que todos los modos de HSC0 utilizan siempre I0.0 y que todos los de HSC4 usan siempre I0.3. Por tanto, dichas entradas nunca estarán disponibles para otros fines cuando se estén utilizando estos contadores.

Elemento	Entrada (I)														
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	1.0	1.1	1.2	1.3	1.4	1.5	
HSC0	X	x	x												
HSC1							x	x	x	x					
HSC2											x	x	x	X	

HSC3		x												
HSC4				X	x	x								
HSC5					x									
Interrupciones de flanco	X	x	x	X										

Tabla 1.5 Asignación de entradas para los contadores rápidos y las interrupciones de flanco

HSC0					
Modo	Descripción	I0.0	I0.1	I0.2	
0	Contador adelante/atrás de fase simple con control interno del sentido de conteo.	Reloj		Puesta a 0	
1	SM37.3 = 0, conteo atrás SM37.3 = 1, conteo adelante				
3	Contador adelante/atrás de fase simple con control externo del sentido de conteo.	Reloj	Sentido	Puesta a 0	
4	I0.1 = 0, conteo atrás I0.1 = 1, conteo adelante				
6	Contador de dos fases con entradas de reloj para conteo adelante/atrás	Reloj (adelante)	Reloj (atrás)	Puesta a 0	
7					
9	Contador A/B la fase A está desfasada 90 grados respecto a B en sentido horario; la fase B está desfasada 90 grados respecto a A en sentido antihorario.	Reloj (fase A)	Reloj (fase B)	Puesta a 0	
10					

Tabla 1.6 Modos de operación de HSC0 (CPU 221, CPU 222, CPU 224 y CPU 226)

HSC1							
Modo	Descripción	I0.6	I0.7	I1.0	I1.1		
0	Contador adelante/atrás de fase simple con control interno del sentido de conteo.	Reloj		Puesta a 0			
1							
2	SM47.3 = 0, conteo atrás SM47.3 = 1, conteo adelante	Reloj	Sentido	Puesta a 0		Arranque	
3	Contador adelante/atrás de fase simple con control externo del sentido de conteo.						
4	I0.7 = 0, conteo atrás	Reloj (adelante)	Reloj (atrás)	Puesta a 0			
5	I0.7 = 1, conteo adelante						
6	Contador de dos fases con entradas de reloj para conteo adelante/atrás	Reloj (adelante)	Reloj (atrás)	Puesta a 0			Arranque
7							
8							Arranque
9	Contador A/B la fase A está desfasada 90 grados respecto a B en sentido horario; la fase B está desfasada 90 grados respecto a A en sentido	Reloj (fase A)	Reloj (fase B)	Puesta a 0			
10							
11					Arranque		

	antihorario.				
--	--------------	--	--	--	--

Tabla 1.7 Modos de operación de HSC1 (CPU 224 y CPU 226)

HSC2					
Modo	Descripción	I1.2	I1.3	I1.4	I1.5
0	Contador adelante/atrás de fase simple con control interno del sentido de conteo. SM57.3 = 0, Conteo atrás SM57.3 = 1, Conteo adelante	Reloj			
1				Puesta a 0	
2				Arranque	
3	Contador adelante/atrás de fase simple con control externo del sentido de conteo. I1.3 = 0, conteo atrás I1.3 = 1, conteo adelante	Reloj	Sentido		
4				Puesta a 0	
5				Arranque	
6	Contador de dos fases con entradas de reloj para conteo adelante/atrás	Reloj (adelante)	Reloj (atrás)		
7				Puesta a 0	
8				Arranque	
9	Contador A/B la fase A está desfasada 90 grados respecto a B en sentido horario; la fase B está desfasada 90 grados respecto a A en sentido antihorario.	Reloj (fase A)	Reloj (fase B)		
10				Puesta a 0	
11				Arranque	

Tabla 1.8 Modos de operación de HSC2 (CPU 224 y CPU 226)

HSC3					
Modo	Descripción	I0.1			
0	Contador adelante/atrás de fase simple con control interno del sentido de conteo. SM137.3 = 0, conteo atrás SM137.3 = 1, conteo adelante	Reloj			

Tabla 1.9 Modos de operación de HSC3 (CPU 221, CPU 222, CPU 224 y CPU 226)

HSC4					
Modo	Descripción	I0.3	I0.4	I0.5	
0	Contador adelante/atrás de fase simple con control interno del sentido de conteo. SM147.3 = 0, conteo atrás SM147.3 = 1, conteo adelante	Reloj			
1				Puesta a 0	
3	Contador adelante/atrás de fase simple con control externo del sentido de conteo. I0.4 = 0, conteo atrás I0.4 = 1, conteo adelante	Reloj	Sentido		
4				Puesta a 0	
6	Contador de dos fases con entradas de reloj para conteo adelante/atrás	Reloj (adelante)	Reloj (atrás)		
7				Puesta a 0	

9	Contador A/B la fase A está desfasada 90 grados respecto a B en sentido horario;	Reloj (fase A)	Reloj (fase B)		
10	la fase B está desfasada 90 grados respecto a A en sentido antihorario.			Puesta a 0	

Tabla 1.10 Modos de operación de HSC4 (CPU 221, CPU 222, CPU 224 y CPU 226)

HSC5					
Modo	Descripción	I0.4			
0	Contador adelante/atrás de fase simple con control interno del sentido de conteo. SM157.3 = 0, conteo atrás SM157.3 = 1, conteo adelante	Reloj			

Tabla 1.11 Modos de operación de HSC5 (CPU 221, CPU 222, CPU 224 y CPU 226)

- **Direccionamiento de los contadores rápidos (HC).**

Para acceder al valor de conteo del contador rápido, se indica la dirección del mismo (utilizando el identificador HC) y el número del contador (p.ej. HC0). El valor actual del contador rápido es de sólo lectura, pudiéndose acceder al mismo sólo en formato de palabra doble (32 bits).

- **Descripción de los diferentes contadores rápidos**

Todos los contadores funcionan de la misma manera en el mismo modo de operación.

Como muestra la tabla 5, hay cuatro tipos básicos de contadores. Es preciso tener en cuenta que no todos los contadores soportan todos los modos. Todos los contadores se pueden utilizar sin entrada de puesta a 0 ni de Arranque, con entrada de puesta a 0 pero sin entrada de arranque, o bien, con entrada de puesta a 0 y de arranque.

Activando la entrada de puesta a 0 se borra el valor actual del contador hasta que vuelve a ser desactivada. Al activarse la entrada de arranque se habilita el contador. Si se desactiva esta entrada se mantiene el valor actual del contador, ignorándose los eventos de reloj. Si se activa la entrada de puesta a 0 mientras está desactivada la entrada del arranque, se ignorará la activación de la entrada de puesta a 0, con lo que no se modificará el valor actual.

Si la entrada de arranque se activa mientras está activada la entrada de puesta a 0, el valor actual se borrará, antes de poder utilizar un contador rápido es preciso elegir su modo de operación. Para ello se utiliza la operación HDEF (Definir modo para contador rápido). HDEF establece la conexión entre un contador rápido (HSCx) y el modo de conteo. Por cada contador sólo se puede ejecutar una operación HDEF. Un contador rápido se define utilizando la marca del primer ciclo SM0.1 (este bit se activa sólo en el primer ciclo y se desactiva posteriormente) para llamar a la subrutina que contiene la operación HDEF.

- **Elegir el nivel de actividad y el modo de conteo simple o cuádruple.**

Cuatro contadores tienen tres marcas de control que se utilizan para configurar el estado activo de las entradas de puesta a 0 y de arranque, así como para seleccionar la velocidad simple o cuádruple (esto sólo en los contadores A/B). Estas marcas están depositadas en el byte de control del respectivo contador y se emplean solamente cuando se ejecuta la operación HDEF. La tabla 1.12 muestra las marcas.

Antes de poder ejecutar la operación HDEF es preciso ajustar las marcas de control de HSC1 y HSC2 al estado deseado. De lo contrario, el contador adoptará la configuración predeterminada del modo de conteo elegido. El ajuste estándar de las entradas de puesta a 0 y de arranque es de actividad alta, y la velocidad de conteo es la cuádruple (es decir, la frecuencia del reloj de entrada multiplicada por cuatro). Una vez ejecutada la operación HDEF, ya no se podrá modificar el ajuste de los contadores, a menos que la CPU se cambie a modo STOP.

HSC0	HSC1	HSC2	HSC4	Descripción (sólo cuando se ejecuta HDEF)
SM37.0	SM47.0	SM57.0	SM147.0	Bit de control para nivel de actividad de la entrada de puesta a 0: 0 = actividad alta; 1 = actividad baja
_____	SM47.1	SM57.1	_____	Bit de control para nivel de actividad de la entrada de arranque: 0 = actividad alta; 1 = actividad baja
SM37.2	SM47.2	SM57.2	SM147.2	Velocidad de conteo de los contadores A/B: 0 = velocidad cuádruple; 1 = velocidad simple

Tabla 1.12 Nivel de actividad de las entradas de puesta a 0 y de arranque, marcas para elegir la velocidad simple o cuádruple

- **Byte de control**

Una vez definido el contador y el modo de conteo se deben programar los parámetros dinámicos del mismo. Todos los contadores rápidos disponen de un byte que los habilita o inhibe, fijando el sentido de control (sólo en los modos 0, 1 y 2). El byte de control determina asimismo el sentido de conteo inicial para todos los modos restantes, así como el valor actual y el valor predeterminado que se cargarán. El byte de control, los valores actuales asignados y los valores predeterminados se comprueban al ejecutarse la operación HSC. La tabla 1.13 describe cada una de las marcas del byte de control.

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	Descripción
SM37.3	SM47.3	SM57.3	SM137.3	SM147.3	SM157.3	Bit de control para el sentido de conteo: 0 = conteo atrás; 1 = conteo adelante
SM37.4	SM47.4	SM57.4	SM137.4	SM147.4	SM157.4	Escribir el sentido de conteo del contador rápido: 0 = no actualizar; 1 = actualizar el valor predeterminado
SM37.5	SM47.5	SM57.5	SM137.5	SM147.5	SM157.5	Escribir el nuevo valor predeterminado en el contador rápido: 0 = no actualizar; 1 = actualizar el valor predeterminado
SM37.6	SM47.6	SM57.6	SM137.6	SM147.6	SM157.6	Escribir el nuevo valor actual en el contador rápido: 0 = no actualizar; 1 = actualizar el valor actual

SM37.7	SM47.7	SM57.7	SM137.7	SM147.7	SM157.7	Habilitar el contador rápido: 0 = inhibir el contador rápido; 1 = habilitar el contador rápido
--------	--------	--------	---------	---------	---------	--

Tabla 1.13 Marcas de control de HSC0, HSC1 y HSC2.

- **Ajustar los valores actuales y predeterminados**

Todos los contadores rápidos disponen de un valor actual y de un valor predeterminado de 32 bits. Ambos son valores enteros con signo. Para cargar un nuevo valor actual o predeterminado en el contador rápido es preciso activar el byte de control y los bytes de las marcas especiales que contienen los valores actuales y/o predeterminados. Después se ejecuta la operación HSC para transferir los nuevos valores al contador rápido. La tabla 13 describe los bytes de marcas especiales que contienen los nuevos valores y los valores predeterminados.

Además de los bytes de control y de los bytes que contienen los nuevos valores predeterminados, también se puede leer el valor actual de cada uno de los contadores rápidos, utilizando el tipo de datos HC (valor actual del contador rápido) seguido del número de contador (0, 1, 2, 3, 4 o 5). Ello permite acceder directamente al valor actual para operaciones de lectura.

Por el contrario, este valor sólo se puede escribir utilizando la operación HSC que se describe más arriba.

Valor a cargar	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
Nuevo actual	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
Nuevo valor predeterminado	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162

Tabla 1.14 Valores actuales y predeterminado de los contadores HSC0, HSC1, HSC2, HSC3, HSC4 y HSC5

- **Byte de estado**

Todos los contadores rápidos disponen de un byte para marcas de estado. Éstas indican el sentido de conteo actual y si el valor actual es igual o mayor que el valor predeterminado, en la tabla 1.15 se muestra las marcas de estado de los contadores rápidos.

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	Descripción
SM36.0	SM46.0	SM56.0	SM136.0	SM146.0	SM156.0	No utilizado
SM36.1	SM46.1	SM56.1	SM136.1	SM146.1	SM156.1	No utilizado
SM36.2	SM46.2	SM56.2	SM136.2	SM146.2	SM156.2	No utilizado
SM36.3	SM46.3	SM56.3	SM136.3	SM146.3	SM156.3	No utilizado
SM36.4	SM46.4	SM56.4	SM136.4	SM146.4	SM156.4	No utilizado
SM36.5	SM46.5	SM56.5	SM136.5	SM146.5	SM156.5	Bit de estado del sentido de conteo actual: 0 = conteo atrás; 1 = conteo adelante

SM36.6	SM46.6	SM56.6	SM136.6	SM146.6	SM156.6	El valor actual es igual al bit de estado del valor predeterminado: 0 = diferente; 1 = igual
SM36.7	SM46.7	SM56.7	SM136.7	SM146.7	SM156.7	El valor actual es mayor que el bit de estado del valor predeterminado: 0 = menor o igual; 1 = mayor que

Tabla 1.15 Marcas de estado de los contadores HSC0, HSC1, HSC2, HSC3, HSC4 y HSC5

- **Interrupciones de los contadores rápidos**

Todos los modos de los contadores soportan una interrupción si el valor actual es igual al valor predeterminado. Los modos de los contadores que utilizan una entrada de puesta a 0 externa soportan una interrupción que se ejecuta cuando se activa dicha entrada. Todos los modos de conteo (con excepción de los modos 0, 1 y 2) soportan una interrupción que se ejecuta cuando se produce un cambio del sentido de conteo. Cada una de estas condiciones puede habilitarse o inhibirse por separado.

- **Modos de inicialización 0, 1 ó 2**

Para inicializar HSC1 como contador adelante/atrás de fase simple con control interno del sentido de conteo (modos 0, 1 ó 2):

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB47 en la subrutina de inicialización conforme a la operación de control deseada.
Por ejemplo:
SMB47 = 16#F8 Resultados:
 - Se habilita el contador.
 - Se escribe un nuevo valor actual.
 - Se escribe un nuevo valor predeterminado.
 - Se ajusta el sentido de conteo adelante.
 - Se ajusta la actividad alta de las entradas, arranque y puesta a 0.
3. Ejecute la operación HDEF con la entrada HSC puesta a 1 y la entrada MODE a 0 para puesta a 0 o arranque no externos, o bien a 1 para puesta a 0 externa y sin arranque, o bien a 2 para puesta a 0 y arranque externos.
4. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrarla marca).
5. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).

6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción.
7. Para poder detectar una puesta a 0 externa, programe una interrupción asociando el evento de interrupción Puesta a 0 externa (evento 15) a una rutina de interrupción.
8. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
9. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.
10. Finalice la subrutina.

- **Modos de inicialización 3, 4 ó 5.**

Para inicializar HSC1 como contador adelante/atrás de fase simple con control externo del sentido de conteo (modos 3, 4 ó 5):

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB47 en la subrutina de inicialización conforme a la operación de control deseada.
Por ejemplo:
SMB47 = 16#F8 Resultados:
 - Se habilita el contador.
 - Se escribe un nuevo valor actual.
 - Se escribe un nuevo valor predeterminado.
 - Se ajusta el sentido inicial de conteo adelante.
 - Se ajusta la actividad alta de las entradas, arranque y puesta a 0.
3. Ejecute la operación HDEF con la entrada HSC puesta a 1 y la entrada MODE a 3 para puesta a 0 o arranque no externos, o bien a 4 para puesta a 0 externa y sin arranque, o bien a 5 para puesta a 0 y arranque externos.
4. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
5. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).
6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción.

7. Para poder detectar un cambio del sentido de conteo, programe una interrupción asociando el evento de interrupción de cambio de sentido (evento 14) a una rutina de interrupción.
8. Para poder detectar una puesta a 0 externa, programe una interrupción asociando el evento de interrupción Puesta a 0 externa (evento 15) a una rutina de interrupción.
9. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
10. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.
11. Finalice la subrutina.

- **Modos de inicialización 6, 7 u 8.**

Para inicializar HSC1 como contador adelante/atrás de dos fases con relojes adelante/atrás (modos 6, 7 u 8):

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB47 en la subrutina de inicialización conforme a la operación de control deseada.
Por ejemplo:
SMB47 = 16#F8 Resultados:
 - Se habilita el contador.
 - Se escribe un nuevo valor actual.
 - Se escribe un nuevo valor predeterminado.
 - Se ajusta el sentido inicial de conteo adelante.
 - Se ajusta la actividad alta de las entradas, arranque y puesta a 0.
3. Ejecute la operación HDEF con la entrada HSC puesta a 1 y la entrada MODE a 6 para puesta a 0 o arranque no externos, o bien a 7 para puesta a 0 externa y sin arranque, o bien a 8 para puesta a 0 y arranque externos.
4. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
5. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).
6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción.

7. Para poder detectar un cambio del sentido de conteo, programe una interrupción asociando el evento de interrupción de cambio de sentido (evento 14) a una rutina de interrupción.
8. Para poder detectar una puesta a 0 externa, programe una interrupción asociando el evento de interrupción Puesta a 0 externa (evento 15) a una rutina de interrupción.
9. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
10. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.
11. Finalice la subrutina.

- **Modos de inicialización 9, 10 u 11.**

Para inicializar HSC1 como contador A/B (modos 9, 10 u 11):

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización, puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB47 en la subrutina de inicialización conforme a la operación de control deseada, ejemplo (frecuencia simple):

SMB47 = 16#FC Resultados:

Se habilita el contador.

Se escribe un nuevo valor actual.

Se escribe un nuevo valor predeterminado.

Se ajusta el sentido inicial de conteo adelante.

Se ajusta la actividad alta de las entradas, arranque y puesta a 0.

Ejemplo (frecuencia cuádruple):

SMB47 = 16#F8 Resultados:

Se habilita el contador.

Se escribe un nuevo valor actual.

Se escribe un nuevo valor predeterminado.

Se ajusta el sentido inicial de conteo adelante.

Se ajusta la actividad alta de las entradas, arranque y e puesta a 0.

3. Ejecute la operación HDEF con la entrada HSC puesta a 1 y la entrada MODE a 9 para puesta a 0 o arranque no externos, o bien a 10 para puesta a 0 externa y sin arranque, o bien a 11 para puesta a 0 y arranque externos.

4. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
5. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).
6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción.
7. Para poder detectar un cambio del sentido de conteo, programe una interrupción asociando el evento de interrupción de cambio de sentido (evento 14) a una rutina de interrupción.
8. Para poder detectar una puesta a 0 externa, programe una interrupción asociando el evento de interrupción Puesta a 0 externa (evento 15) a una rutina de interrupción.
9. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
10. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.
11. Finalice la subrutina.

- **Cambio de sentido en los modos 0, 1 ó 2.**

Para configurar el cambio de sentido de HSC1 como contador de fase simple con control interno del sentido de conteo (modos 0,1 ó 2):

1. Cargue SMB47 para escribir la dirección deseada:
SMB47 = 16#90 Habilita el contador.
Ajusta el sentido de conteo atrás.
SMB47 = 16#98 Habilita el contador.
Ajusta el sentido de conteo adelante.
2. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.

- **Cargar nuevo valor actual (cualquier modo)**

Si se modifica el valor actual, el contador se inhibirá automáticamente. Mientras está inhibido el contador, no cuenta ni tampoco se generan interrupciones. Para modificar el valor actual del contador HSC1 (cualquier modo):

1. Cargue SMB47 para escribir el valor actual deseado:
SMB47 = 16#C0 Habilita el contador.

Escribe el nuevo valor actual.

2. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
3. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.

- **Cargar nuevo valor predeterminado (cualquier modo)**

Para modificar el valor predeterminado de HSC1 (cualquier modo):

1. Cargue SMB47 para escribir el valor predeterminado deseado: SMB47 = 16#A0 Habilita el contador.
Escribe el nuevo valor predeterminado.
2. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).
3. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.

- **Inhibir un contador rápido (cualquier modo).**

Para inhibir el contador rápido HSC1 (cualquier modo):

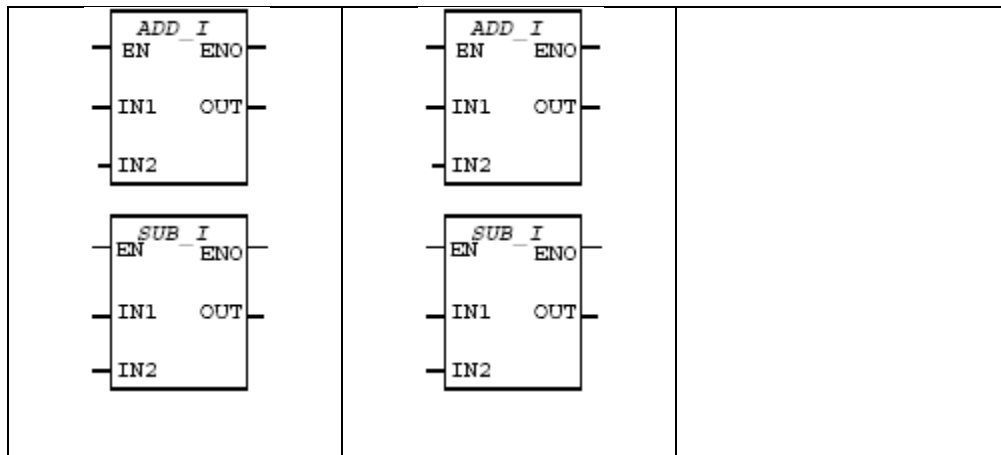
1. Cargue SMB47 para inhibir el contador:
SMB47 = 16#00 Inhibe el contador.
2. Ejecute la operación HSC para inhibir el contador.

Arriba se describe cómo modificar de forma individual el sentido de conteo, el valor actual o el valor predeterminado. No obstante, también se pueden cambiar todos o sólo algunos de estos ajustes en ese mismo orden, definiendo el valor de SMB47 de forma apropiada y ejecutando luego la operación HSC.

1.3.2.5 Operaciones aritméticas con enteros (SIMATIC).

- **Sumar y restar enteros de 16 bits.**

KOP	FUP	AWL
		+I IN1, OUT
		-I IN1, OUT



Las operaciones **Sumar enteros de 16 bits** y **Restar enteros de 16 bits** suman/restan dos enteros de 16 bits, arrojando un resultado de 16 bits (OUT).

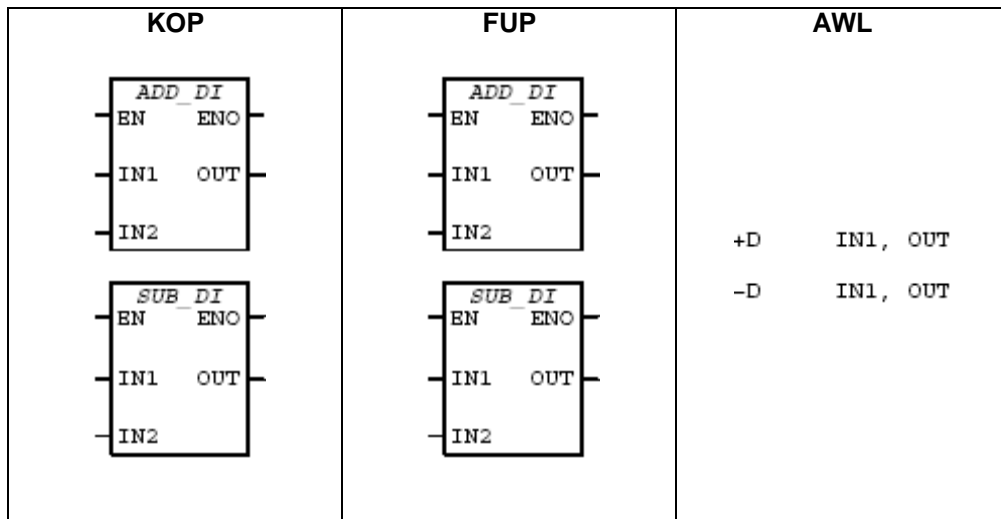
En KOP y FUP: $IN1 + IN2 = OUT$

$$IN1 - IN2 = OUT$$

En AWL: $IN1 + OUT = OUT$

$$OUT - IN1 = OUT$$

- **Sumar y restar enteros de 32 bits.**



Las operaciones **Sumar enteros de 32 bits** y **Restar enteros de 32 bits** suman/restan dos enteros de 32 bits, arrojando un resultado de 32 bits (OUT).

En KOP y FUP: $IN1 + IN2 = OUT$

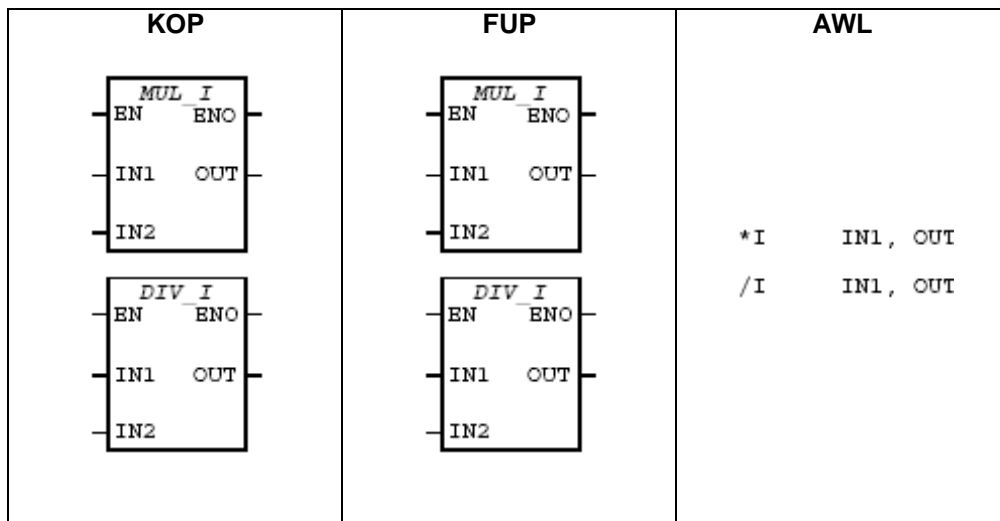
$$IN1 - IN2 = OUT$$

En AWL: $IN1 + OUT = OUT$

$$\text{OUT} - \text{IN1} = \text{OUT}$$

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo).

- **Multiplicar y dividir enteros de 16 bits.**



La operación **Multiplicar enteros de 16 bits** multiplica dos números enteros de 16 bits, arrojando un producto de 16 bits.

La operación **Dividir enteros de 16 bits** divide dos números enteros de 16 bits, arrojando un cociente de 16 bits. No se guarda ningún resto.

La marca de desbordamiento se activa si el resultado es mayor que una salida de palabra.

En KOP y FUP: $\text{IN1} * \text{IN2} = \text{OUT}$

$$\text{IN1} / \text{IN2} = \text{OUT}$$

En AWL:

$$\text{IN1} * \text{OUT} = \text{OUT}$$

$$\text{OUT} / \text{IN1} = \text{OUT}$$

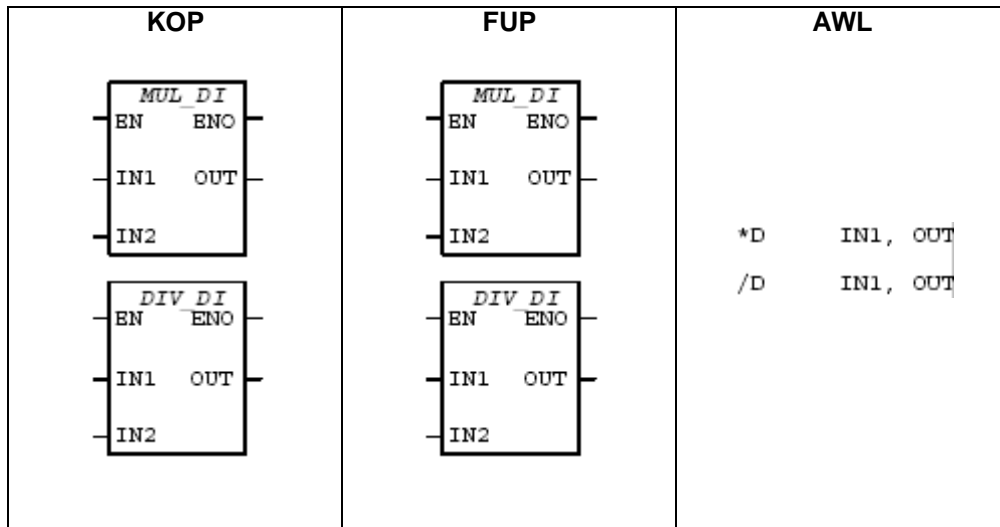
Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo); SM1.3 (división por cero).

Si SM1.1 (marca de desbordamiento) se activa durante una operación de multiplicación o de división, no se escribe en la salida y todos los demás bits de estado aritméticos se ponen a 0.

Si se activa SM1.3 (división por cero) durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado, así como los operandos de entrada originales.

En otro caso, todos los bits aritméticos de estado soportados contendrán el estado válido al finalizar la operación aritmética.

- **Multiplicar y dividir enteros de 32 bits**



La operación **Multiplicar enteros de 32 bits** multiplica dos enteros de 32 bits, arrojando un producto de 32 bits. La operación **Dividir enteros de 32 bits** divide dos enteros de 32 bits, arrojando un cociente de 32 bits. No se guarda ningún resto.

En KOP y FUP: $IN1 * IN2 = OUT$
 $IN1 / IN2 = OUT$

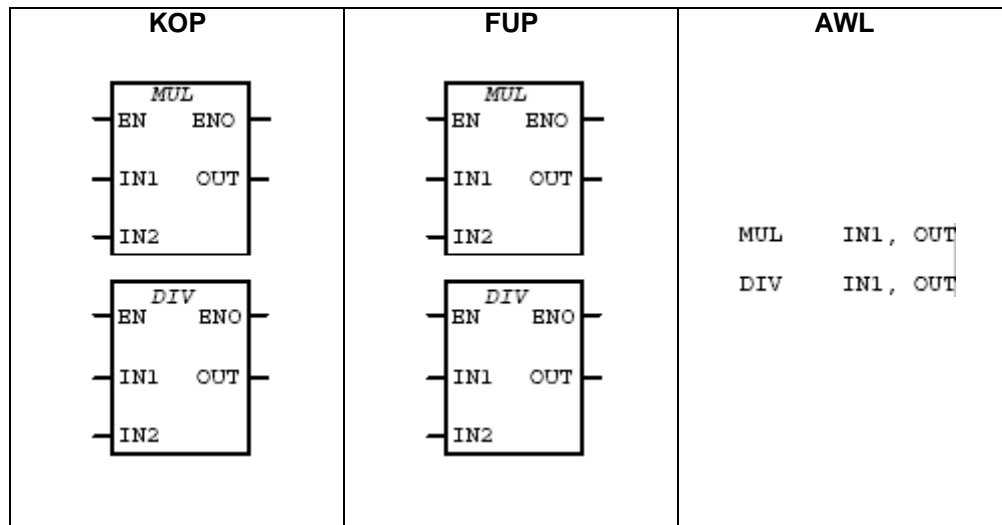
En AWL: $IN1 * OUT = OUT$
 $OUT / IN1 = OUT$

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo); SM1.3 (división por cero) Si SM1.1 (marca de desbordamiento) se activa durante una operación de multiplicación o de división, no se escribe en la salida y todos los demás bits de estado aritméticos se ponen a 0.

Si se activa SM1.3 (división por cero) durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado, así como los operandos de entrada originales.

En otro caso, todos los bits aritméticos de estado soportados contendrán el estado válido al finalizar la operación aritmética.

- **Multiplicar y dividir enteros de 16 bits a enteros de 32 bits.**



La operación **Multiplicar enteros de 16 bits a enteros de 32 bits** multiplica dos números enteros de 16 bits, arrojando un producto de 32 bits.

La operación **Dividir enteros de 16 bits a enteros de 32 bits** divide dos números enteros de 16 bits, arrojando un resultado de 32 bits compuesto de un cociente de 16 bits (los menos significativos) y un resto de 16 bits (los más significativos).

En la operación AWL de multiplicación, la palabra menos significativa (16 bits) del OUT de 32 bits se utiliza como uno de los factores, en la operación AWL de división, la palabra menos significativa (16 bits) del OUT de 32 bits se utiliza como dividendo.

En KOP y FUP: $IN1 * IN2 = OUT$
 $IN1 / IN2 = OUT$

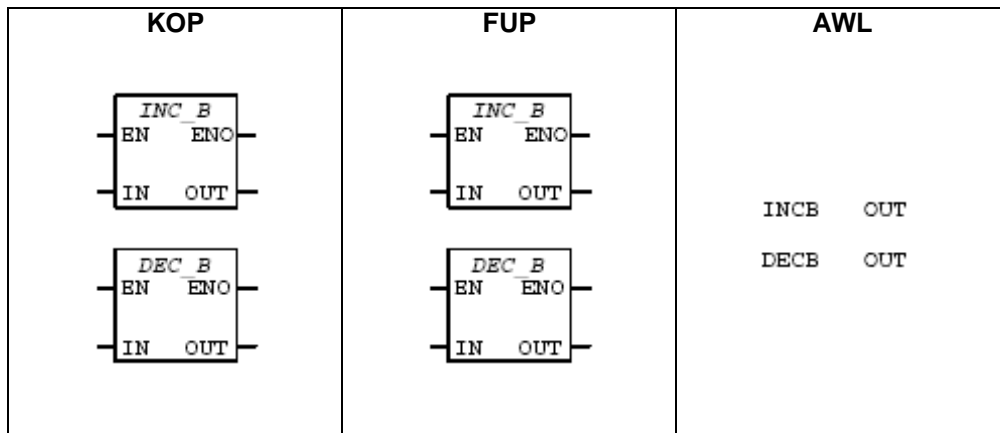
En AWL: $IN1 * OUT = OUT$
 $OUT / IN1 = OUT$

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo); SM1.3 (división por cero).

Si se activa SM1.3 (división por cero) durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado, así como los operandos de entrada originales.

En otro caso, todos los bits aritméticos de estado soportados contendrán el estado válido al finalizar la operación aritmética.

- **Incrementar y Decrementar byte.**



Las operaciones **Incrementar byte** y **Decrementar byte** suman/restan 1 al byte de entrada (IN) y depositan el resultado en la variable indicada por OUT.

Las operaciones Incrementar byte y Decrementar byte no llevan signo.

En KOP y FUP: $IN + 1 = OUT$

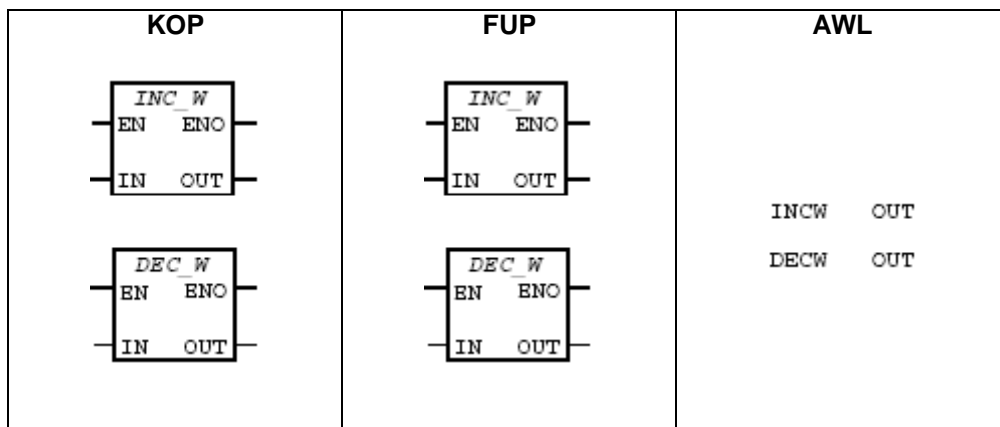
$IN - 1 = OUT$

En AWL: $OUT + 1 = OUT$

$OUT - 1 = OUT$

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento).

- **Incrementar y decrementar palabra.**



Las operaciones **Incrementar palabra** y **Decrementar palabra** suman/restan 1 al valor de la palabra de entrada (IN) y depositan el resultado en OUT.

Las operaciones Incrementar palabra y Decrementar palabra llevan signo (16#7FFF > 16#8000).

En KOP y FUP: $IN + 1 = OUT$

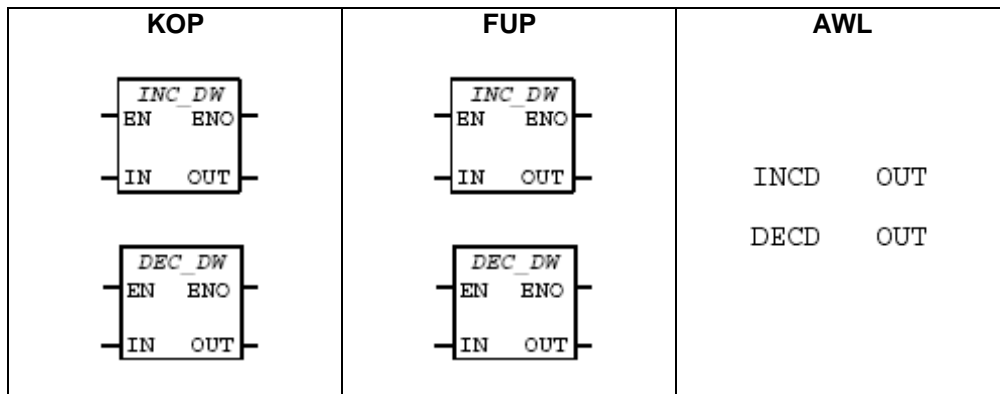
$IN - 1 = OUT$

En AWL: $OUT + 1 = OUT$

$OUT - 1 = OUT$

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo).

- **Incrementar y decrementar palabra doble**



Las operaciones **Incrementar palabra doble** y **Decrementar palabra doble** suman/restan 1 al valor de la palabra doble de entrada (IN) y depositan el resultado en OUT.

En KOP y FUP: $IN + 1 = OUT$

$IN - 1 = OUT$

Las operaciones Incrementar palabra doble y Decrementar palabra doble llevan signo (16#7FFFFFFF > 16#80000000).

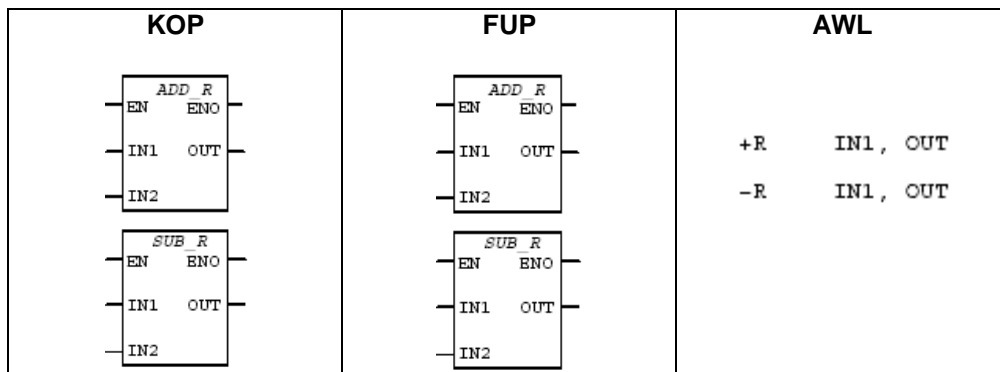
En AWL: $OUT + 1 = OUT$

$OUT - 1 = OUT$

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo).

1.3.2.6 Operaciones aritméticas con números reales (SIMATIC).

- Sumar y restar reales



Las operaciones **Sumar reales** y **Restar reales** suman/restan dos números reales de 32 bits, dando como resultado un número real de 32 bits (OUT).

En KOP y FUP: $IN1 + IN2 = OUT$

$$IN1 - IN2 = OUT$$

En AWL:

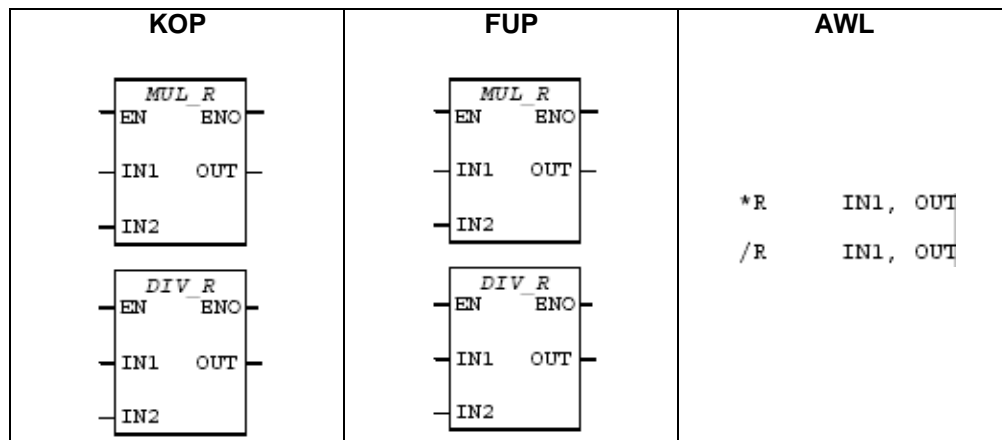
$$IN1 + OUT = OUT$$

$$OUT - IN1 = OUT$$

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo) SM1.1 se utiliza para indicar errores de desbordamiento y valores no válidos.

Si se activa SM1.1, el estado de SM1.0 y de SM1.2 no será válido y no se alterarán los operandos de entrada originales. Si SM1.1 y SM1.2 no se activan durante una operación de división, la operación aritmética habrá finalizado con un resultado válido, y tanto SM1.0 como SM1.2 contendrán un estado válido.

- **Multiplicar y dividir reales.**



La operación **Multiplicar reales** multiplica dos números reales de 32 bits, dando como resultado un número real de 32 bits (OUT).

La operación **Dividir reales** divide dos números reales de 32 bits, dando como resultado un cociente de número real de 32 bits.

En KOP y FUP: $IN1 * IN2 = OUT$

$$IN1 / IN2 = OUT$$

En AWL:

$$IN1 * OUT = OUT$$

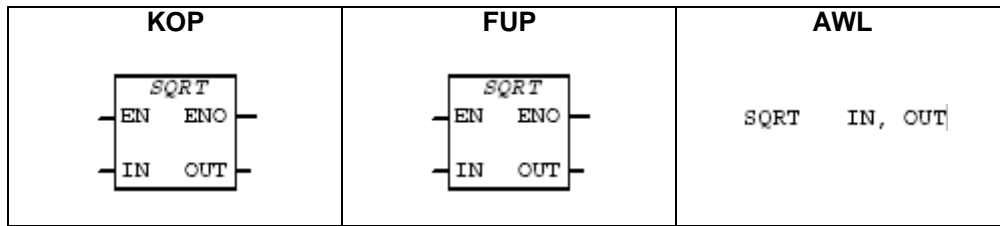
$$OUT / IN1 = OUT$$

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento o valor no válido generado durante la operación o parámetro de entrada no válido); SM1.2 (negativo); SM1.3 (división por cero).

Si se activa SM1.3 durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado, así como los operandos de entrada originales. SM1.1 se utiliza para indicar errores de desbordamiento y valores no válidos. Si se activa SM1.1, el estado de SM1.0 y de SM1.2 no será válido y no se alterarán los operandos de entrada originales. Si SM1.1 y SM1.3 no se activan (durante una operación de división), la operación aritmética habrá finalizado con un resultado válido, y tanto SM1.0 como SM1.2 contendrán un estado válido.

1.3.2.7 Operaciones con funciones numéricas (SIMATIC).

- **Raíz cuadrada.**



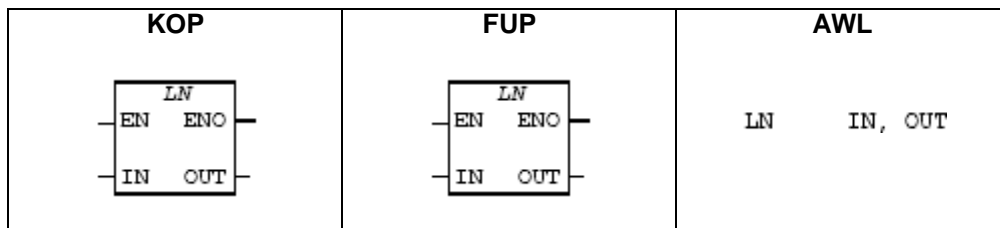
La operación **Raíz cuadrada** extrae la raíz cuadrada de un número real de 32 bits (IN), dando como resultado un número real de 32 bits (OUT), como muestra la ecuación:

$$\sqrt{IN} = OUT$$

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo) SM1.1 sirve para indicar errores de desbordamiento y valores no válidos.

Si se activa SM1.1, el estado de SM1.0 y de SM1.2 no será válido y no se alterarán los operandos de entrada originales. Si SM1.1 y SM1.2 no se activan durante una operación de división, la operación aritmética habrá finalizado con un resultado válido, y tanto SM1.0 como SM1.2 contendrán un estado válido.

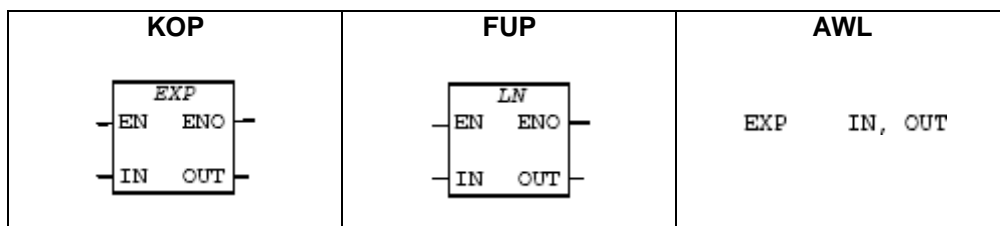
- **Logaritmo natural**



La operación **Logaritmo natural** calcula el logaritmo natural del valor de IN y deposita el resultado en OUT. Para obtener la base 10 del logaritmo natural, utilice la operación Dividir reales (DIV_R (/R)), con objeto de dividir el logaritmo natural por 2,302585 (que es aproximadamente el logaritmo natural de 10).

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo); SM4.3 (división por cero).

- **Exponencial natural**



La operación **Exponencial natural** ejecuta el cálculo exponencial de la constante “e” elevada a la potencia del valor de IN y deposita el resultado en OUT. La operación Exponencial natural se puede utilizar en combinación con la operación Logaritmo natural para elevar cualquier número real a la potencia de otro número real, incluyendo exponentes fraccionarios. Por consiguiente, X elevado a la potencia de Y se puede calcular de la siguiente forma: $EXP(Y * LN(X))$.

Ejemplos:

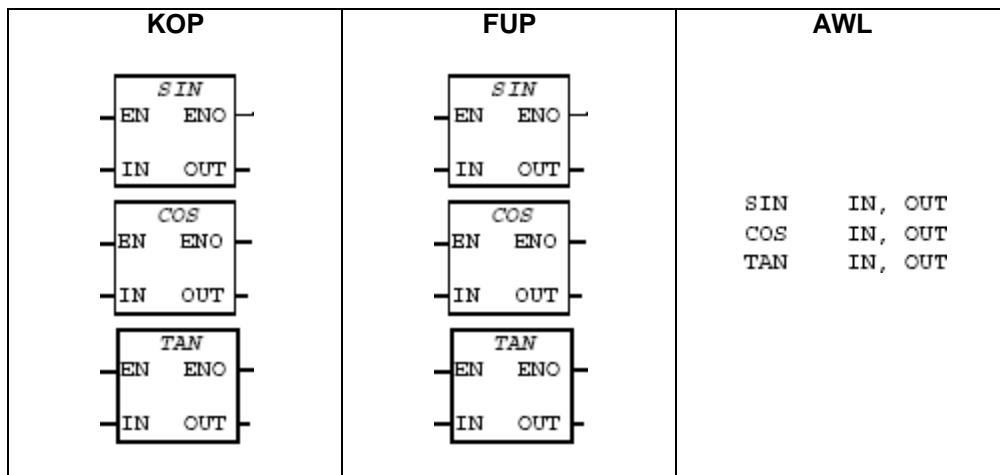
5 elevado al cubo = $5^3 = EXP(3 * LN(5)) = 125$

Raíz cúbica de 125 = $125^{1/3} = EXP(1/3 * LN(125)) = 5$

Raíz cuadrada de 5 elevado al cubo = $5^{3/2} = EXP(3/2 * LN(5)) = 11,18034 \dots$

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo); SM4.3 (división por cero)

- **Seno, Coseno y Tangente**



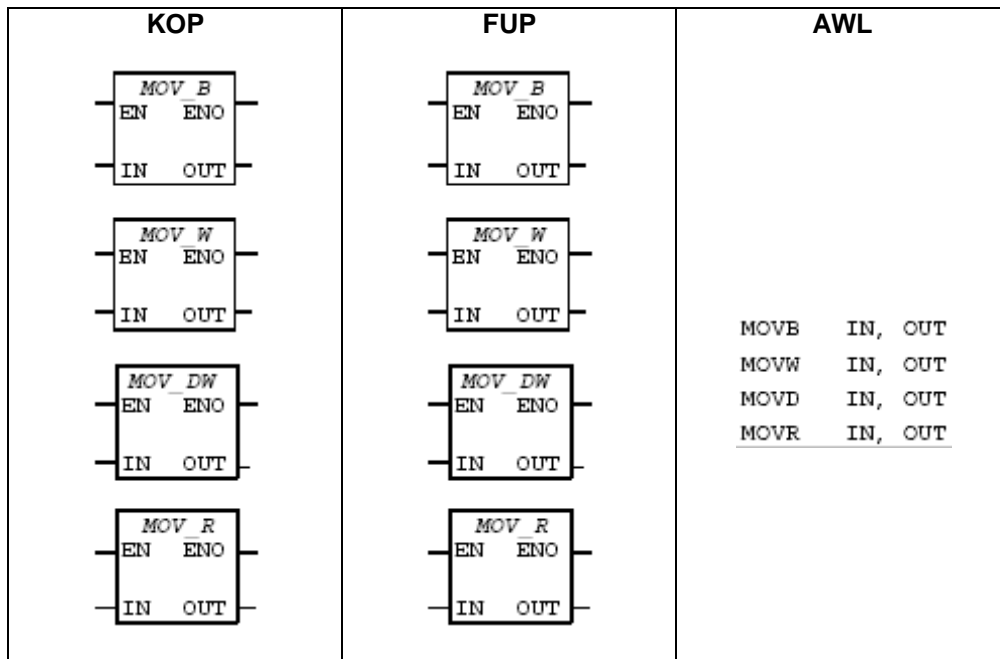
Las operaciones **Seno, Coseno y Tangente** evalúan la función trigonométrica del valor del ángulo IN y depositan el resultado en OUT. El valor del ángulo de entrada se indica en radianes.

Para convertir un ángulo de grados a radianes, utilice la operación Multiplicar reales (MUL_R (*R)) para multiplicar el ángulo en grados por 1,745329E-2 (aproximadamente $\pi/180^\circ$).

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo); SM1.3 (división por cero).

1.3.2.8 Operaciones de transferencia (SIMATIC).

- **Transferir byte, Transferir palabra, Transferir palabra doble y Transferir real.**



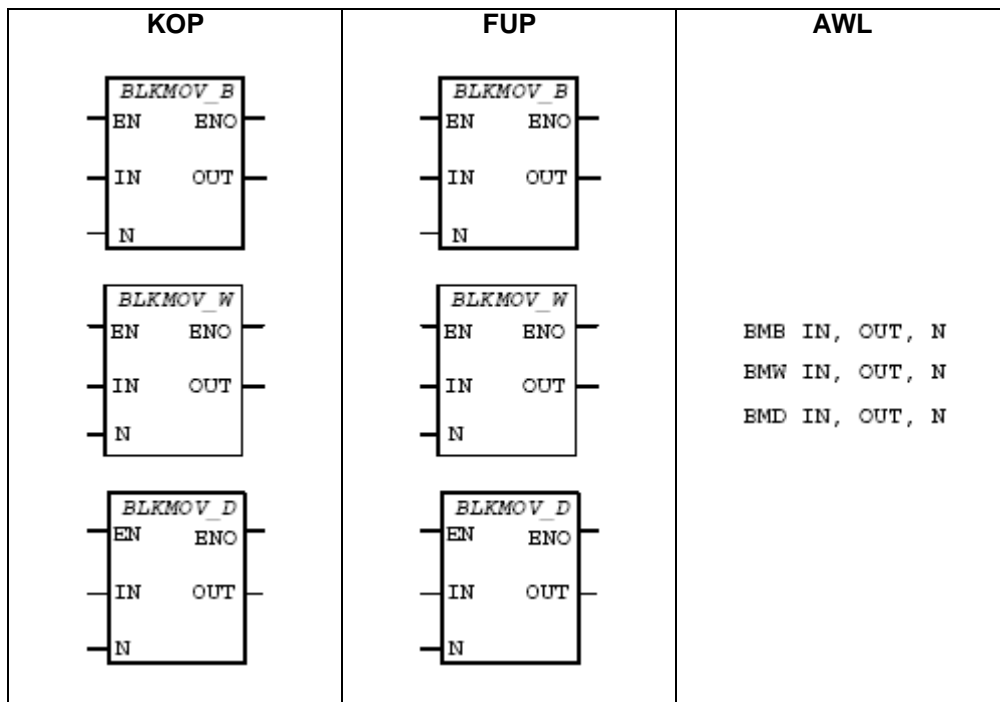
La operación **Transferir byte** transfiere el byte de entrada (IN) al byte de salida (OUT). El byte de entrada permanece inalterado.

La operación **Transferir palabra** transfiere la palabra de entrada (IN) a la palabra de salida (OUT). La palabra de entrada permanece inalterada.

La operación **Transferir palabra doble** transfiere la palabra doble de entrada (IN) a la palabra doble de salida (OUT). La palabra doble de entrada permanece inalterada.

La operación **Transferir real** transfiere un número real de 32 bits de la palabra doble de entrada (IN) a la palabra doble de salida (OUT). La palabra doble de entrada permanece inalterada.

- **Transferir bytes en bloque, Transferir palabras en bloque, Transferir palabras dobles en bloque.**

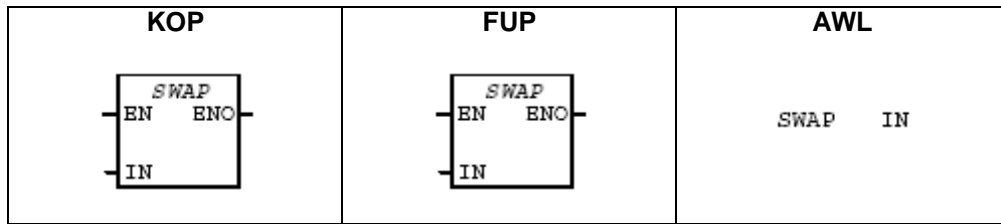


La operación **Transferir bytes en bloque** transfiere un número determinado de bytes (N) de la dirección de entrada IN a la dirección de salida OUT. N puede estar comprendido entre 1 y 255.

La operación **Transferir palabras en bloque** transfiere un número determinado de palabras (N) de la dirección de entrada IN a la dirección de salida OUT. N puede estar comprendido entre 1 y 255.

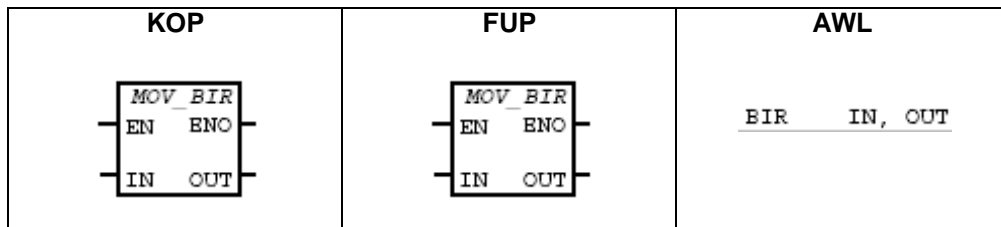
La operación **Transferir palabras dobles en bloque** transfiere un número determinado de palabras dobles (N) de la dirección de entrada IN a la dirección de salida OUT. N puede estar comprendido entre 1 y 255.

- **Invertir bytes de una palabra.**



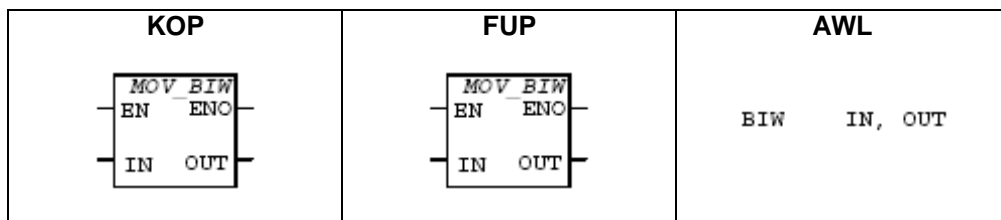
La operación **Invertir bytes de una palabra** intercambia el byte más significativo y el byte menos significativo de una palabra (IN).

- **Lectura directa y transferencia de bytes.**



La operación **Lectura directa y transferencia de bytes** lee la entrada física IN y escribe el resultado en OUT.

- **Escritura directa y transferencia de bytes.**

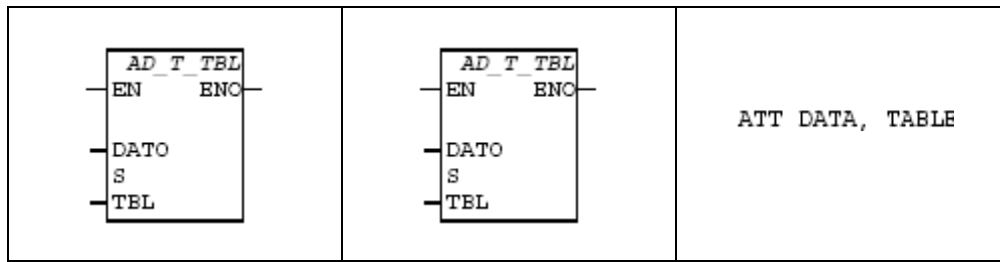


La operación **Escritura directa y transferencia de bytes** lee de la dirección IN y escribe en la salida física OUT.

1.3.2.9 Operaciones de tabla (SIMATIC).

- **Registrar valor en tabla**

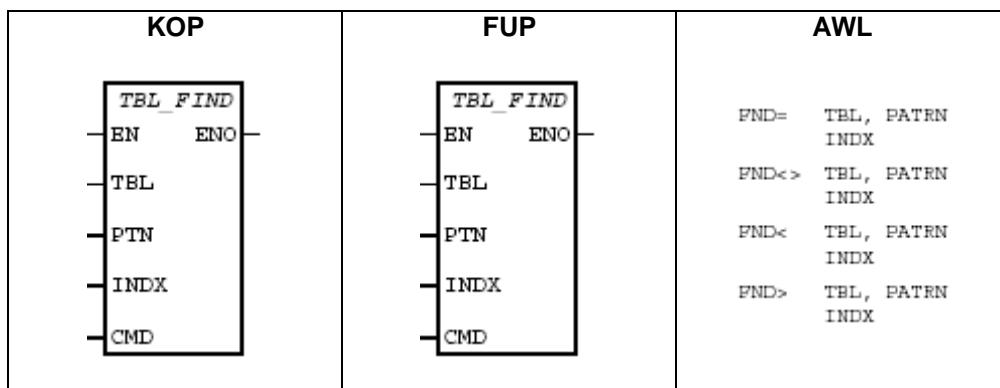




La operación **Registrar valor en tabla** registra valores de palabra (DATA) en la tabla (TBL). El primer valor de la tabla indica la longitud máxima de la misma (TL). El segundo valor (EC) indica el número de registros que contiene la tabla (v. figura 9-31). Los nuevos datos se añaden al final de la tabla, debajo del último registro. Cada vez que se añade un registro a la tabla, se incrementa el número efectivo de registros. Una tabla puede tener como máximo 100 registros.

Estas operaciones afectan a las siguientes marcas especiales: SM1.4 se activa si se intenta introducir demasiados registros en la tabla.

- **Buscar valor en tabla**

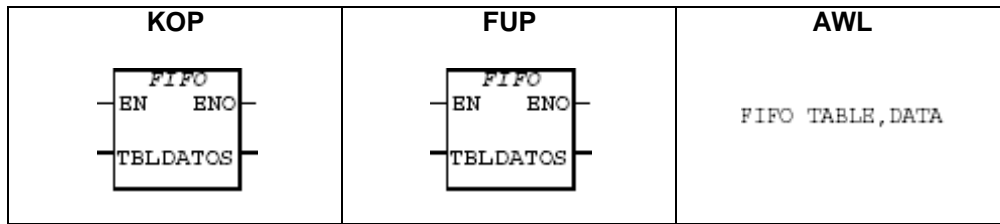


La operación **Buscar valor en tabla** rastrea la tabla (TBL), comenzando con el registro indicado por INDX, y busca el valor (PTN) que corresponda a los criterios de búsqueda definidos por CMD. El parámetro de comando (CMD) indica un valor numérico comprendido entre 1 y 4 que corresponde a la relación =, <>, <, y >, respectivamente.

Si se cumple un criterio, INDX señalará el registro en cuestión. Para buscar el siguiente registro se habrá de incrementar INDX antes de volver a llamar nuevamente a la operación Buscar valor en tabla. Si no se encuentra ningún registro que corresponda al criterio, el valor INDX será igual al número de registros que contiene la tabla.

Una tabla puede tener como máximo 100 registros. Los registros de la tabla (el área donde se desea buscar) están numerados de 0 hasta el valor máximo (99).

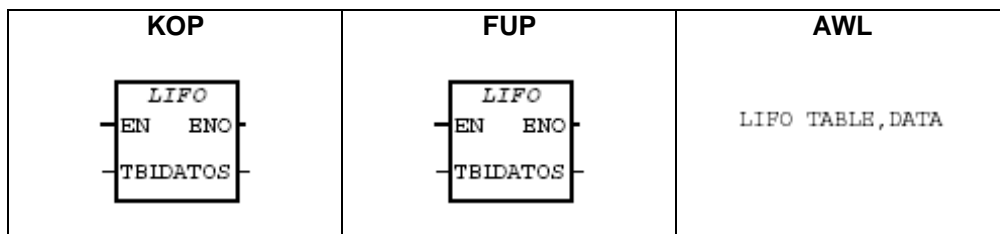
- **Borrar primer registro de la tabla**



La operación **Borrar primer registro de la tabla** borra el primer registro de la tabla (TBL) y transfiere el valor a la dirección DATA. Todos los demás registros se desplazan una posición hacia arriba. El número de registros (EC) de la tabla decrementa cada vez que se ejecuta esta operación.

Estas operaciones afectan a las siguientes marcas especiales: SM1.5 se activa si se intenta borrar un registro de una tabla vacía.

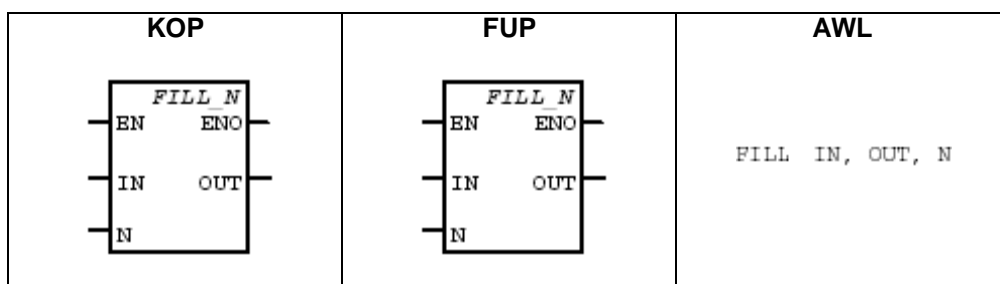
- **Borrar último registro de la tabla.**



La operación **Borrar último registro de la tabla** borra el último registro de la tabla (TBL) y transfiere el valor a la dirección indicada por DATA. El número de registros (EC) de la tabla decrementa cada vez que se ejecuta esta operación.

Estas operaciones afectan a las siguientes marcas especiales: SM1.5 se activa si se intenta borrar un registro de una tabla vacía.

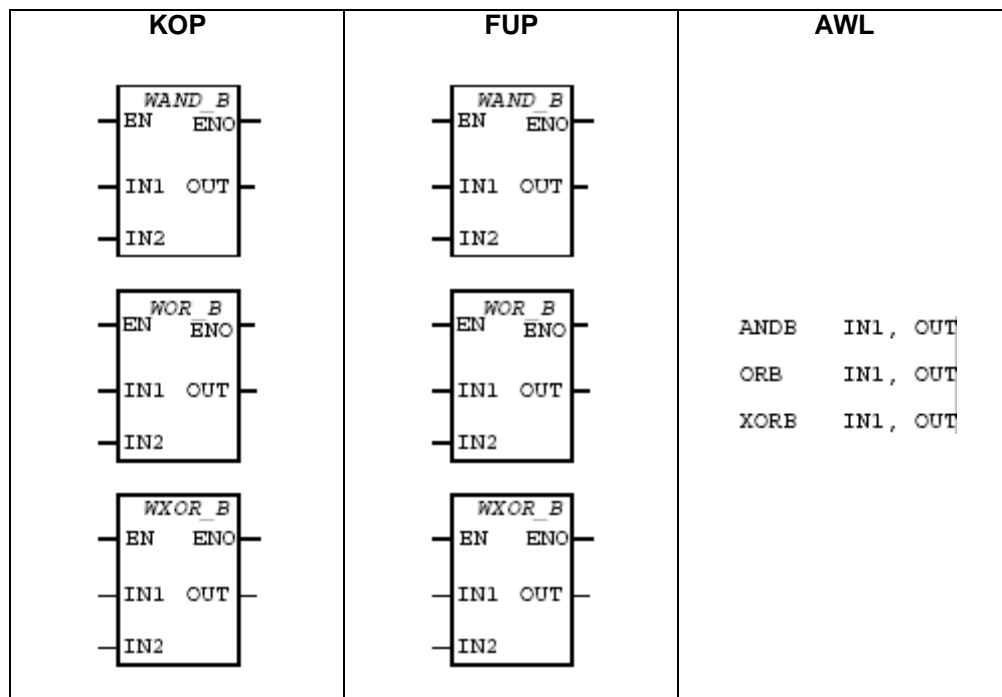
- **Inicializar memoria**



La operación **Inicializar memoria** rellena el número de palabras N de la dirección IN en la dirección OUT.
 N puede estar comprendido entre 1 y 255.

1.3.2.10 Operaciones lógicas (SIMATIC).

- **Combinación Y con bytes, Combinación O con bytes y Combinación O-exclusiva con bytes.**



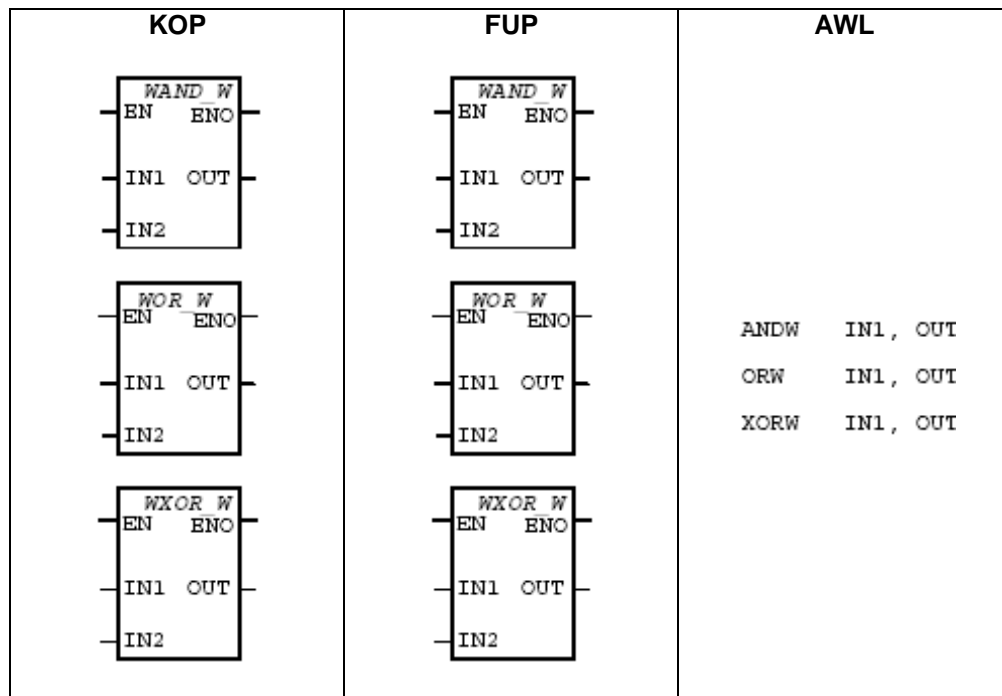
La operación **Combinación Y con bytes** combina los bits correspondientes de los dos bytes de entrada mediante Y, y carga el resultado (OUT) en un byte.

La operación **Combinación O con bytes** combina los bits correspondientes de los dos bytes de entrada mediante O, y carga el resultado (OUT) en un byte.

La operación **Combinación O-exclusiva con bytes** combina los bits correspondientes de los dos bytes de entrada mediante O-exclusiva, y carga el resultado (OUT) en un byte.

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero).

- **Combinación Y con palabras, Combinación O con palabras y Combinación O-exclusiva con palabras.**



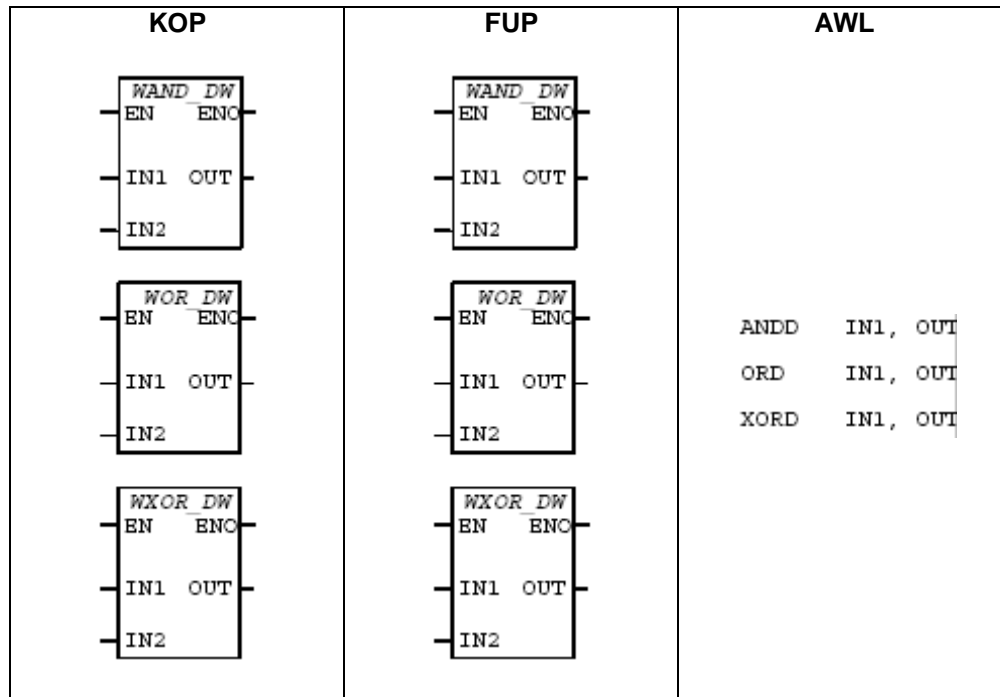
La operación **Combinación Y con palabras** combina los bits correspondientes de las dos palabras de entrada mediante Y, y carga el resultado (OUT) en una palabra.

La operación **Combinación O con palabras** combina los bits correspondientes de las dos palabras de entrada mediante O, y carga el resultado (OUT) en una palabra.

La operación **Combinación O-exclusiva con palabras** combina los bits correspondientes de las dos palabras de entrada mediante O-exclusiva, y carga el resultado (OUT) en una palabra.

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero).

- **Combinación Y con palabras dobles, Combinación O con palabras dobles y Combinación O-exclusiva con palabras dobles**



La operación **Combinación Y con palabras dobles** combina los bits correspondientes de las dos palabras dobles de entrada mediante Y, y carga el resultado (OUT) en una palabra doble.

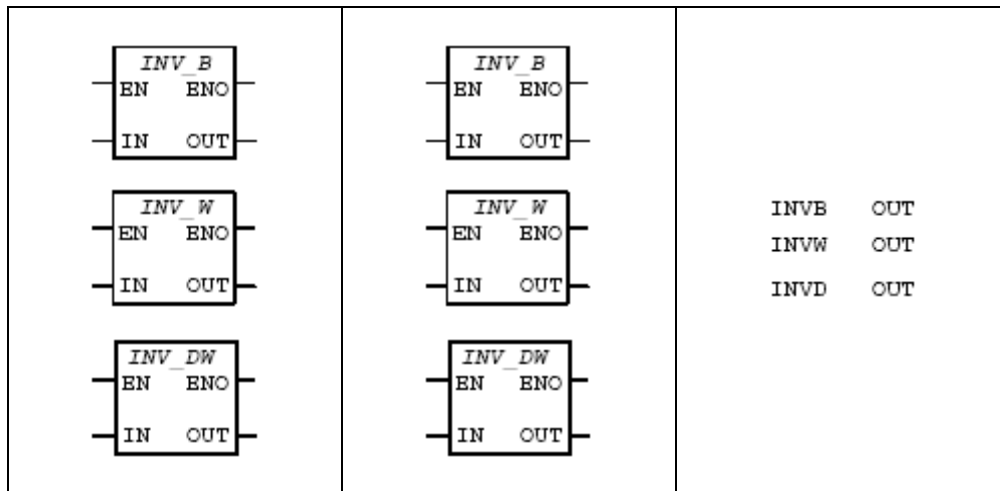
La operación **Combinación O con palabras dobles** combina los bits correspondientes de las dos palabras dobles de entrada mediante O, y carga el resultado (OUT) en una palabra doble.

La operación **Combinación O-exclusiva con palabras dobles** combina los bits correspondientes de las dos palabras dobles de entrada mediante O-exclusiva, y carga el resultado (OUT) en una palabra doble.

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero).

- **Invertir byte, Invertir palabra, Invertir palabra doble.**





La operación **Invertir byte** forma el complemento a 1 del valor del byte de entrada IN y carga el resultado en el valor de byte OUT.

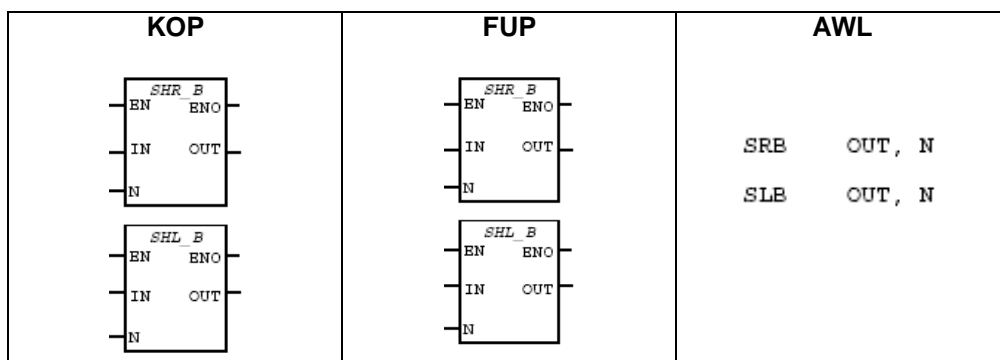
La operación **Invertir palabra** forma el complemento a 1 del valor de la palabra de entrada IN y carga el resultado en el valor de palabra OUT.

La operación **Invertir palabra doble** forma el complemento a 1 del valor de la palabra doble de entrada IN y carga el resultado en el valor de palabra doble OUT.

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero).

1.3.2.11 Operaciones de desplazamiento y rotación (SIMATIC).

- **Desplazar byte a la derecha, desplazar byte a la izquierda.**



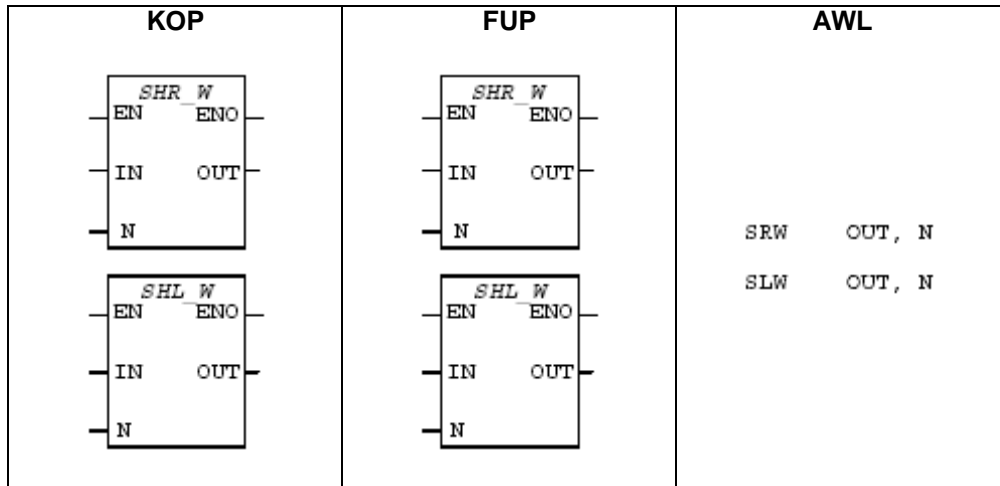
Las operaciones **Desplazar byte a la derecha** y **Desplazar byte a la izquierda** desplazan el valor del byte de entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N), y cargan el resultado en el byte de salida (OUT).

Las operaciones de desplazamiento se rellenan con ceros cada vez que se desplaza un bit. Si el valor de desplazamiento (N) es mayor o igual a 8, el valor se desplazará como máximo 8 veces.

Si el valor de desplazamiento es mayor que 0, la marca de desbordamiento (SM1.1) adoptará el valor del último bit desplazado hacia afuera. La marca cero (SM1.0) se activará si el resultado de la operación de desplazamiento es cero, las operaciones de desplazamiento de bytes no llevan signo.

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento).

- **Desplazar palabra a la derecha, Desplazar palabra a la izquierda.**

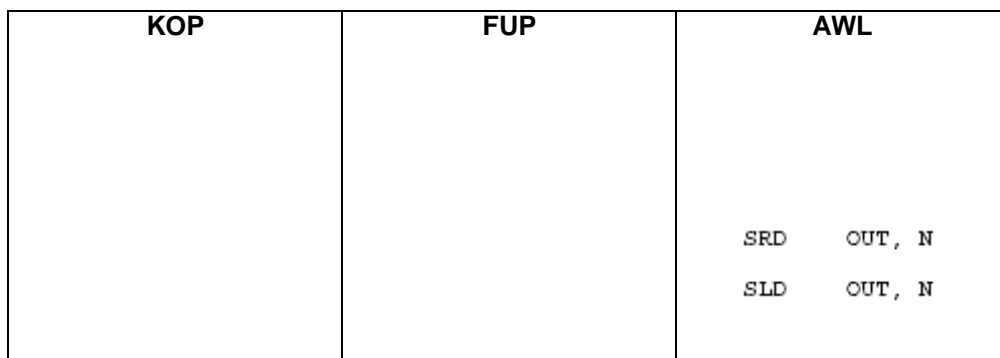


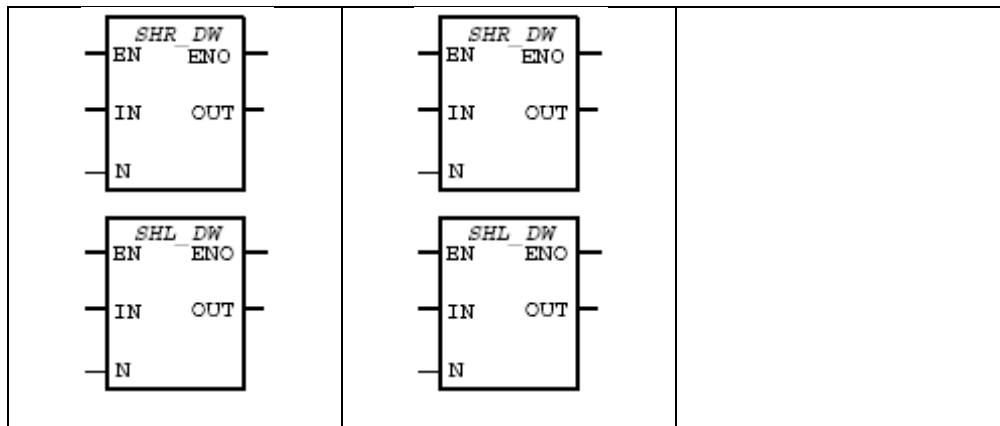
Las operaciones **Desplazar palabra a la derecha** y **Desplazar palabra a la izquierda** desplazan el valor de la palabra de entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N) y cargan el resultado en la palabra de salida (OUT).

Las operaciones de desplazamiento se rellenan con ceros cada vez que se desplaza un bit. Si el valor de desplazamiento (N) es mayor o igual a 16, el valor se desplazará como máximo 16 veces. Si el valor de desplazamiento es mayor que 0, la marca de desbordamiento (SM1.1) adoptará el valor del último bit desplazado hacia afuera. La marca cero (SM1.0) se activará si el resultado de la operación de desplazamiento es cero. El bit de signo se desplaza cuando se utilizan tipos de datos con signo.

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento).

- **Desplazar palabra doble a la derecha, Desplazar palabra doble a la izquierda.**





Las operaciones **Desplazar palabra doble a la derecha** y **Desplazar palabra doble a la izquierda** desplazan el valor de la palabra doble de entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N) y cargan el resultado en la palabra doble de salida (OUT).

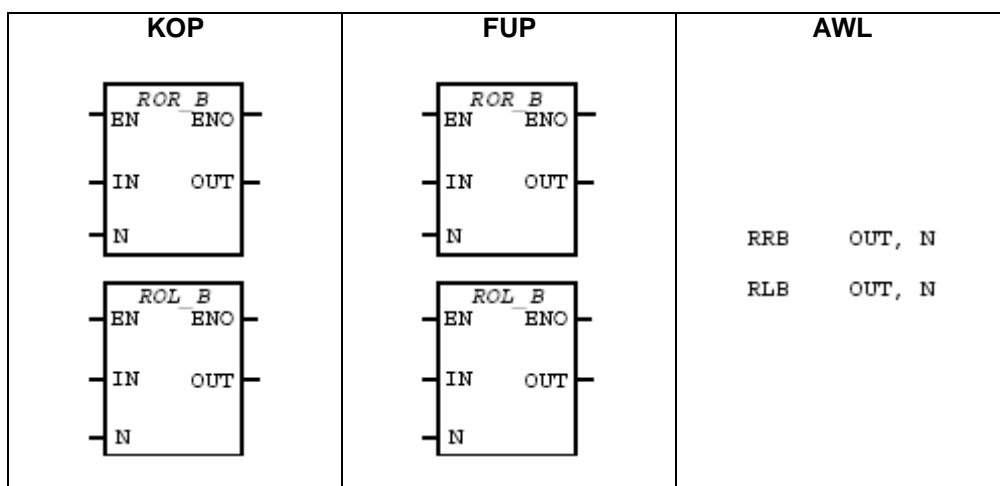
Las operaciones de desplazamiento se rellenan con ceros cada vez que se desplaza un bit. Si el valor de desplazamiento (N) es mayor o igual a 32, el valor se desplazará como máximo 32 veces.

Si el valor de desplazamiento es mayor que 0, la marca de desbordamiento (SM1.1) adoptará el valor del último bit desplazado hacia afuera. La marca cero (SM1.0) se activará si el resultado de la operación de desplazamiento es cero.

El bit de signo se desplaza cuando se utilizan tipos de datos con signo.

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento).

- **Rotar byte a la derecha, Rotar byte a la izquierda.**



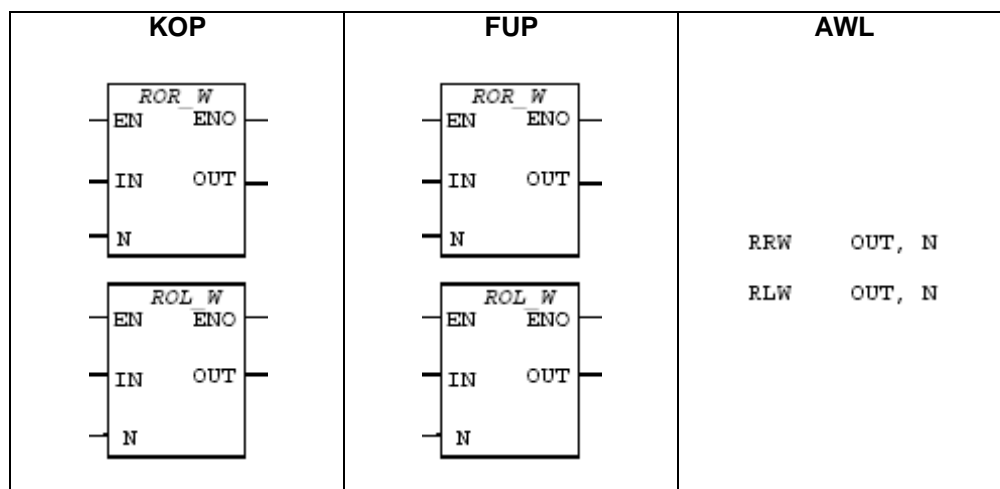
Las operaciones **Rotar byte a la derecha** y **Rotar byte a la izquierda** rotan el valor del byte de entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N) y cargan el resultado en el byte de salida (OUT). La rotación es circular.

Si el valor de desplazamiento (N) es mayor o igual a 8, antes de la operación de rotación se ejecutará una operación módulo 8 en el valor de desplazamiento (N). De ello resulta un valor de rotación de 0 a 7. Si el valor de desplazamiento es igual a 0, no se rotará el valor. Si se ejecuta la rotación, el valor del último bit rotado se copiará en la marca de desbordamiento (SM1.1).

Si el valor de desplazamiento no es un entero múltiplo de 8, el último bit rotado se copiará en la marca de desbordamiento (SM1.1). La marca cero (SM1.0) se activará si el valor a rotar es igual a cero.

Las operaciones de rotación de bytes no llevan signo, estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento).

- **Rotar palabra a la derecha, Rotar palabra a la izquierda.**



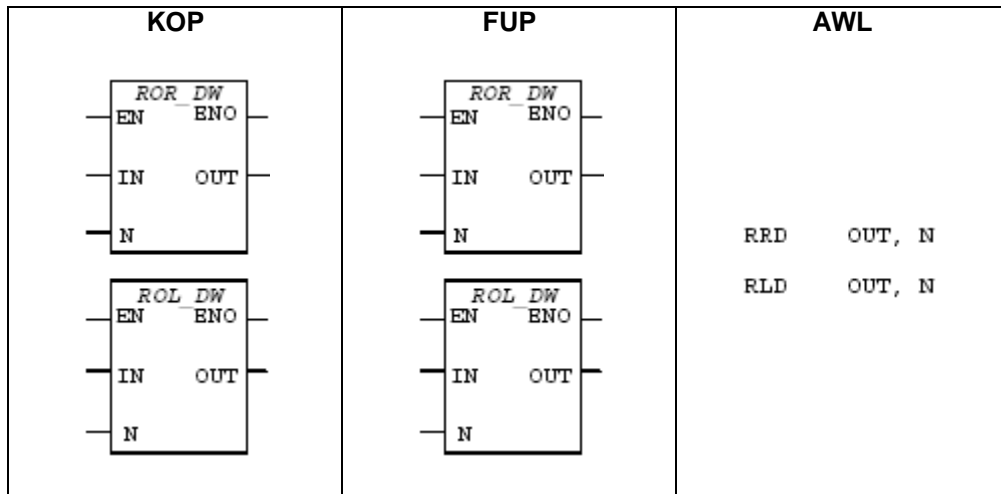
Las operaciones **Rotar palabra a la derecha** y **Rotar palabra a la izquierda** rotan el valor de la palabra de entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N) y cargan el resultado en la palabra de salida (OUT), la rotación es circular, si el valor de desplazamiento (N) es mayor o igual a 16, antes de la operación de rotación se ejecutará una operación módulo 16 en el valor de desplazamiento (N), de ello resulta un valor de rotación de 0 a 15. Si el valor de desplazamiento es igual a 0, no se rotará el valor, si se ejecuta la rotación, el valor del último bit rotado se copiará en la marca de desbordamiento (SM1.1).

Si el valor de desplazamiento no es un entero múltiplo de 16, el último bit rotado se copiará en la marca de desbordamiento (SM1.1). La marca cero (SM1.0) se activará si el valor a rotar es igual a cero.

El bit de signo se desplaza cuando se utilizan tipos de datos con signo.

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento).

- **Rotar palabra doble a la derecha, Rotar palabra doble a la izquierda.**



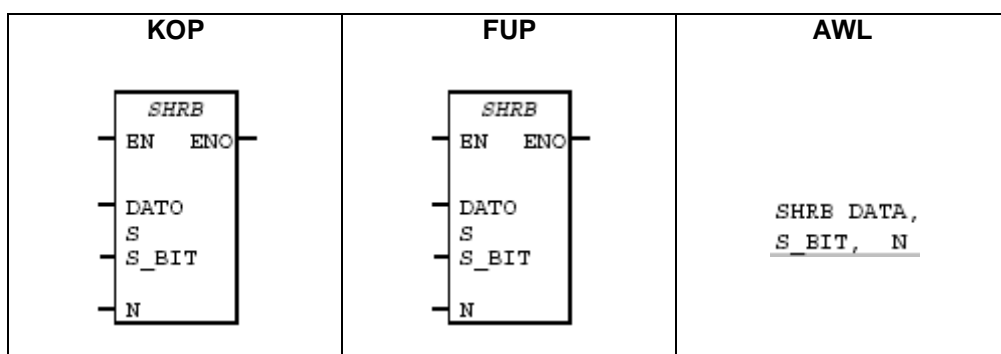
Las operaciones **Rotar palabra doble a la derecha** y **Rotar palabra doble a la izquierda** rotan el valor de la palabra doble de entrada (IN) a la derecha y a la izquierda respectivamente, tantas posiciones como indique el valor de desplazamiento (N) y cargan el resultado en la palabra doble de salida (OUT), la rotación es circular, si el valor de desplazamiento (N) es mayor o igual a 32, antes de la operación de rotación se ejecutará una operación módulo 32 en el valor de desplazamiento (N), de ello resulta un valor de rotación de 0 a 31. Si el valor de desplazamiento es igual a 0, no se rotará el valor, si se ejecuta la rotación, el valor del último bit rotado se copiará en la marca de desbordamiento (SM1.1).

Si el valor de desplazamiento no es un entero múltiplo de 32, el último bit rotado se copiará en la marca de desbordamiento (SM1.1). La marca cero (SM1.0) se activará si el valor a rotar es igual a cero.

El bit de signo se desplaza cuando se utilizan tipos de datos con signo.

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento).

- **Registro de desplazamiento**

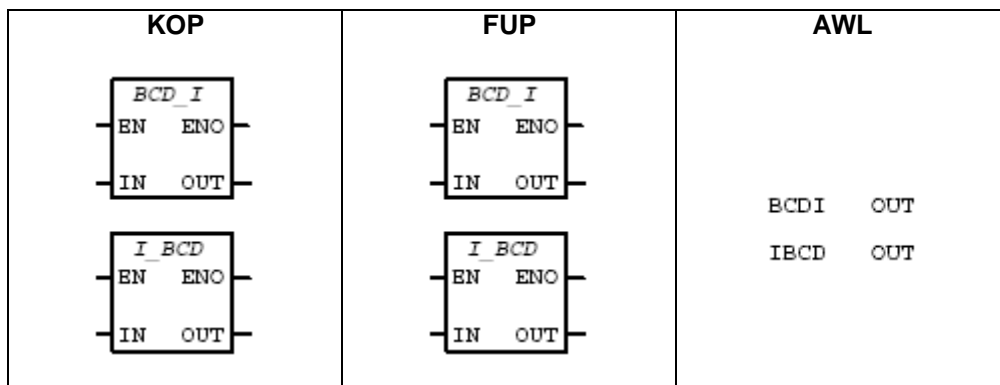


La operación **Registro de desplazamiento** (SHRB) desplaza el valor de DATA al registro de desplazamiento. S_BIT indica el bit menos significativo de este registro. N indica la longitud del registro y el sentido de desplazamiento (valor positivo = N, valor negativo = -N), los bits desplazados por la operación Registro de desplazamiento se depositarán en la marca de desbordamiento (SM1.1).

Estas operaciones afectan a las siguientes marcas especiales: SM1.1 (desbordamiento)

1.3.2.12 Operaciones de conversión (SIMATIC).

- **Convertir de BCD a entero, Convertir de entero a BCD.**

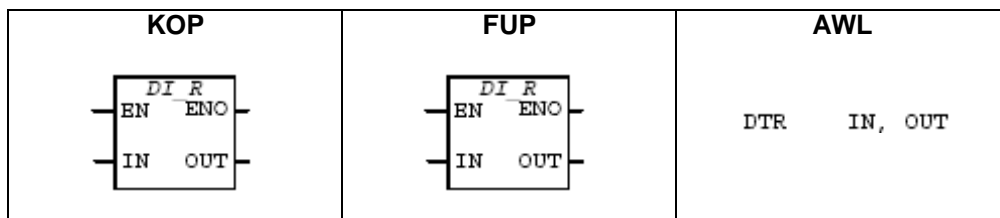


La operación **Convertir de BCD a entero** convierte el valor BCD de entrada (IN) en un valor de entero y carga el resultado en la variable indicada por OUT. El margen válido de IN está comprendido entre 0 y 9999 BCD.

La operación **Convertir de entero a BCD** convierte el valor entero de entrada (IN) en un valor BCD y carga el resultado en la variable indicada por OUT. El margen válido de IN está comprendido entre 0 y 9999 entero.

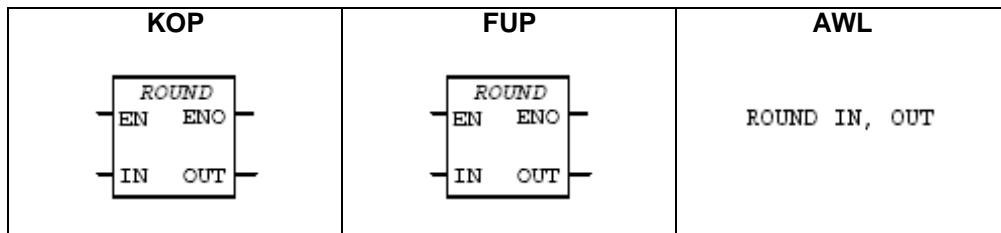
Estas operaciones afectan a las siguientes marcas especiales: SM1.6 (BCD no válido).

- **Convertir de entero doble a real.**



La operación **Convertir de entero doble a real** convierte un entero de 32 bits con signo (IN) en un número real de 32 bits y deposita el resultado en la variable indicada por OUT.

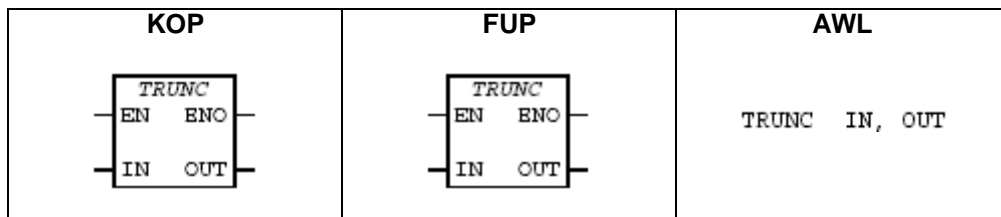
- **Redondear a entero doble.**



La operación **Redondear a entero doble** convierte el valor real (IN) en un valor de entero doble y deposita el resultado en la variable indicada por OUT. Si la fracción es 0,5 o superior, el número se redondeará al próximo entero superior.

Estas operaciones afectan a las siguientes marcas especiales: SM1.1 (desbordamiento).

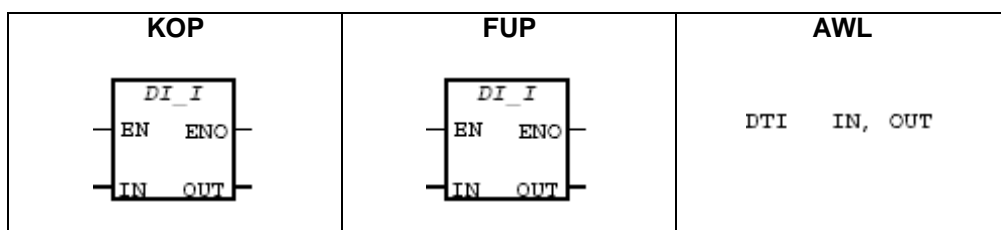
- **Truncar.**



La operación **Truncar** convierte un número real de 32 bits (IN) en un entero de 32 bits con signo y carga el resultado en la variable indicada por OUT. Sólo se convierte la parte entera del número real. La fracción se pierde, si el valor a convertir no es un número real válido o si es demasiado grande para ser representado en la salida, la marca de desbordamiento se activará y la salida no se verá afectada.

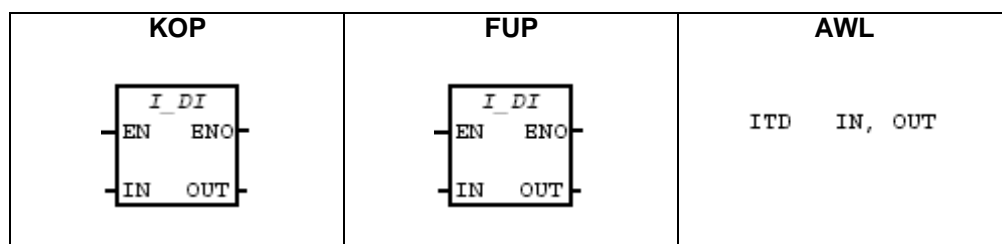
Estas operaciones afectan a las siguientes marcas especiales: SM1.1 (desbordamiento).

- **Convertir de entero doble a entero.**



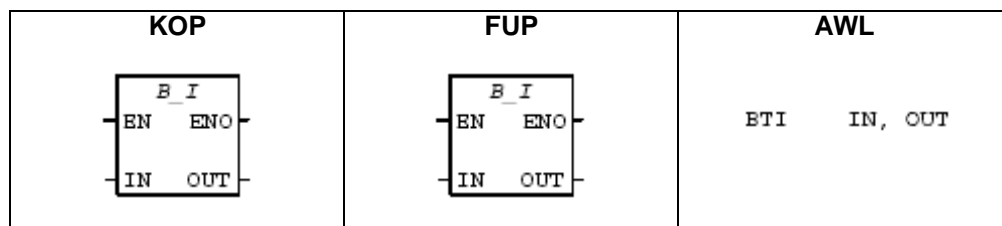
La operación **Convertir de entero doble a entero** convierte el valor de entero doble (IN) en un valor de entero y deposita el resultado en la variable indicada por OUT, si el valor a convertir es demasiado grande para ser representado en la salida, la marca de desbordamiento se activará y la salida no se verá afectada, estas operaciones afectan a las siguientes marcas especiales: SM1.1 (desbordamiento).

- **Convertir de entero a entero doble.**



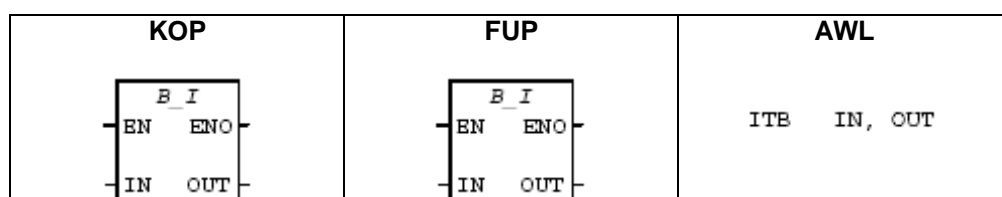
La operación **Convertir de entero a entero doble** convierte el valor de entero (IN) en un valor de entero doble y deposita el resultado en la variable indicada por OUT. El signo se amplía, para convertir un entero a un número real, utilice la operación Convertir de entero a entero doble y luego la operación Convertir de entero doble a real.

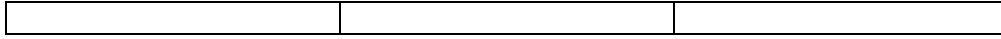
- **Convertir de byte a entero.**



La operación **Convertir de byte a entero** convierte el valor de byte (IN) en un valor de entero y deposita el resultado en la variable indicada por OUT. El byte no tiene signo, por lo tanto, no hay ampliación de signo.

- **Convertir de entero a byte**



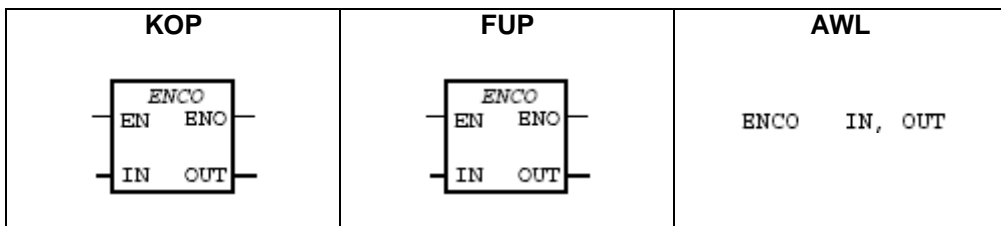


La operación **Convertir de entero a byte** convierte el valor de entero (IN) en un valor de byte y deposita el resultado en la variable indicada por OUT.

Se convierten los valores comprendidos entre 0 y 255. Todos los demás valores producen un desbordamiento y la salida no se ve afectada.

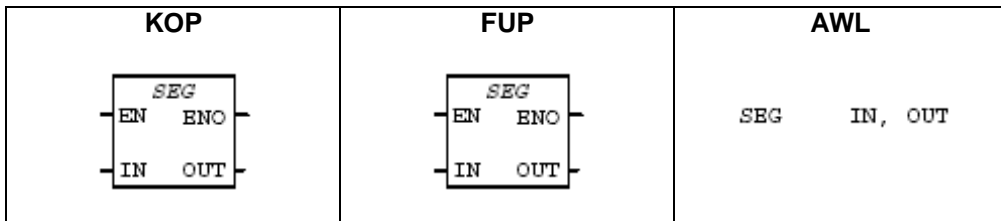
Estas operaciones afectan a las siguientes marcas especiales: SM1.1 (desbordamiento).

- **Codificar.**



La operación **Codificar** escribe el número del bit menos significativo de la palabra de entrada (IN) en el medio byte menos significativo (4 bits) del byte de salida (OUT).

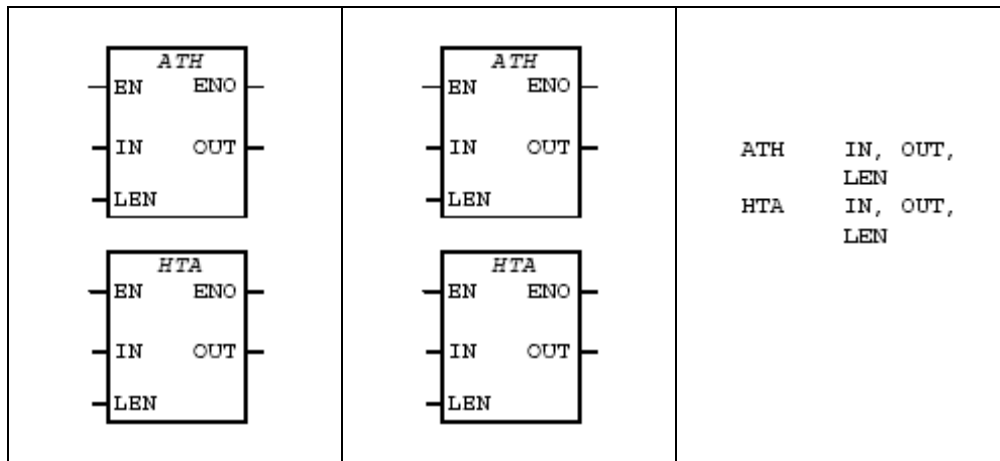
- **Segmento.**



La operación **Segmento** utiliza el carácter indicado por IN para generar una configuración binaria (OUT) que ilumina los segmentos de un indicador de siete segmentos. Los segmentos iluminados representan el carácter depositado en el dígito menos significativo del byte de entrada (IN).

- **Convertir de ASCII a hexadecimal, Convertir de hexadecimal a ASCII.**





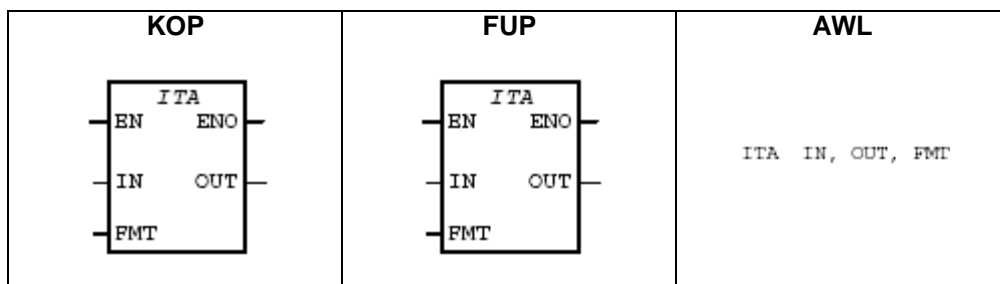
La operación **Convertir de ASCII a hexadecimal** convierte la cadena ASCII de longitud (LEN), a partir del carácter IN, en dígitos hexadecimales, comenzando en OUT. La cadena ASCII puede tener una longitud máxima de 255 caracteres.

La operación **Convertir de hexadecimal a ASCII** convierte los dígitos hexadecimales a partir del byte de entrada (IN) en una cadena ASCII, comenzando en OUT. El número de dígitos hexadecimales a convertir viene indicado por la longitud (LEN), es posible convertir 255 dígitos hexadecimales como máximo.

Los caracteres ASCII admisibles son los valores hexadecimales 30 a 39 y 41 a 46.

Estas operaciones afectan a las siguientes marcas especiales: SM1.7 (ASCII no válido).

- **Convertir de entero a ASCII.**



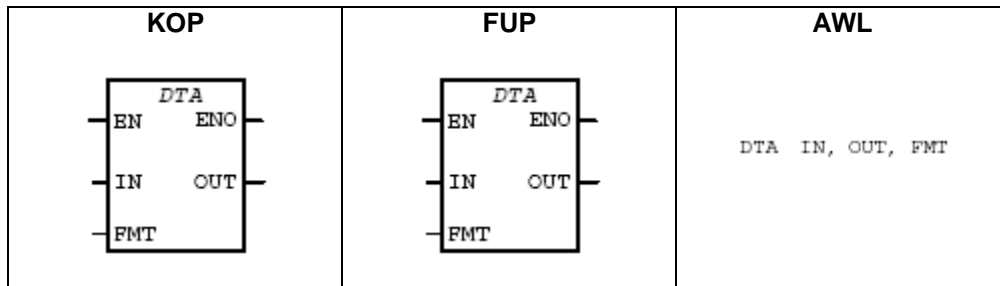
La operación **Convertir de entero a ASCII** convierte un entero (IN) en una cadena ASCII. El formato (FMT) indica la precisión de la conversión a la derecha del decimal, así como si el punto decimal debe aparecer en forma de coma o de punto. La conversión resultante se deposita en 8 bytes consecutivos comenzando en OUT. La cadena ASCII comprende siempre 8 caracteres.

El búfer de salida se formatea conforme a las siguientes reglas:

1. Los valores positivos se escriben sin signo en el búfer de salida.

2. Los valores negativos se escriben precedidos de un signo menos (–) en el búfer de salida.
3. Los ceros a la izquierda del punto decimal (con excepción del dígito adyacente a dicho punto) se suprimen.
4. Los valores se justifican a la derecha en el búfer de salida.

- **Convertir de entero doble a ASCII.**

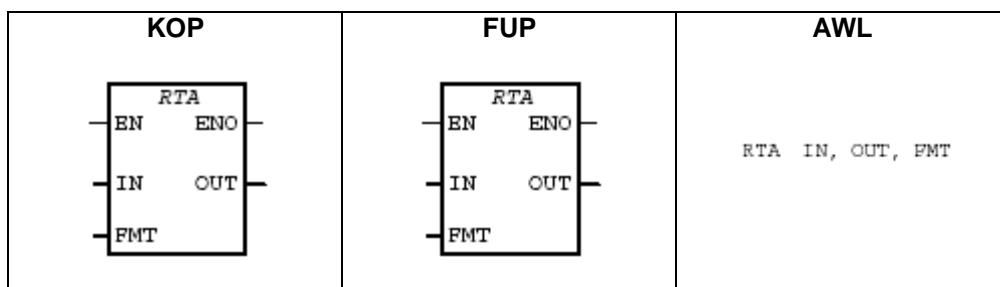


La operación **Convertir de entero doble a ASCII** convierte una palabra doble (IN) en una cadena ASCII. El formato (FMT) indica la precisión de conversión a la derecha del decimal. La conversión resultante se deposita en 12 bytes consecutivos comenzando en OUT.

El búfer de salida se formatea conforme a las siguientes reglas:

1. Los valores positivos se escriben sin signo en el búfer de salida.
2. Los valores negativos se escriben precedidos de un signo menos (–) en el búfer de salida.
3. Los ceros a la izquierda del punto decimal (con excepción del dígito adyacente a dicho punto) se suprimen.
4. Los valores se justifican a la derecha en el búfer de salida.

- **Convertir de real a ASCII**



La operación **Convertir de real a ASCII** convierte el valor en coma flotante (IN) en una cadena ASCII. El formato (FMT) indica la precisión de la conversión a la derecha del decimal, así como si el punto decimal debe aparecer en forma de coma o de punto, y también el tamaño del búfer de salida. La conversión resultante se deposita en un búfer de salida que comienza en OUT.

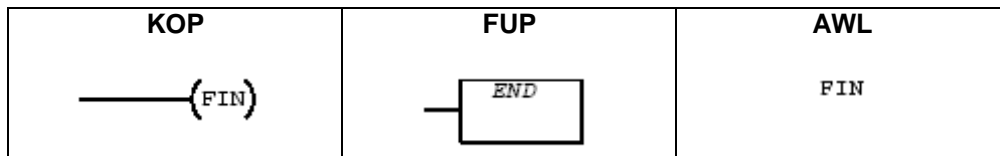
La longitud de la cadena ASCII resultante corresponde al tamaño del búfer de salida, pudiendo indicarse en un margen comprendido entre 3 y 15.

El búfer de salida se formatea conforme a las siguientes reglas:

1. Los valores positivos se escriben sin signo en el búfer de salida.
2. Los valores negativos se escriben precedidos de un signo menos (–) en el búfer de salida.
3. Los ceros a la izquierda del punto decimal (con excepción del dígito adyacente a dicho punto) se suprimen.
4. Los valores a la derecha del punto decimal se redondean para que correspondan al número de dígitos indicado.
5. El búfer de salida deberá ser por lo menos tres bytes más grande que el número de dígitos a la derecha del punto decimal.
6. Los valores se justifican a la derecha en el búfer de salida.

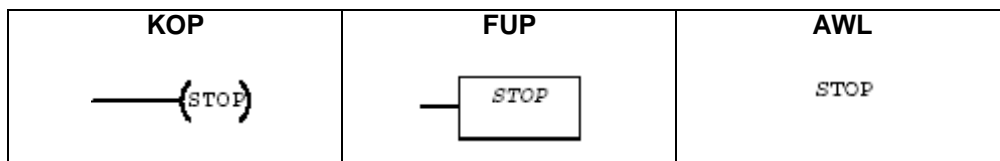
1.3.2.13 Operaciones de control del programa (SIMATIC).

- **END condicional.**



La operación condicional **Finalizar programa principal** finaliza el programa en función de la combinación lógica precedente.

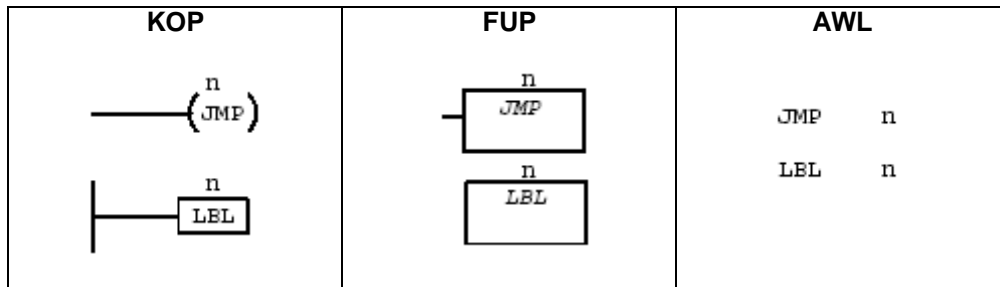
- **Stop.**



La operación **STOP** finaliza inmediatamente la ejecución del programa haciendo que la CPU cambie de RUN a STOP.

Si la operación STOP se ejecuta en una rutina de interrupción, ésta se finalizará inmediatamente ignorando las interrupciones pendientes. Las demás acciones en el ciclo actual se completan, incluyendo la ejecución del programa principal. El cambio de RUN a STOP se produce al final del ciclo actual.

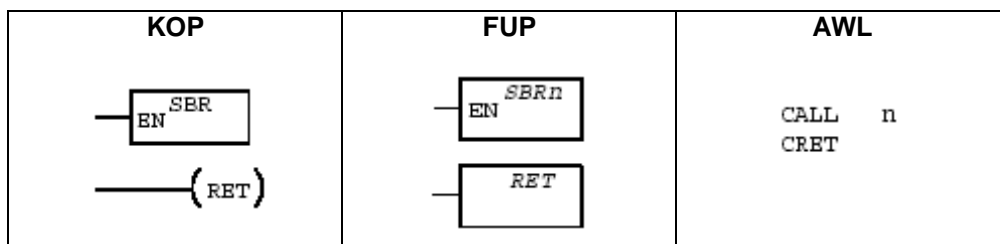
- **Saltar a meta, Definir meta.**



La operación **Saltar a meta** deriva la ejecución del programa a la meta indicada (n). Al saltar, el primer valor de la pila es siempre un “1” lógico; la operación **Definir meta** indica la dirección de la meta de salto (n).

Tanto la operación de salto como la correspondiente meta deben encontrarse en el programa principal, en una subrutina o en una rutina de interrupción. Desde el programa principal no se puede saltar a una meta que se encuentre en una subrutina o en una rutina de interrupción. Tampoco es posible saltar desde una subrutina o una rutina de interrupción a una meta que se encuentre fuera de ella.

- **Llamar subrutina, Retorno de subrutina.**



La operación **Llamar subrutina** transfiere el control a la subrutina (n). Esta operación se puede utilizar con o sin parámetros. Para añadir una subrutina, elija los comandos de menú **Edición > Insertar > Subrutina**.

La operación **Retorno condicional de subrutina** se utiliza para finalizar una subrutina en función de la combinación lógica precedente.

Una vez ejecutada la subrutina, el control vuelve a la operación que sigue a la llamada de la subrutina (CALL).

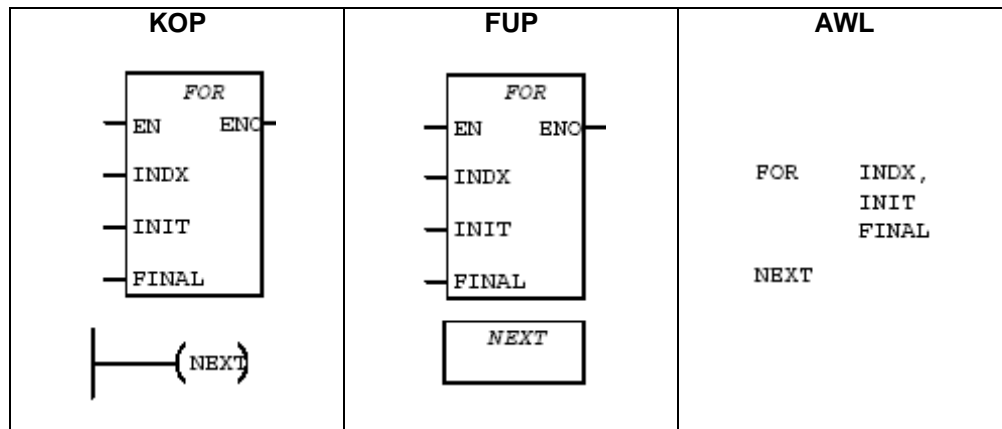
En el programa principal, se pueden anidar (situar una llamada a subrutina en otra) hasta ocho subrutinas. Las subrutinas no se pueden anidar en una rutina de interrupción. Una llamada a subrutina no se puede disponer en ninguna otra subrutina a la que se llame desde una rutina de interrupción. Si bien la recursión (la subrutina se llama a sí misma) está permitida, hay que utilizarla con cautela.

Cuando se llama a una subrutina, se almacena toda la pila lógica, poniéndose a “1” el nivel superior de la pila. Sus demás niveles se ponen a “0” y la ejecución se transfiere a la subrutina que se ha llamado. Cuando

ésta se termina de ejecutar, se restablece la pila con los valores almacenados al llamar a la subrutina y se retorna a la rutina que ha efectuado la llamada.

Los acumuladores son comunes a las subrutinas y a la rutina de llamada. Los acumuladores no se almacenan ni se restablecen si se utilizan con subrutinas.

- **FOR, NEXT.**



La operación **FOR** ejecuta las operaciones que se encuentren entre FOR y NEXT. Se deben indicar el valor del índice o el conteo actual del bucle (INDX), el valor inicial (INIT) y el valor final (FINAL).

La operación **NEXT** marca el final del bucle FOR y pone a “1” el primer valor de la pila.

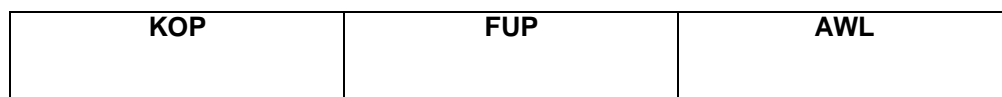
- **Reglas para utilizar el bucle FOR/NEXT:**

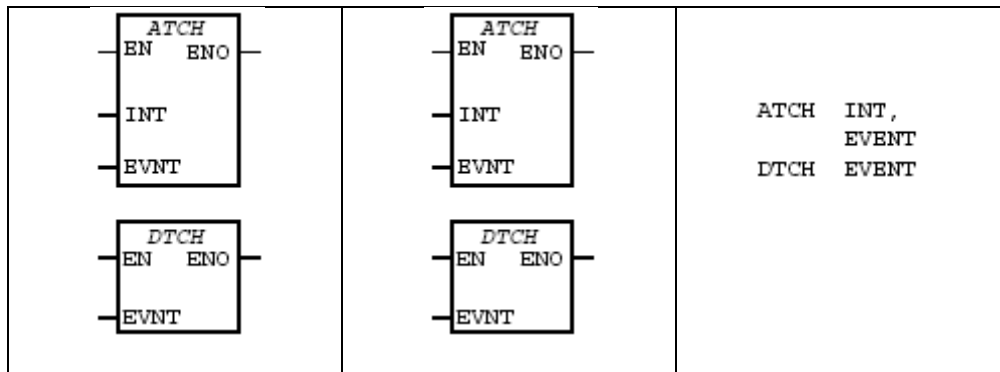
- Al habilitar el bucle FOR/NEXT, éste se ejecuta hasta finalizar las iteraciones, a menos que Ud. cambie el valor final dentro del bucle. Los valores se pueden cambiar mientras se ejecute FOR/NEXT.
- Si se vuelve a habilitar el bucle, éste copia el valor inicial (INIT) en el valor actual de conteo del bucle (IDX). La operación FOR/NEXT se desactiva automáticamente la próxima vez que se habilite.

Las operaciones FOR/NEXT repiten un bucle del programa un número determinado de veces, toda operación FOR exige una operación NEXT. Los bucles FOR/NEXT pueden anidarse (insertar un bucle FOR/NEXT dentro de otro) hasta una profundidad de ocho niveles.

1.3.2.14 Operaciones de interrupción y comunicación (SIMATIC).

- **Asociar interrupción, Desasociar interrupción.**





La operación **Asociar interrupción** asocia el número de una rutina de interrupción (INT) a un evento de interrupción (EVNT), habilitando así éste último.

La operación **Desasociar interrupción** desasocia un evento de interrupción (EVNT) de todas las rutinas de interrupción, deshabilitando así el evento.

- **Descripción de las operaciones Asociar interrupción y Desasociar interrupción**

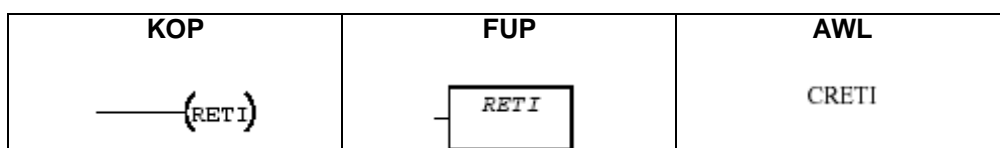
Antes de poder llamar a una rutina de interrupción es preciso establecer un enlace entre el evento de interrupción y la parte del programa que se desee ejecutar cuando se presente el evento. La operación Asociar interrupción (ATCH) sirve para asignar el evento de interrupción (indicado por el número de evento) a una parte del programa (indicada por el número de la rutina de interrupción). También es posible asociar varios eventos de interrupción a una única rutina de interrupción. Por el contrario, no se puede asociar un sólo evento a distintas rutinas. Cuando se produce un evento estando habilitadas las interrupciones, se ejecuta únicamente la última rutina de interrupción asociada a dicho evento.

Cuando se asocia un evento a una rutina de interrupción, se habilita automáticamente el evento. Si se inhiben todos los eventos de interrupción, entonces cada vez que se presente la interrupción, se pondrá en cola de espera hasta que las interrupciones se habiliten de nuevo, utilizando para ello la operación Habilitar todos los eventos de interrupción.

También es posible inhibir ciertos eventos de interrupción, eliminando la asociación entre el evento y la correspondiente rutina mediante la operación DTCH (Desasociar interrupción).

Esta operación retorna la interrupción a un estado inactivo o ignorado.

- **Retorno condicional desde rutina de interrupción.**



--	--	--

La operación **Retorno condicional desde rutina de interrupción** finaliza una rutina en función de la combinación lógica precedente. Para añadir una interrupción, elija los comandos de menú **Edición > Insertar > Interrupción**.

En la pantalla de STEP 7-Micro/WIN 32, los retornos desde rutinas de interrupción se visualizan en fichas por separado.

- **Rutinas de interrupción**

La rutina de interrupción se ejecuta como respuesta a un evento interno o externo asociado, tras haberse ejecutado la última operación de la rutina de interrupción, el control retorna al programa principal. Para salir de la rutina se puede ejecutar una operación Retorno condicional desde rutina de interrupción (CRETI).

- **Reglas para utilizar interrupciones.**

El procesamiento de interrupciones permite reaccionar rápidamente ante determinados eventos internos o externos. Las rutinas de interrupción se deben estructurar de forma que una vez ejecutadas determinadas tareas - devuelvan el control al programa principal. A tal efecto es conveniente crear rutinas de interrupción cortas con indicaciones precisas, de manera que se puedan ejecutar rápidamente sin interrumpir otros procesos durante períodos demasiado largos. Si no se observan estas medidas, es posible que se produzcan estados imprevistos que pueden afectar a la instalación controlada por el programa principal. Al utilizar interrupciones, conviene atenerse al lema de “cuanto más breve, mejor”.

- **Restricciones**

No utilice las operaciones DISI, ENI, HDEF, LSCR y END en las rutinas de interrupción.

Soporte del sistema durante las interrupciones.

Como las interrupciones pueden afectar a la lógica de contactos, bobinas y acumuladores, el sistema almacena la pila lógica, los acumuladores y las marcas especiales (SM) que indican el estado de los acumuladores y las operaciones, volviéndolos a cargar posteriormente, de este modo se previenen perturbaciones en el programa principal debidas a derivaciones a rutinas de interrupción o desde ellas.

Llamar a subrutinas desde rutinas de interrupción.

Desde una rutina de interrupción se puede llamar a un nivel de anidamiento de subrutinas, los acumuladores y la pila lógica son compartidos por la rutina de interrupción y por la subrutina invocada.

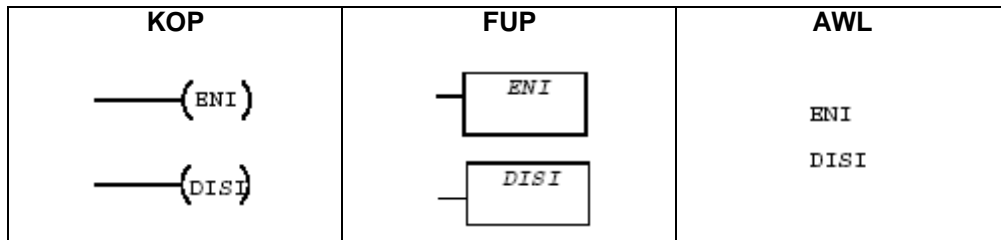
Datos compartidos por el programa principal y las rutinas de interrupción.

El programa principal y una o varias rutinas de interrupción pueden compartir datos. Por ejemplo, una parte del programa principal puede suministrar datos a una rutina de interrupción o viceversa. En el caso de que el programa esté compartiendo datos, habrá que considerar también el hecho de que las rutinas de interrupción se ejecutan de forma asíncrona al programa principal. Por lo tanto, se pueden presentar en cualquier momento durante la ejecución de éste último. Los problemas de coherencia de los datos compartidos pueden ser ocasionados por las acciones de las rutinas de interrupción, al interrumpir éstas la ejecución de las operaciones del programa principal.

Hay diversas técnicas de programación que se pueden utilizar para garantizar que el programa principal y las rutinas de interrupción compartan los datos correctamente. Estas técnicas restringen la forma de acceder a las direcciones compartidas en la memoria o evitan que se interrumpan las secuencias de operaciones que utilicen direcciones compartidas.

- En un programa AWL que comparta sólo una variable: Si los datos compartidos son una sola variable en formato de byte, palabra o palabra doble, y el programa se ha escrito en AWL, los resultados intermedios de operaciones con datos compartidos sólo se podrán almacenar en direcciones o en acumuladores que no se compartan.
- En un programa KOP que comparta sólo una variable: Si los datos compartidos son una sola variable en formato de byte, palabra o palabra doble, y el programa se ha escrito en KOP, es preciso acceder a las direcciones compartidas utilizando las operaciones de transferencia (MOVB, MOVW, MOVDW, MOVR). En tanto que numerosas operaciones KOP comprenden secuencias de instrucciones AWL que se pueden interrumpir, estas operaciones de transferencia equivalen a una sola operación AWL, cuya ejecución no se ve afectada por los eventos de interrupción.
- En un programa AWL o KOP que comparta varias variables: Si los datos compartidos son varios bytes, palabras o palabras dobles contiguas, la ejecución de la rutina de interrupción se puede controlar con las operaciones Habilitar todos los eventos de interrupción (ENI) e Inhibir todos los eventos de interrupción (DISI). Las interrupciones se deben inhibir en aquel punto del programa principal donde figuran las operaciones que acceden a las direcciones compartidas. Una vez ejecutadas todas las operaciones que utilicen las direcciones compartidas, se deberán habilitar de nuevo las interrupciones. Mientras esté inhibida la interrupción no se podrá ejecutar la rutina correspondiente. Por lo tanto, no será posible acceder entonces a las direcciones compartidas. Sin embargo, esta técnica de programación puede causar que se ignoren los eventos de interrupción.

- **Habilitar todos los eventos de interrupción, Inhibir todos los eventos de interrupción.**



La operación **Habilitar todos los eventos de interrupción** habilita la ejecución de todos los eventos asociados.

La operación **Inhibir todos los eventos de interrupción** inhibe la ejecución de todos los eventos asociados.

Cuando el CPU pasa a modo RUN, las interrupciones se inhiben. Estando en modo RUN, se pueden habilitar todos los eventos de interrupción con la operación global ENI. La operación DISI permite poner las interrupciones en cola de espera, pero no llamar a ninguna rutina de interrupción.

CAPITULO II

ESTUDIO DE LAS MÁQUINAS Y DETERMINACIÓN DE PARÁMETROS

2.1 Estudio de la Máquina Carda C4 Rieter 84.

2.1.1 Descripción de la Carda.

El cardado es un proceso en el cual se endereza parcialmente las fibras y forma con ellas una trama delgada que se unen en una cuerda suave conocida como cinta de cardado. La máquina para cardar se compone de cilindros cubiertos con una guarnición gruesa y pesada, de elementos llamados chapones y otro elementos como peinador, otros cilindros de salida y de una apiladora donde se recoge la cinta en botes que es el resultado del proceso de esta máquina, en la figura 2.1 se observa una carda.

El material previamente procesado ingresa a la tolva de alimentación, la cual guía y regula la cantidad de material que va ingresar al cilindro alimentador que es controlado electrónicamente para obtener una alimentación uniforme dependiendo tanto de los dispositivos de la entrada que es la tolva en la cual se tiene un sensor de desplazamiento, en la salida los cilindros calandrades que es por donde se obtiene la cinta de carda se mide la variación de la cinta, en base a un set point, un motor controla el sistema de regulación mediante la variación de velocidad para tener una cinta de material uniforme a la salida.



Figura 2.1 Carda C4 Rieter 84.

Los cilindros de alimentación son los que lleva el material al likerín el cual es el encargado de abrir y limpiar el material, mediante estiramientos se logra obtener una fina capa de material en forma de velo el cual es llevado por el gran tambor, el mismo es forrado por una guarnición en forma de dientes de sierra, los chapones son los encargados de limpiar, sacar fibra corta y da uniformidad al velo que sigue la trayectoria hacia el peinador, el mismo dispone de unas guarniciones que arrastra el material y mediante rodillos se lleva hacia los cilindros calandrades procesos en el cual el velo se transforma en cinta que es recogida en botes, función que realiza la apiladora.

En el mismo eje de estos cilindros calandrades se encuentra acoplado un disco con agujeros que da la señal al sensor de proximidad, este sensor es que da la información de la cantidad de material que se está entregando, en la figura 2.2 se muestra el lugar de donde se capta la señal que es enviada para el procesamiento del contador.



Figura 2.2 Ubicación de sensor de proximidad en la Carda C4

El sensor utilizado es uno de proximidad inductivo tipo NPN (figura 2.3), este sensor tiene tres terminales, necesita de una fuente de alimentación de 24Vdc, cuando se le aproxima un metal entrega una señal en bajo “0”, sin la presencia de metal su estado es alto, la conexión del sensor se lo muestra en la figura 2.4.



Figura 2.3 Sensor de proximidad utilizado por la Carda C4

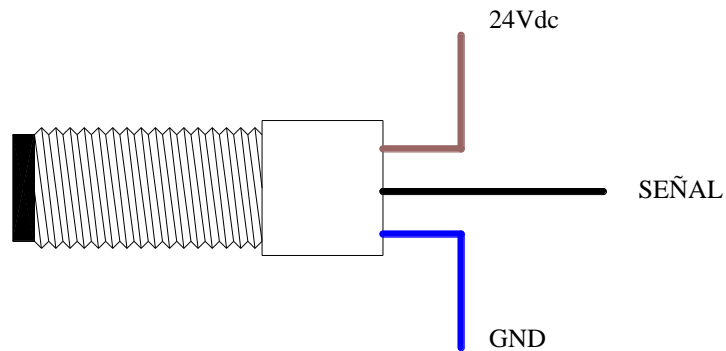


Figura 2.4 Identificación de cables de sensor de proximidad.

2.2 Estudio de la máquina peinadora E75, E76.

2.2.1 Descripción de la peinadora.

El proceso fundamental del peinado es colocar las fibras en posición paralela y eliminar cualquier fibra corta del resto de material, de manera que las fibras peinadas tendrán una longitud más uniforme. De la máquina de peinado las fibras salen como cinta peinada, se debe tomar en cuenta que las fibras de mayor longitud son costosas, además de que se desperdicia una cierta cantidad de fibras que son eliminadas dependiendo de la calidad que se quiera obtener en el producto final que es hilo de algodón, en la figura 2.5 se muestra una peinadora E75.

En los cilindros alimentadores se colocan 8 napas, las cuales ingresan por las mesas, las cintas las cuales son movidas por los trinquetes, la pinza de la mesa sujeta todas las cintas, haciendo que se produzcan un movimiento de pinzado y peinado entre el peine fijo y el peine circular, el velo es agarrado por dos cilindros ranurados de acero y dos cilindros de caucho en la parte superior llamados cilindros arrancadores, aquí se produce un estiramiento mínimo y produce la formación del velo que luego pasa por una tolva en la que el velo elimina suciedad, fibra corta, paralelización de la fibra, los neps.

Continúa a través de un cilindro ranurado conocido como cilindro de alimentación, ingresa el velo por un embudo transformándose en cinta por medio de los cilindros calandrades de la peinadora, se obtienen 8 cintas las cuales son guiadas por la mesa que pasan a un tren de estiraje conformado por un elemento inferior y un elemento superior.

El tren de estiraje inferior consta de 5 cilindros ranurados de acero, el tren de estiraje superior tiene tres cilindros de caucho entre cruzados produciendo un estiramiento mediante presión de aire, en la siguiente fase el material ingresa por un embudo formando una sola cinta la cual es transportada por una banda a un embudo y cilindros calandrades, la cinta sigue su proceso por un plato giratorio llamado apiladora la cual se envuelve en un bote, la base del cambiadotes tiene un movimiento de vaivén para que la cinta se envuelva de una manera correcta.



Figura 2.5 Peinadora E75.

Al igual que en las cardas el sensor está ubicado junto a un piñón que está en el mismo eje de los cilindros calandrades que están ubicados junto a la apiladora de material, a diferencia de las cardas la peinadora no tiene un disco agujerado, en su lugar tiene un piñón con diez dientes, en la figura 2,6 se muestra la ubicación del sensor en la peinadora.



Figura 2.6 Ubicación de sensor en la Peinadora.

Al igual que en la carda el tipo de sensor es el mismo el que se utiliza en la peinadora.

2.3 Estudio de la Máquina Hila G51.

2.3.1 Descripción de la Hila.

La hilatura proporciona la torsión que hace del hilado simple un hilado de fibra discontinua. La hilatura en anillos estira, tuerce y enrolla en una sola operación continua. El cursor transporta el hilo mientras se desliza alrededor del anillo, impartiendo así el torcido. Como el hilado es un proceso textil lento en procesos de productividad por unidad producida, se ha limitado en la velocidad del cursor, tamaño de paquete, adaptabilidad a la automatización, se ha tenido mucho interés en la hilatura de cabo abierto que se asemeja en muchos aspectos a la hilatura primitiva en ruelas sin volante, en la figura 2.7 se observa una Hila G51.



Figura 2.7 Imagen de una Hila.

Los pabilos que salen de las mecheras son llevados y colocados manualmente en la parte superior de la hila en unos colgantes, las mechas deben ser guiadas de una manera correcta para un fácil desenrollado de material, estas pasan por un “guía pabilo” y se dirigen hacia el tren de estiraje; como ya se explicó anteriormente la función fundamental es estirar el material mediante unos brazos de presión.

El tren de estiraje está conformado de la siguiente manera: El tren de estiraje inferior se encuentra formado por tres ejes metálicos ranurados los mismos que se encuentran ubicados a lo largo de toda la máquina, el tren de estiraje superior se encuentran formados por los brazos de presión que trabajan con una presión máxima de 2.2 bares siempre y cuando la hila esté funcionando con cilindros nuevos, en los brazos contienen 3 cilindros de caucho además en este tiene un clip que se diferencia la medida por colores y dependiendo del título de hilo que se quiera realizar, los mismos que se utilizan para dar altura a los cilindros intermedios superiores de los intermedios inferiores.

Al salir del tren de estiraje pasa por el “guía hilos” luego es colocado el hilo en el cursor que se encuentra en los anillos antibalón y por último ser bobinado. La función del cursor es servir de guía para el enrollamiento del hilo en la bobina, el mismo que es seleccionado de diferente número para cada título, además es útil para evitar demasiadas roturas en el hilo.

El tren de estiraje es un sistema de piñones, en uno de los piñones se encuentra ubicado el sensor de la Hila, en la figura 2.8 se puede apreciar la ubicación del sensor.



Figura 2.8 Ubicación de sensor de proximidad en la Hila.

Al igual que en la Carda y en la Peinadora, en la hilo se usa el mismo tipo de sensor.

2.4 Determinación de Parámetros de las Máquinas.

Todos los parámetros que se pueden considerar están basados a la señal que entrega el sensor de proximidad, esta señal consiste en pulsos de onda cuadrada de frecuencia dependiente de la velocidad de cada máquina.

En base a los pulsos que el sensor envía, la primera acción que el contador realiza es los pulsos transformarla en frecuencia, esto se lo realiza con un contador que sea capaz de contabilizar los pulsos en un periodo de 1Seg. este número de pulsos contados en un segundo equivale a un valor de frecuencia, el cálculo de revoluciones (RPM) se lo realiza mediante la siguiente fórmula.

$$RPM = \frac{Frecuencia \times 60}{ND}$$

Donde: Frecuencia= Pulsos/Segundo
 60= 1 minuto tiene 60 Seg.
 ND= Número de dientes ó agujeros.

Para la determinación de los parámetros debemos tomar en cuenta solo el número de agujeros o el número de dientes que tiene un piñón:

- En caso de la carda se tienen 14 pulsos por revolución.
- Para la peinadora tenemos un piñón de 10 dientes, es decir 10 pulsos por revolución.
- Para la hilo tenemos un piñón de 36 dientes que equivalen a una revolución expresada por 36 pulsos.

CAPITULO III

DESARROLLO DEL SISTEMA

3.1 DISEÑO DEL SOFTWARE A IMPLEMENTARSE.

3.1.1 Diseño de Tacómetro, visualización de parámetros de entrega.

En el desarrollo del software se realizaron varios programas de aplicación para llegar al objetivo trazado, empezando por un tren de pulsos, el cual da una señal de habilitación y una señal de reset a un contador encargado de recibir los pulsos provenientes del sensor de proximidad de las diferentes máquinas, el tren de pulsos tiene un periodo de 1 segundo, de esta manera los valores de pulsos/segundo equivalen a un valor de frecuencia, por ejemplo 5 pulsos/seg. es equivalente a 5Hz, en la figura 3.1 se observa el programa para obtener un tren de pulsos.

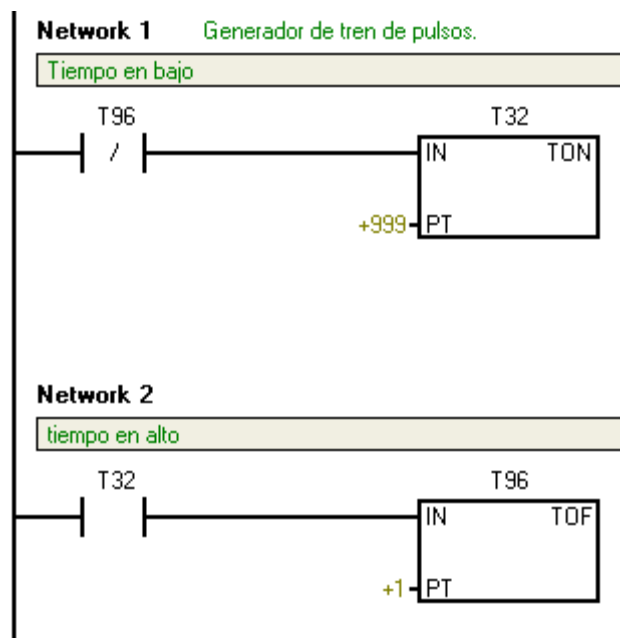


Figura 3.1 Generador de tren de pulsos.

Con este generador de pulsos tenemos una base de tiempo de 1Seg, mediante un contacto de T32 tenemos una señal impulso para llamar a la subrutina “TACOMETRO”, esta subrutina se origina automáticamente al programar un contador rápido, en este caso se utilizó el contador rápido HC3 en modo 1, la entrada de reloj

para este contador es la entrada I0.1 del PLC y el reset se lo realiza cada vez que se llama la subrutina “TACOMETRO”, de esta manera cada segundo tenemos una nueva lectura de pulsos en 1 segundo que es equivalente a un valor de frecuencia en Hertz, para poder visualizar y realizar las operaciones respectivas se toman los datos 2mSeg. antes de que se termine el ciclo de 1 Seg, esto se lo realiza mediante un contacto de comparación que en ese instante expresa el valor del contador HC3, al mismo tiempo permite la visualización en el TD-200, se debe tomar en cuenta en donde está conectado el sensor de proximidad, si está cerca de un piñón ó cerca de una rueda de agujeros, el valor de frecuencia al ser dividido por el número de dientes del piñón o el número de agujeros de la rueda obtendremos un valor expresado como rev/seg, al multiplicar este valor por 60 el valor se expresa como rev/min o RPM, como se indica la siguiente fórmula.

$$RPM = \frac{Frecuencia \times 60}{ND}$$

Donde: Frecuencia= Pulsos/Segundo

60= 1 minuto tiene 60 Seg.

ND= Número de dientes ó agujeros.

En la figura 3.2 se muestra un diagrama de tiempo en donde se indica la operación de los temporizadores T32(Off-delay) y T96(On-delay), estos temporizadores son los que proporcionan el tiempo en alto y el tiempo en bajo para obtener el generador de pulsos que es parte principal para la correcta operación del tacómetro, en base al tacómetro se puede calcular la “velocidad de entrega” que es un factor indispensable en las lecturas que entrega el contador EZA1, con el valor entregado en RPM se multiplica por un factor y obtenemos el valor de *velocidad de entrega* que también se visualiza en el TD-200.

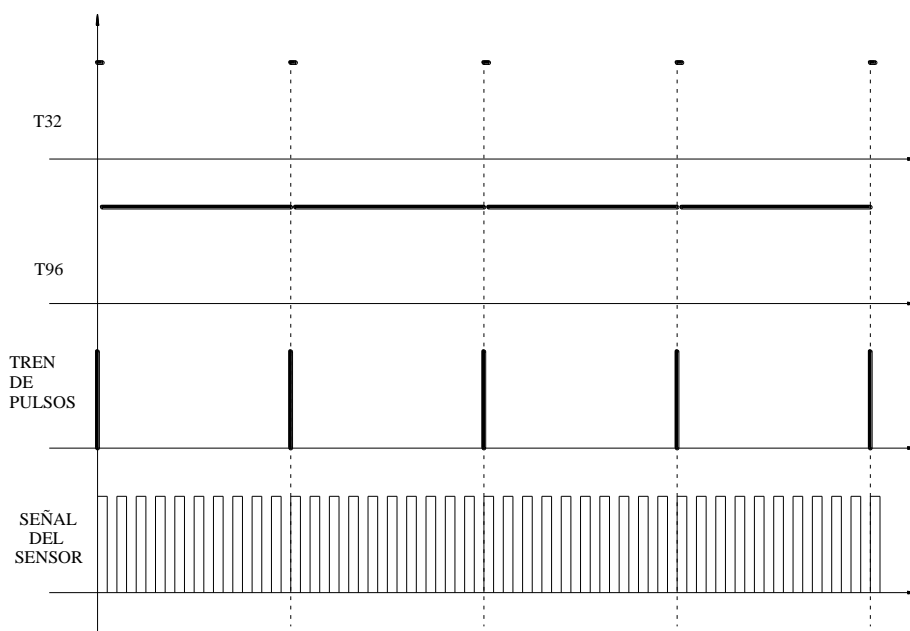


Figura 3.2 Diagrama de tiempo de tren de pulsos.

En la figura 3.3 se indica el programa para obtener valores de T32 por medio de una multiplicación de números en coma fija, la multiplicación se habilita mediante la marca especial SM0.0, este valor se expresa por medio de la variable VD550, este valor es usado en un contacto de comparación que habilita la multiplicación entre el valor del contador rápido (HC3) y la constante 1, esto se realiza para obtener un valor en palabra doble (VD76) el mismo que visualiza la frecuencia en el TD-200, en la figura 3.3 también se muestra el contacto T32 que habilita cada segundo la interrupción “TACOMETRO”.

Además se tiene la salida Q0.5 del PLC, esta salida funciona cuando el PLC está en normal funcionamiento o en la posición RUN, si el PLC se cambia a la posición STOP la máquina no puede operar ya que el contador no está listo para trabajar.

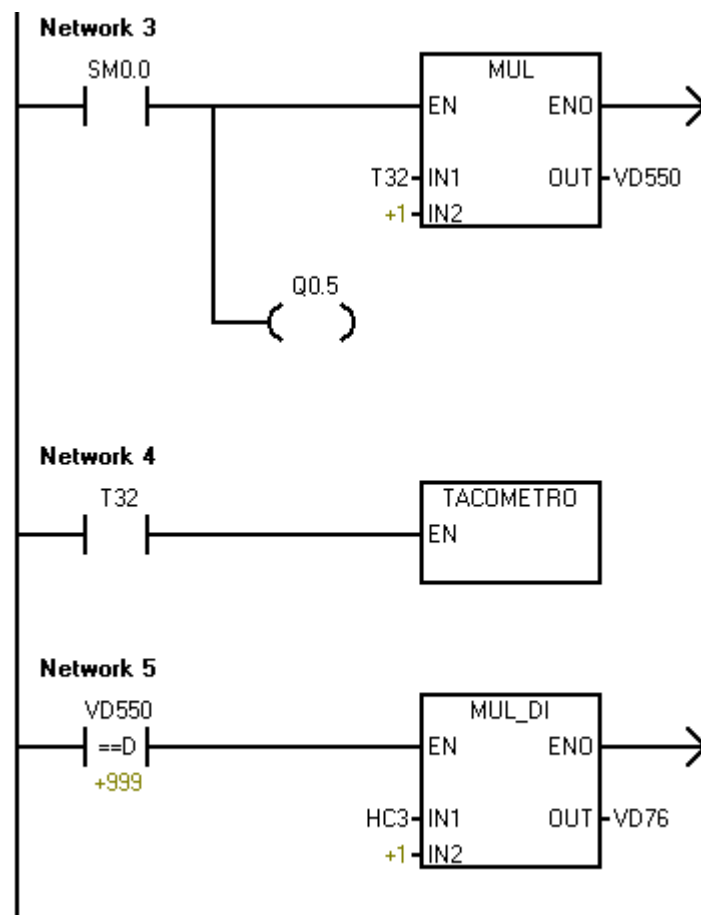


Figura 3.3 Programa para obtención de frecuencia.

En el siguiente paso del programa se utiliza operaciones matemáticas para realizar cálculos y obtener los valores de RPM y Velocidad de entrega, estos valores se han combinado en operaciones en coma fija y en coma flotante, estos últimos se utilizan cuando se trabaja con puntos decimales, en el PLC se los conoce

como números reales¹, en el caso de la velocidad de entrega utiliza un dígito después de la coma, mediante la programación del TD-200 se escoge la cantidad de dígitos que se deben visualizar después de la coma, en la figura 3.4 se indica el juego de operaciones matemáticas para la visualización de datos, en esta parte del programa también se ingresa el valor de distancia equivalente a 1 pulso dado por el sensor.

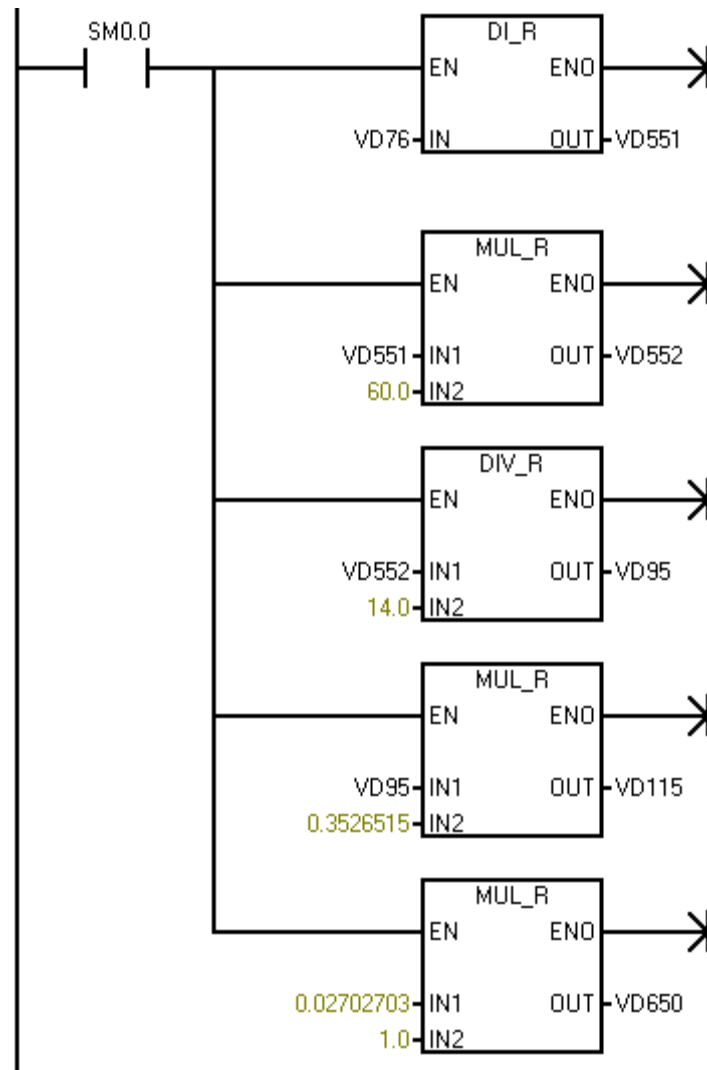


Figura 3.4 Operaciones matemáticas para RPM y Vel. de entrega

El contacto SM0.0 proviene de la marca especial 0.0, que se activa una vez que el PLC esta en modo RUN, al estar habilitados este contacto se habilita el juego de operaciones matemáticas para obtener los valores de RPM y velocidad de entrega, a continuación se encuentra la operación DI_R, esta operación nos transforma el dato de doble palabra VD76 a un dato en número real nombrado como VD551, seguido de esta operación tenemos la multiplicación de números reales entre la variable VD551 y la constante 60.0 (1 min=60.0 Seg), la salida de esta operación está nombrada por la variable real VD552, esta variable en la siguiente operación se la divide por el número de dientes de un piñón o el número de agujeros, en este caso la rueda agujerada tiene 14 hoyos, el resultado de esta división entre números reales se la expresa con la variable VD95 esta variable es visualizada en el TD-200 indicándonos el valor de RPM, la siguiente operación a realizarse es una multiplicación de números reales (coma flotante) entre el valor de las RPM (VD55) y la constante 0.3526515

¹ **Números reales.** Es el nombre que se le da a los datos que trabajan en coma flotante

para obtener un valor expresado en la variable VD75 que indica la Velocidad de entrega, este valor también se visualiza en TD-200, la última operación que tenemos en esta línea del programa es una multiplicación de reales (coma flotante) entre dos constantes la primera tiene el valor de equivalencia de distancia por pulso, este valor es 0.02702703 y la segunda es 1.0, se multiplica por 1 para no alterar el valor que se desea ingresar a la variable VD650, que se utiliza posteriormente para los cálculos de distancia que registra el nuevo contador.

Al programar un contador rápido automáticamente se genera el programa para la subrutina del contador en uso, en la figura 3.5 se indica el programa que el PLC genera automáticamente.

En este programa se puede realizar modificaciones de acuerdo a los bits de control del contador rápido, en estas funciones se tiene el control de velocidad del contador, los valores en los cuales empieza a contar, los valores de actualización de datos, el reset por software o un reset externo, la entrada física por la cual ingresa los pulsos, etc.

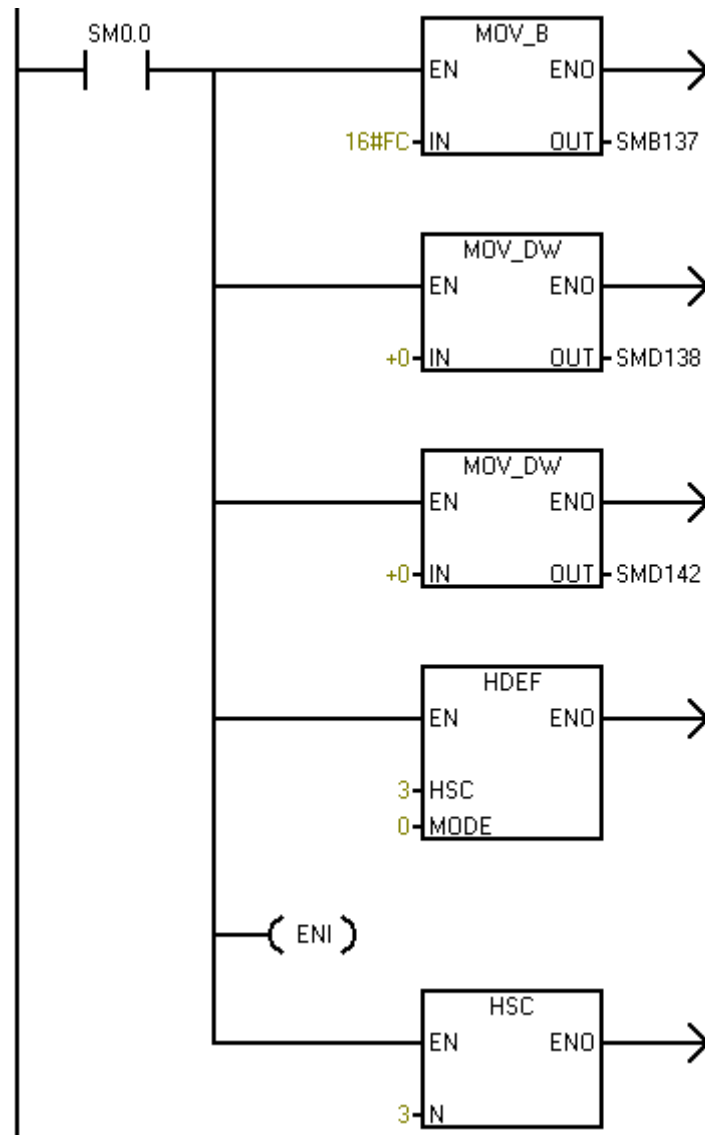


Figura 3.5 Programa de subrutina de contador rápido

El contacto SM0.0 permanece siempre habilitado por lo tanto todas las operaciones indicadas se ejecutan sin ninguna restricción, la operación MOV_B transfiere el valor hexadecimal 16#FC en la marca especial SMB137 esta marca trabaja a nivel de bits y es el bit de control del contador rápido, a continuación se transfiere el valor inicial (SMD138) y del valor actual (SMD142) del contador rápido mediante la operación MOV_DW, en este caso empieza desde 0 y no importa el valor actual ya que automáticamente se resetea cada segundo para entregar datos de frecuencia, que se explicó anteriormente, seguido tenemos la operación HDEF en la cual se ingresa el número de contador rápido que se está usando y el modo en que opera el contador rápido, a continuación tenemos la salida ENI, que asocia todos los elementos de interrupción, seguido tenemos el último paso de este programa, es la operación HSC en la cual debemos confirmar el contador rápido que estamos utilizando, en este caso es el número 3.

3.1.2 Selección de Turno y operaciones de conteo de material entregado.

Para el conteo de producción se dispone de 4 turnos, cada turno utiliza un contador rápido diferente para poder tener un conteo independiente entre los diferentes turnos, al igual que tacómetro cada turno esta basado en un contador rápido y su selección depende del el TD-200, que dispone de 4 teclas F1, F2, F3, F4 y la combinación con la tecla SHIFT que dan como resultado 8 combinaciones, en la siguiente tabla se indica las combinaciones y las marcas que corresponde a cada una de ellas.

COMBINACION	F1	F2	F3	F4	SHF F1	SHF F2	SHF F3	SHF F4
	MARCAS	M0.0	M0.1	M0.2	M0.3	M0.4	M0.5	M0.6

De esta manera el funcionamiento quedaría de la siguiente manera, al presionar F1 empieza a trabajar el Turno 1, al presionar F2 se suspende el Turno 1 y empieza a trabajar el Turno 2, es decir el turno que se seleccione trabaja mientras que los otros turnos restantes quedan suspendidos, las combinaciones SHIFT F1...F4 se la utiliza para resetear el contador descendente, este contador es independiente del contador general, este contador se puede ingresar un valor mediante el TD-200 de acuerdo a las exigencias del operador, es un contador descendente desde el valor que se ingresa por teclado, al momento que llega a "0" este da una señal para la máquina, la misma reacciona y realiza un cambio automático de bote y el contador se resetea sin afectar al contador general que lleva toda cantidad producida en el turno seleccionado, el diseño de software se lo detalla a continuación.

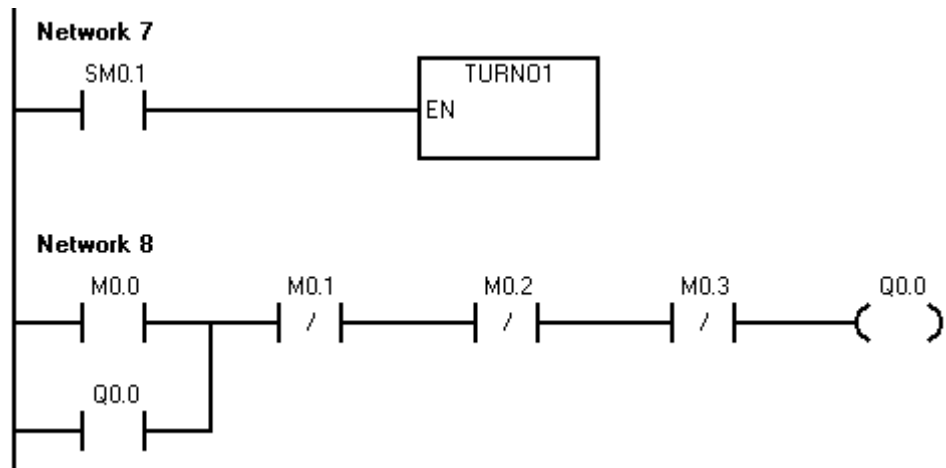


Figura 3.6 Selección de turno 1 para conteo.

En la figura 3.6 se indica los pasos de programación para seleccionar el Turno 1, primero se debe llamar a la subrutina “TURNO1” la misma que es generada automáticamente al programar un contador rápido, seguido de esto tenemos un diagrama básico de contactos con memoria que activan la salida Q0.0 los contactos M0.0, M0.1, M0.2, M0.3 son los usados por el turno 1, turno 2, turno 3, turno 4 respectivamente.

En la figura 3.7 tenemos el juego de operaciones necesarias para convertir los pulsos que se reciben desde el sensor de la máquina en expresiones matemáticas que se realizan para visualizar resultados esperados en el TD-200, se empieza con el contacto de la salida Q0.0 este contacto se cierra si está seleccionado el turno 1, después de este contacto tenemos la operación MUL_DI que es una multiplicación de doble entero, el valor del contador rápido HC0 es multiplicado por 1 y a la vez transforma valores desde un nivel de palabra a un nivel de palabra doble y guardado en la localidad de memoria MD28, este valor utiliza la memoria no volátil del CPU del PLC, en caso de cortarse la energía eléctrica se guarda el último valor de conteo.

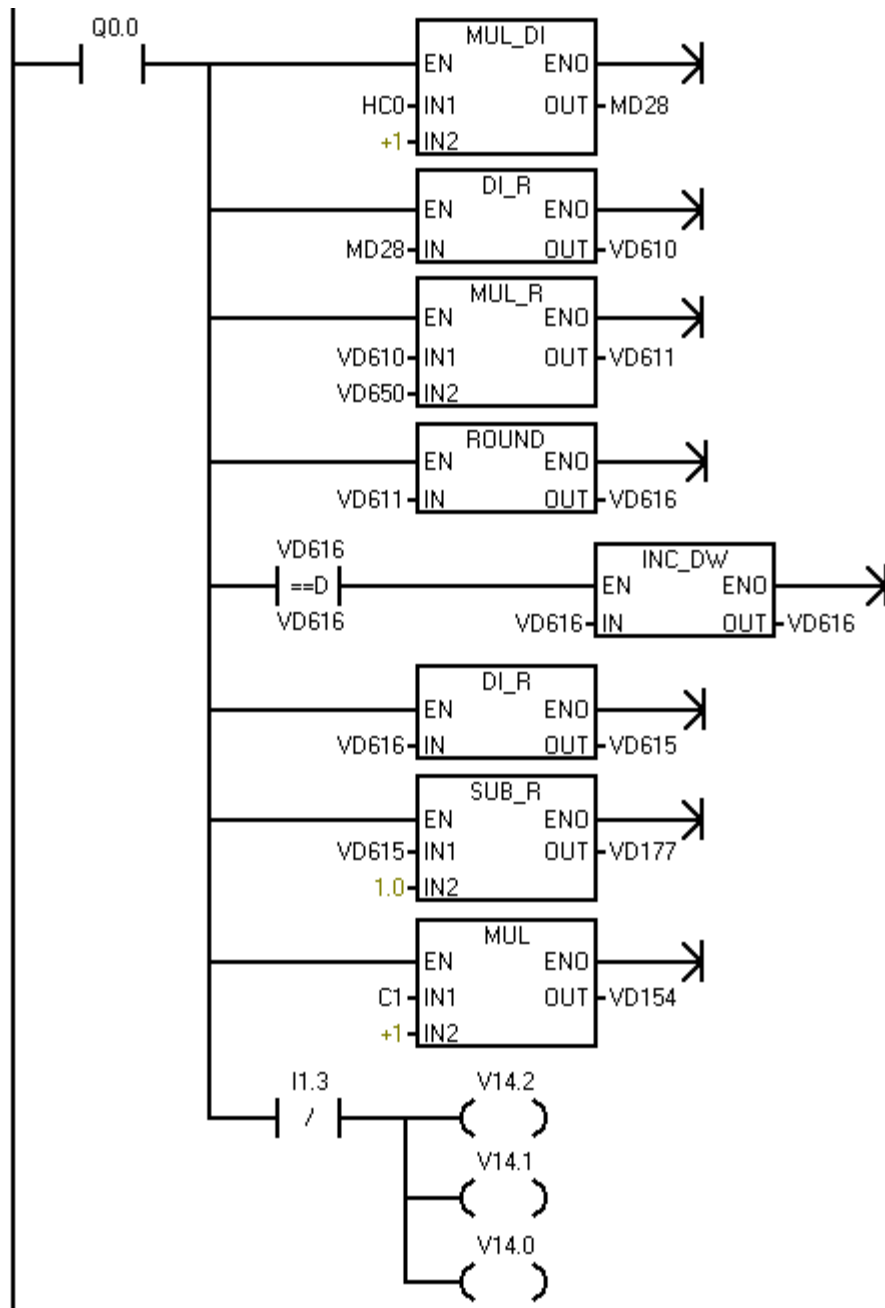


Figura 3.7 Operaciones necesarias para el contador general.

Después de la multiplicación de enteros dobles tenemos la transformación de un valor doble entero a un valor real (coma flotante), el valor de salida es la variable VD610 que en la siguiente operación se multiplica con la constante VD650, esta variable es una constante que tiene el factor equivalente a la distancia por cada pulso, estos valores se multiplican a nivel de variables reales, el resultado de este multiplicación se lo expresa por medio de la variable VD611, esta variable en la siguiente operación se hace un redondeo, expresando su resultado en la variable VD616, esta variable posee valores enteros que se necesitan en un contacto de comparación que habilita el incremento de esta variable (VD616) mediante la operación INC_DW, realizado el incremento de la variable indicada nuevamente se la transforma en un dato real que se carga a la variable VD615, con este resultado se tiene un desfase de 1m con el valor del contador general y el contador reciclable descendente, para eliminar este desfase se utiliza una resta de números reales, el valor de la

variable VD615 se le resta 1 y este resultado es visualizado en TD-200 a través de la variable VD177, este valor lleva la producción total, la siguiente operación es una multiplicación entre los valores de C1 y la constante 1, el resultado se lo expresa en la variable VD 154, la misma que es visualizada en el TD-200, este contador descendente y reciclable a valor de selección.

A continuación tenemos el contacto normalmente cerrado I1.3, este contacto actúa cuando se habilita el interruptor para “seteo de metraje del bote”, en este caso interrumpe solo la visualización de los parámetros de visualización del turno seleccionado que se despliega en la pantalla del TD-200.

En la figura 3.8 se indican las dos últimas líneas de programa correspondiente a un turno seleccionado, en la línea número 10 tenemos la programación del contador C1, empezamos con un contacto de comparación entre las variables VD177 y VD611, cuando el valor de estas variables reales son iguales proporcionan un pulso para que el contador pueda contar en forma descendente desde el valor que tiene la variable VW376 que se acciona mediante el interruptor “seteo de metraje de bote”, el mismo que activa en el TD-200 el ingreso de datos a través del teclado, una vez ingresado el valor se carga en la variable de palabra doble VW36.

Para resetear este contador se puede realizar de tres maneras, la primera con el contacto de comparación entre la variable VD154 y la constante cero, o sea cuando el contador llega a un valor de cero, se carga nuevamente en el valor que tiene la variable VW376, de esta manera tenemos un contador reciclable. Al mismo tiempo cuando la variable VD154 llega a cero también activa un temporizador Off-delay (T101) este temporizador está programado para 2 segundos, tiempo necesario para realizar el cambio de bote.

La segunda forma de reseteo es desde la máquina mediante el pulsador de “cambio de bote” al presionar este pulsador el contador se ubica en su posición inicial, y a la vez se da señal al temporizador T101 para enviar a la máquina una señal de cambio de bote.

La tercera forma es mediante el teclado del TD-200, presionando la tecla “SHIFT” y luego la tecla del turno seleccionado, se activa la marca M0.4 y resetea el contador pero no realiza el cambio de bote, de similar forma que lo realiza el contador EZA1 al presionar la tecla “Reset”.

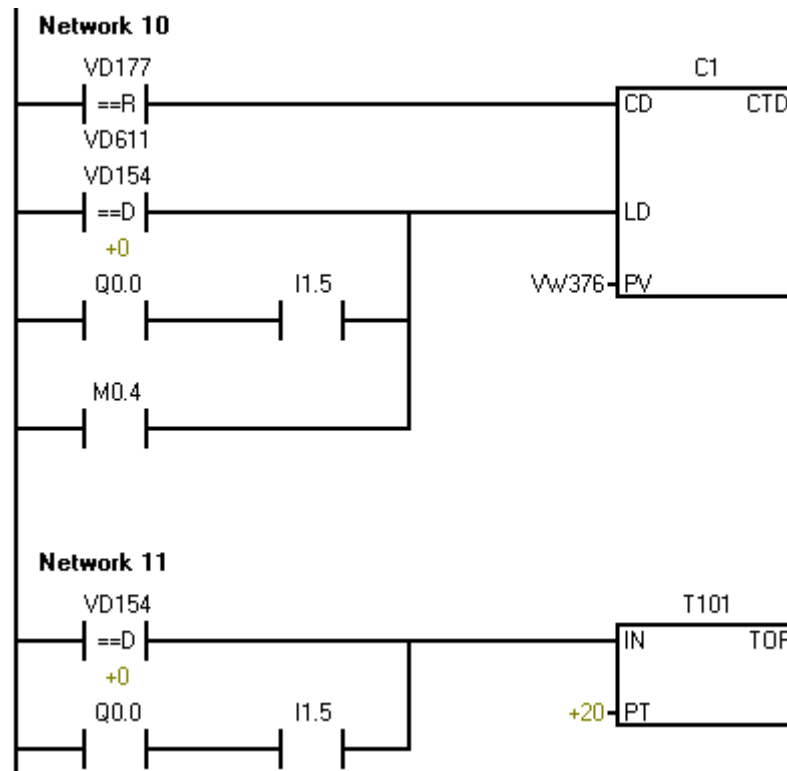


Figura 3.8 Programación de contador y temporizador para cambio de bote.

Todo este procedimiento de programación descrito hasta aquí es necesario para el funcionamiento de 1 turno de trabajo, para los siguientes turnos se utiliza el mismo procedimiento pero con diferentes variables, después de la programación para los cuatro turnos tenemos un juego de contactos para activar la salida Q0.4 la misma que activa un relé de 24V y los contactos de este relé activan en la máquina la señal de cambio de bote.

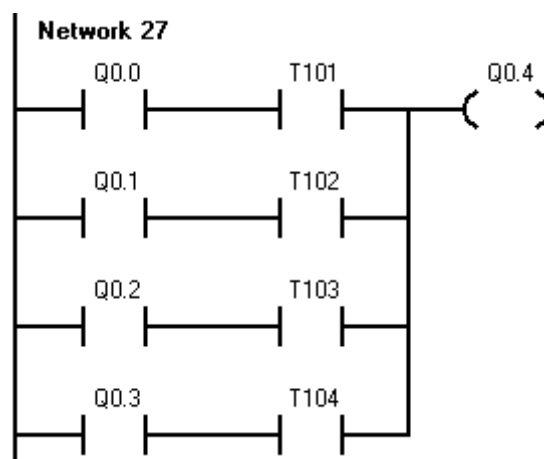


Figura 3.9 Distribución de contactos para activar Cambio de Bote.

En la figura 3.9 podemos observar que un juego de contactos tiene como final la salida Q0.4, los contactos Q0.0, Q0.1, Q0.2 y Q0.3 se activan cuando se selecciona el turno 1, 2, 3 y 4 respectivamente y los contactos desde T101 hasta T104 se activan cuando el contador descendente reciclable ha llegado a cero y en consecuencia se tiene que enviar a la máquina una señal de cambio de bote a través de la salida Q0.4.

En la figura 3.10 se indica como se realiza un retardo para que después de un determinado tiempo se pueda mostrar en la pantalla del TD-200 un mensaje que diga “REVISE LA MAQUINA”, este retardo consta de un temporizador T39 calibrado a 20Seg. el contacto de este temporizador On-delay, está en serie con el contacto I0.4 que es que recibe la señal de una falla en el funcionamiento de la máquina, la salida V14.7 es la que habilita la visualización en el display del PLC.

Este procedimiento se lo realiza a fin de evitar falsas alarmas cuando la máquina está funcionando.

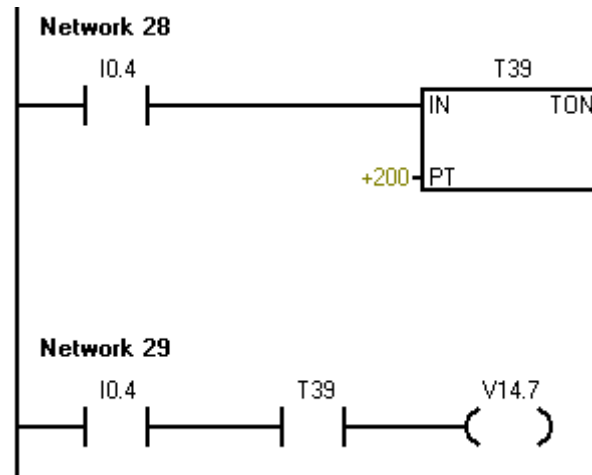


Figura 3.10 Retardo de visualización de mensaje en el PLC

En la figura 3.11 se indica el programa para tener acceso al ingreso de valores por medio del teclado del TD-200, la entrada I1.3 esta asociada directamente con el interruptor “Seteo de metraje de bote”, al accionarse este interruptor se habilita la salida V16.6 y el aviso de edición mediante la salida V374.2, además se bloquean las visualizaciones del display dando prioridad al valor que se ingresa por teclado.

Seguido tenemos al contacto normalmente cerrado I1.3, en serie con el contacto I1.1, esta entrada está asociada directamente con el pulsador que permite visualizar la velocidad angular de la máquina (RPM) y la velocidad de entrega del material producido, estos parámetros se los visualiza mediante la habilitación de las salidas V14.4 y V14.3 respectivamente, se debe indicar cuando se está ingresando valores desde el TD-200, no se puede visualizar ningún parámetro hasta que se termine el ingreso de los datos y se desactive el interruptor.

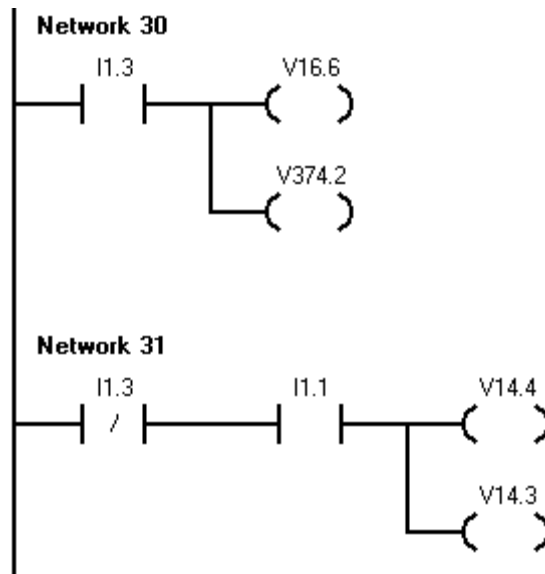


Figura 3.11 Pasos ingreso de datos y visualización de RPM y Vel/entrega

Para finalizar la descripción de software en la figura 3.12 se indica la implementación para indicar que el sensor de proximidad “n1” está en mal estado y se ha perdido el sensado y por lo tanto no hay señal de conteo, para verificar el estado de la fuente de alimentación del sensor en el panel frontal del nuevo contador existe un led rojo que siempre está iluminado indicando que la fuente de alimentación está en perfecto estado.

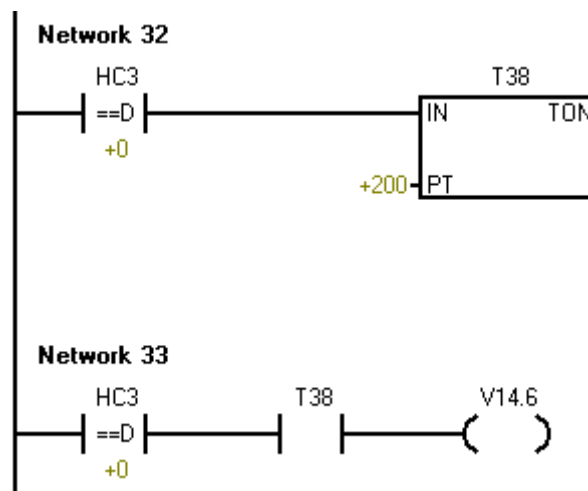


Figura 3.12 Implementación para falla en sensor

Para iniciar ésta implementación se lo hace con un contacto de comparación entre el valor del contador HC3 y la constante cero, cuando el sensor deje de emitir pulsos se pondrá en cero y habilita al temporizador T38 para que después de 20 segundos se habilite en el display el mensaje “FALLA EN SENSOR n1” a través de la salida V14.6.

3.1.3 Configuración de subrutinas.

Al momento que programamos un contador rápido automáticamente se genera una subrutina la misma que hay que configurar de acuerdo a la necesidad de la aplicación, en el caso de la subrutina “TACOMETRO” esta se genera al programar el contador rápido HC3, para realizar la configuración nos basamos en la siguiente tabla que contiene todas las funciones disponibles integrado en un byte de control, en la siguiente tabla se indica la función de cada uno de los bits que forman el byte de control.

BIT DE CONTROL	DESCRIPCION	VALOR TOMADO
SM137.0	Bit de control para nivel de actividad de la entrada de puesta a 0: 0 = actividad alta; 1 = actividad baja	0
SM137.1	Bit de control para nivel de actividad de la entrada de arranque: 0 = actividad alta; 1 = actividad baja	0
SM137.2	Velocidad de conteo de los contadores A/B: 0 = velocidad cuádruple; 1 = velocidad simple	1
SM137.3	Bit de control para el sentido de conteo: 0 = conteo atrás; 1 = conteo adelante	1
SM137.4	Escribir el sentido de conteo en el contador rápido: 0 = no actualizar; 1 = actualizar el sentido de conteo	1
SM137.5	Escribir el nuevo valor predeterminado en el contador rápido: 0 = no actualizar; 1 = actualizar el valor predeterminado	1
SM137.6	Escribir el nuevo valor actual en el contador rápido: 0 = no actualizar; 1 = actualizar el valor actual	1
SM137.7	Habilitar el contador rápido: 0 = inhibir el contador rápido; 1 = habilitar el contador rápido	1

En este caso se ha formado la palabra hexadecimal “FC”, este valor se carga a la marca especial SMB137, en la figura 3.13 se indica la subrutina creada automáticamente al crear el contador rápido HC3.

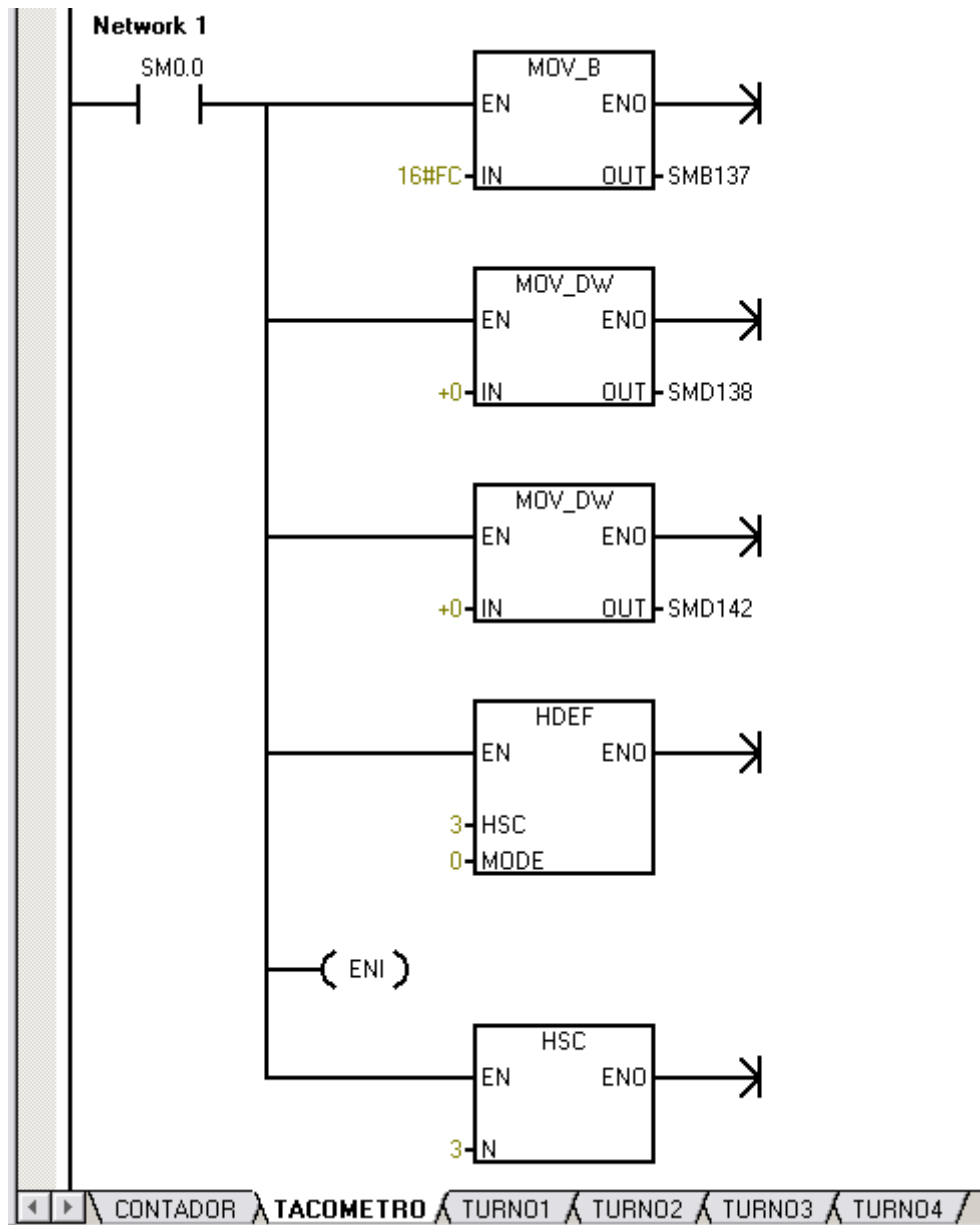


Figura 3.13 Subrutina "TACOMETRO"

La subrutina generada por la utilización del contador rápido HC3 se puede apreciar en la figura 3.13, esta subrutina empieza con la carga del byte de control a la respectiva variable, en este caso SMB137, este valor se lo describió anteriormente, seguido tenemos la carga del nuevo valor actual en este caso cada segundo se registra nuevos valores y retiene el valor tomado 1mSeg. antes de que se complete el periodo de 1 Seg. este valor entrega la variable SMD138 que tiene un valor de cero si no registra pulsos en el HC3, después de esta operación tenemos el valor inicial desde el cual arranca el contador HC3 esta referencia indica la variable SMD142 que tiene un valor asignado de cero, como siguiente operación tenemos la definición de modo del contador HC3 esta operación (HDEF) nos permite seleccionar el contador rápido a usarse y el modo de operación del mismo, a continuación tenemos la salida (ENI) que habilita la ejecución de todos los eventos asociados a la interrupción, por último tenemos la operación activar contador rápido (HSC) esta configura y controla el funcionamiento del contador rápido direccionado, basándose en el estado de las marcas especiales del mismo, además se asigna el número de contador rápido que se esta usando.

De manera similar son el resto de subrutinas creadas al utilizar el contador asignado para cada turno, cada subrutina tiene el nombre del turno asignado como se indica desde la figura 3.14 hasta la figura 3.17.

Cabe indicar que la operación que carga el valor actual del contador se carga una variable de memoria que retiene el último valor del contador en caso de que se corte la energía al PLC, estas variables son MD25, MD26, MD27 y MD28, utilizadas en los turnos 1, 2, 3 y 4 respectivamente.

Para el turno 1 se ha utilizado el HC0, para el turno 2 el HC1, en el turno 3 se utilizó el HC2 y HC4 para el turno 4.

Cada una de las subrutinas se muestra en las siguientes figuras.

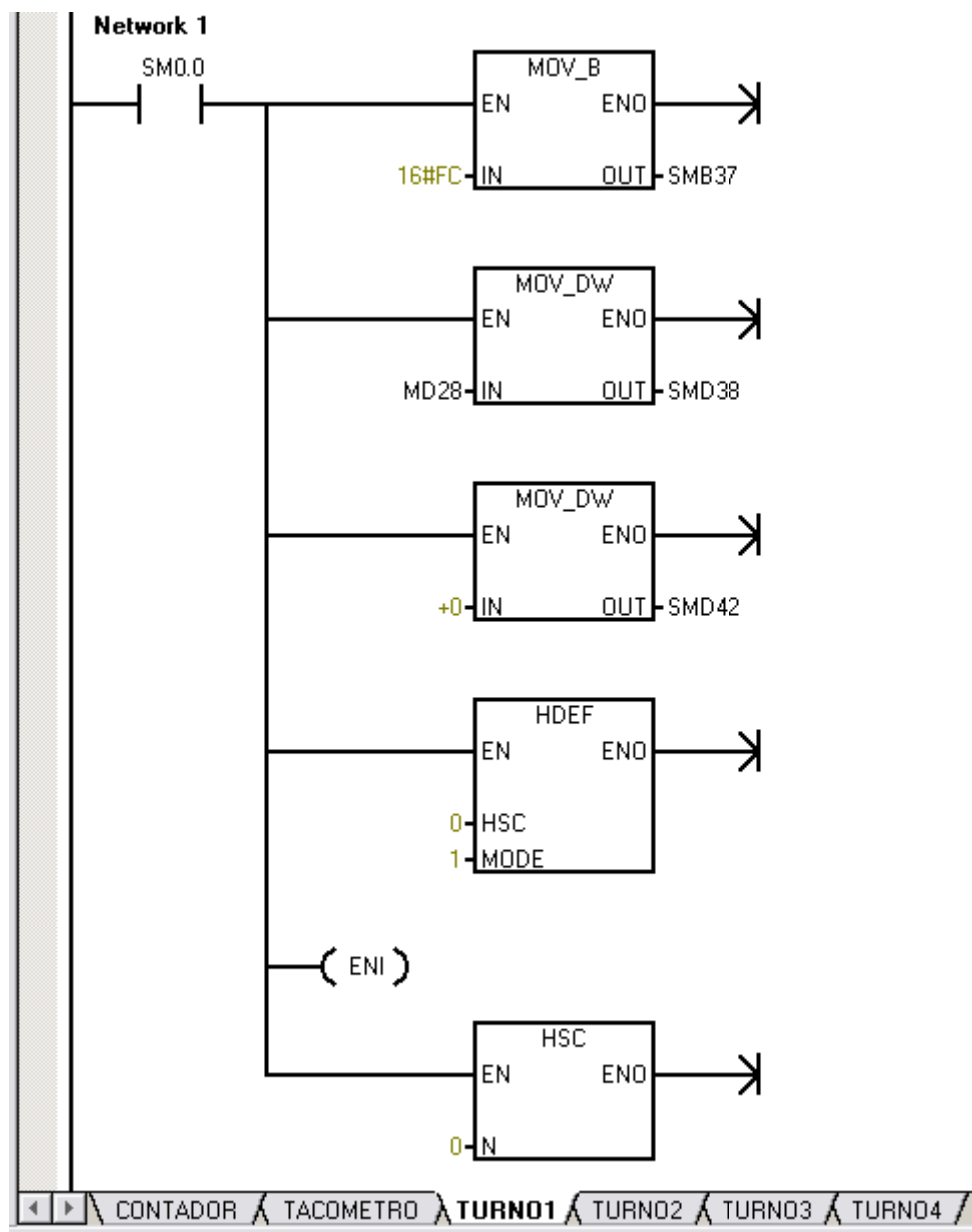


Figura 3.14 Subrutina para el Turno 1.

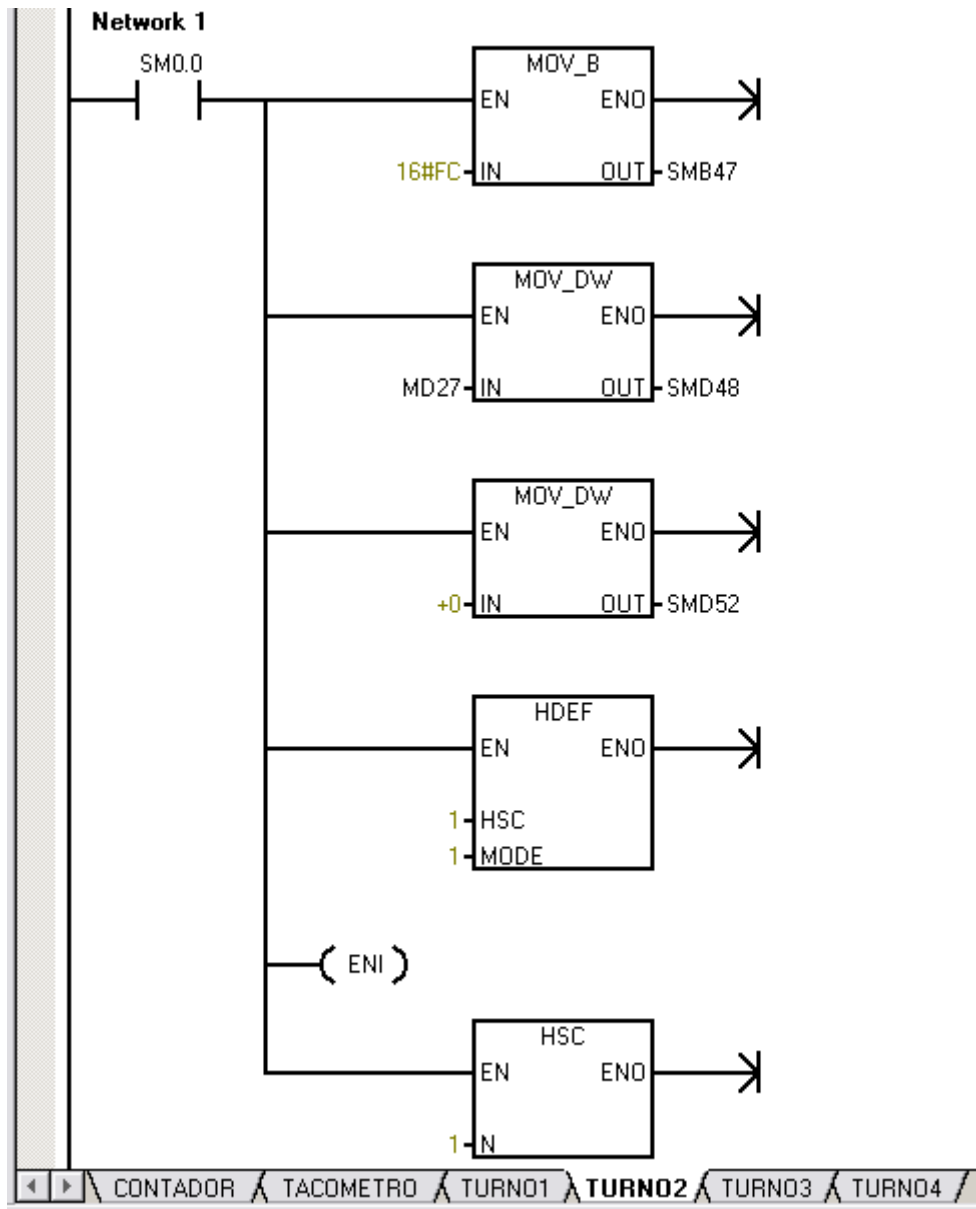


Figura 3.15 Subrutina para el Turno 2

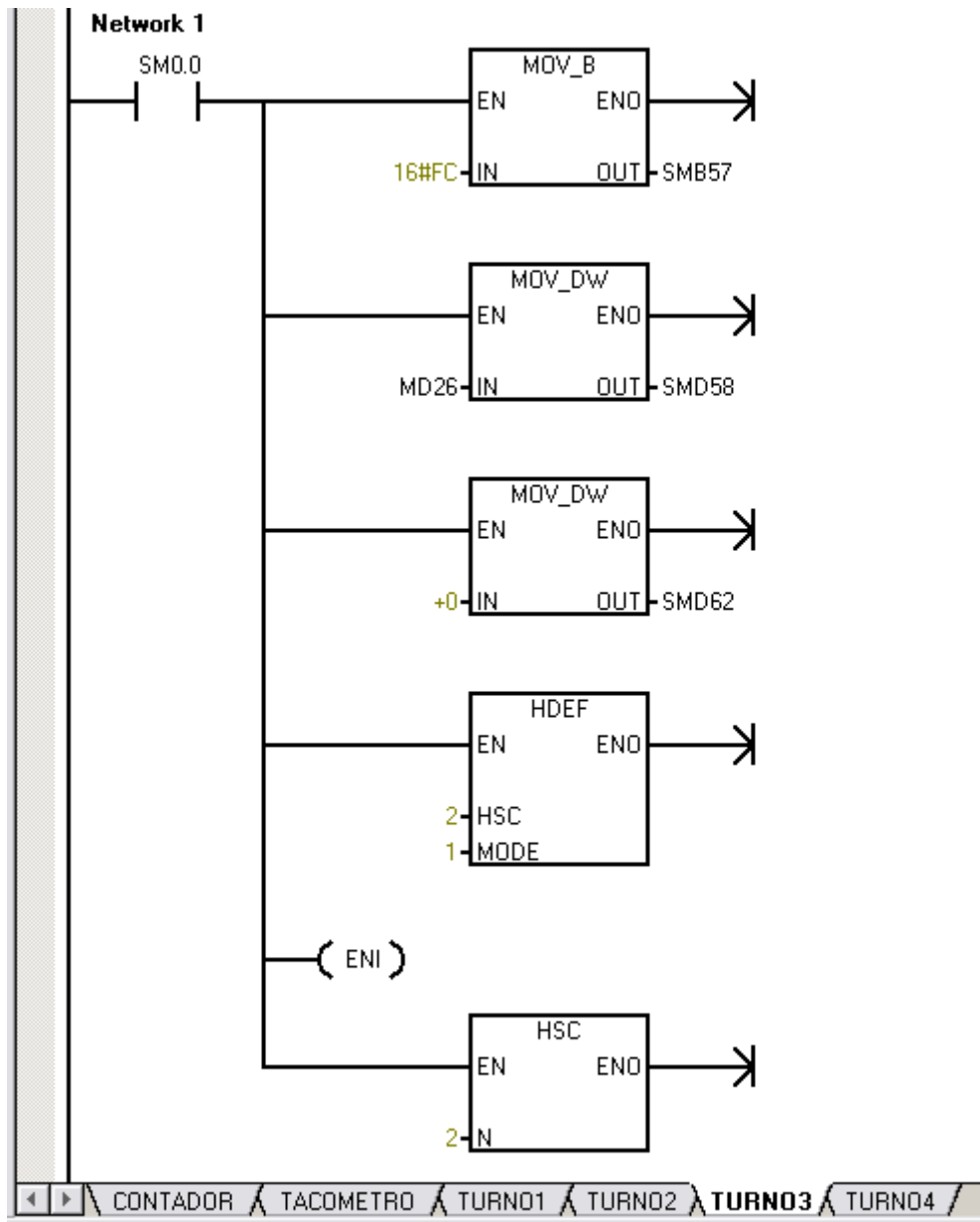


Figura 3.16 Subrutina para el Turno 3

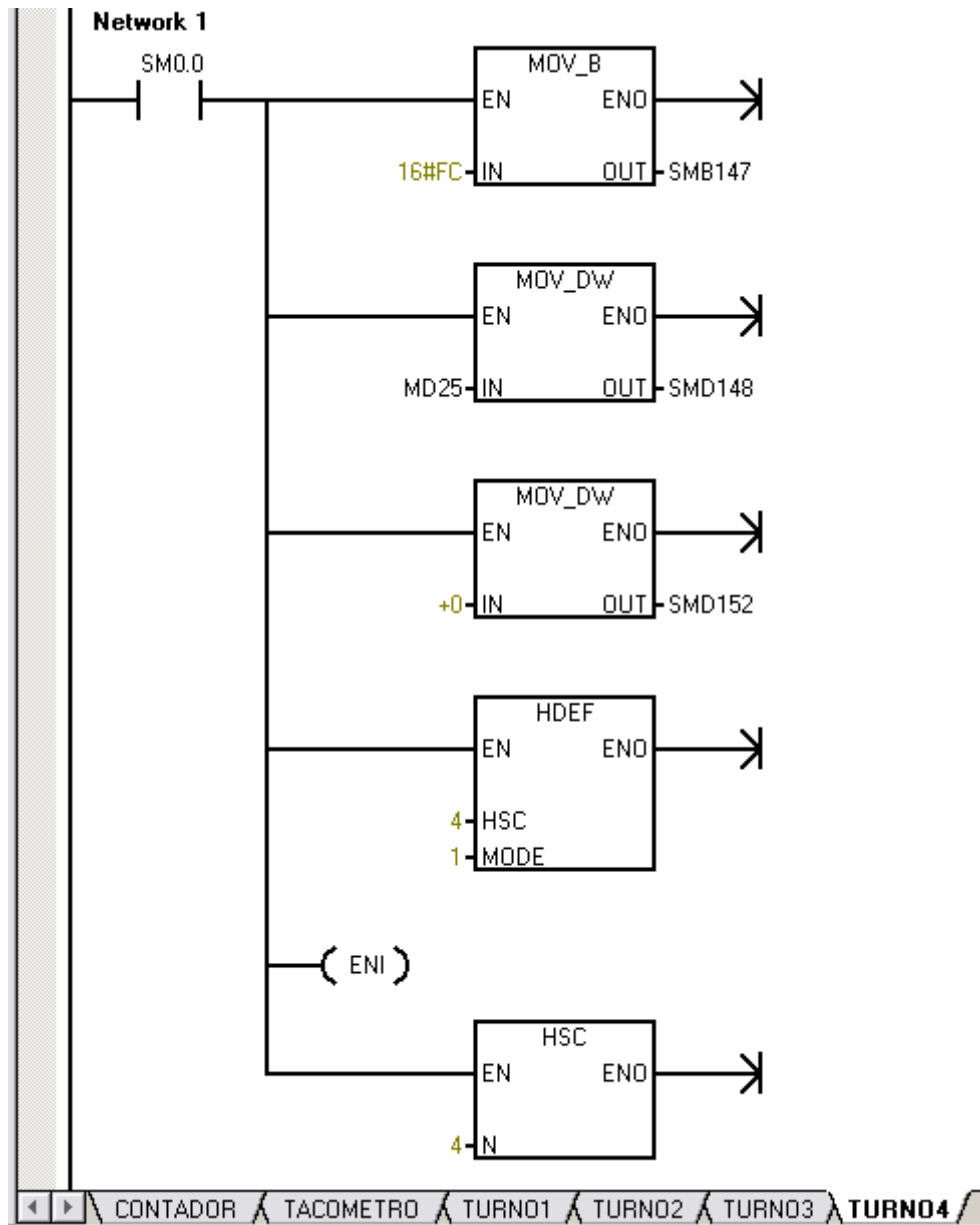


Figura 3.17 Subrutina para el Turno 4

3.1.4 Configuración de Visualizador TD-200.

Dentro de la configuración del visualizador de texto TD-200 se debe tomar en cuenta la prioridad de los mensajes a visualizarse, en este proyecto los textos de alarma son los primeros en visualizarse, la siguiente visualización son las RPM, velocidad de entrega, las siguientes visualizaciones son las de cada uno de los turnos, aquí se visualizan el estado del contador del metraje de bote, también se visualiza el metraje totalizado entregado en el turno seleccionado.

Por último se activa el ingreso del valor de metraje de bote, el mismo que se lo realiza a través del teclado del TD-200.

Para todas las visualizaciones se utilizaron un total de 18 mensajes de 20 caracteres, en la siguiente tabla se indica cada uno de los mensajes y la variable asignada para realizar operaciones.

Nº	TEXTO DE MENSAJE	VARIABLE ASIGNADA
1	R E V I S E L A M A Q U I N A	
2	F A L L A E N S E N S O R n 1	
3	F r e c u e n c i a H z	VD76
4	V e l o c i d a d R P M	VD95
5	V . E n t r e g a m / m i n	VD115
6	* * * * * T U R N O 1 * * * * *	
7	m e t / b o t e m e t r o s	VD154
8	T 1 P r o d . T o t . m t s	VD177
9	* * * * * T U R N O 2 * * * * *	
10	m e t / b o t e m e t r o s	VD214
11	T 2 P r o d . T o t . m t s	VD237
12	* * * * * T U R N O 3 * * * * *	
13	m e t / b o t e m e t r o s	VD274
14	T 3 P r o d . T o t . m t s	VD297
15	* * * * * T U R N O 4 * * * * *	
16	m e t / b o t e m e t r o s	VD334
17	T 4 P r o d . T o t . m t s	VD357
18	C a n t . b o t e ? m e t r o s	VW376

Mensajes programados en visualizador TD-200

3.2 DISEÑO DE HARDWARE.

El diseño de hardware está basado en las conexiones físicas tanto de las entradas como de las salidas, cada contador rápido que maneja el conteo de un turno seleccionado tiene una entrada física en la que ingresan valores que se aprecian como frecuencia, las salidas del PLC controla dos relés de 24V los mismos que con sus contactos dan la señal de cambio de bote y un contacto que indica que el contador esta listo para que la máquina pueda funcionar con normalidad, tal como lo haría el contador EZA1.

3.2.1 FUENTE DE ALIMENTACION

El diseño del contador EZA-1 internamente tiene una fuente de alimentación de 24V, la misma que alimenta los circuitos de mando y a los sensores de proximidad, los mismos que entregan la señal de frecuencia al PLC, la fuente interna del PLC no abastece para manejar toda la carga de 24V, en virtud a este problema se diseñó una fuente adicional para que no exista ningún problema en el abastecimiento de 24Vdc.

La fuente se la realiza en base a un regulador de voltaje LM317 que es un regulador de voltaje variable con una corriente de salida de 1.5A, la fuente de alimentación se la realiza como se indica en la figura 3.18, en donde se observa los elementos de transformación, rectificación, filtrado y regulación.

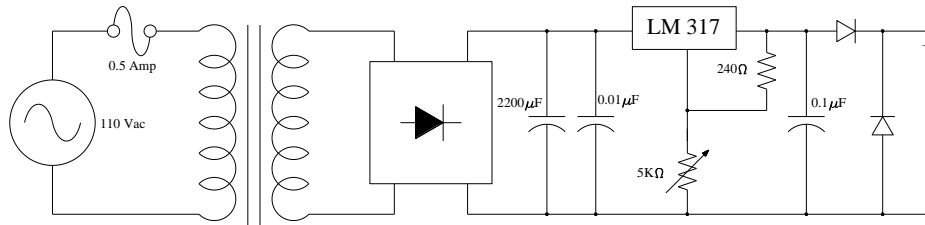


Figura 3.18. Fuente Regulada de 24Vdc.

3.2.2. CONEXIONES DE ENTRADAS Y SALIDAS DEL PLC.

Las conexiones de las entradas del PLC se basan en las entradas del contador EZA1, las entradas que dan señal son: reset externo de bote, reset, falla de máquina, la entrada de sensado (frecuencia) se maneja conjuntamente entre las entradas y salidas del PLC.

Adicionalmente se utilizaron otras entradas para visualización de velocidad de entrega y velocidad de la máquina (RPM).

Un grupo de las salidas se combinan con las entradas y otro grupo de salidas controla a dos relés de 24V, uno de los relés se utiliza para realizar el cambio automático de bote, y el otro relé da una señal que indica que el contador está operando normalmente y la máquina puede continuar funcionando, en la figura 3.19, se indica las conexiones realizadas en las entradas y salidas del PLC.

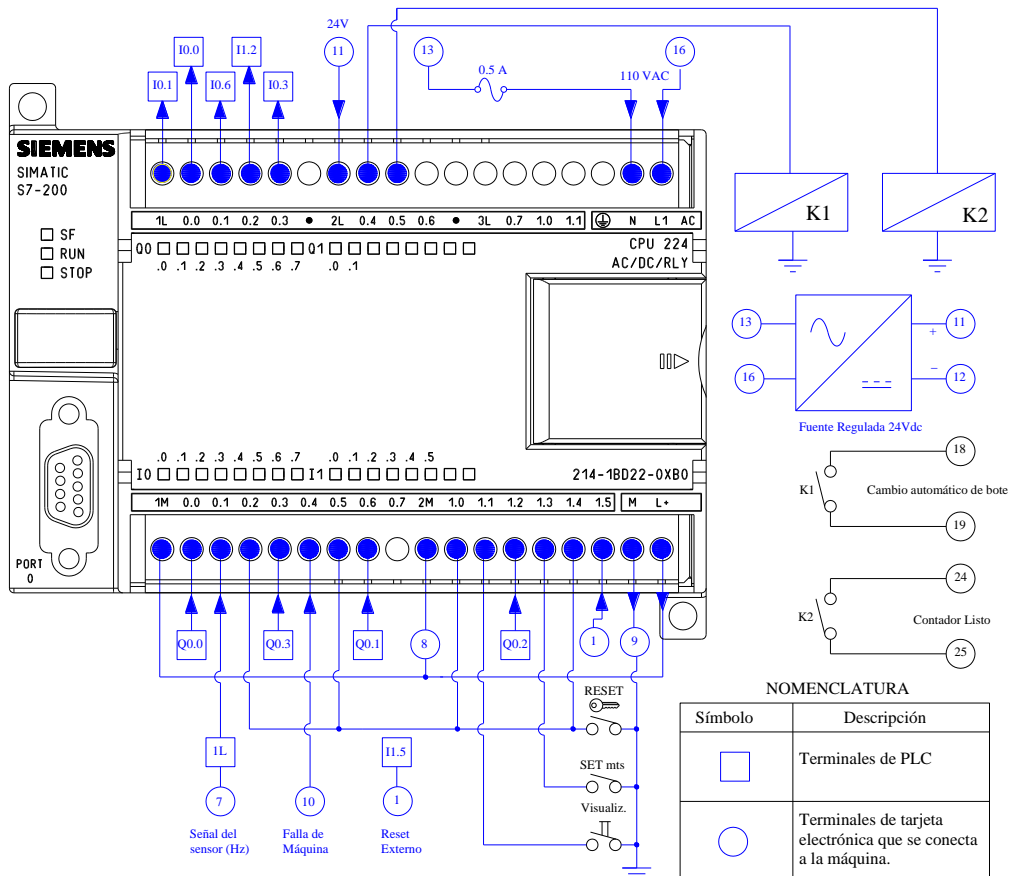


Figura 3.19. Conexiones del PLC.

3.2.3. DISEÑO DE TARJETA ELECTRONICA.

En base a las conexiones del PLC y tomando en cuenta las dimensiones de la tarjeta electrónica original, se realiza la ubicación de los conectores, protecciones y demás elementos electrónicos, también se toma en cuenta las entradas y salidas utilizadas del PLC, se procede al trazado de las pistas, la comprobación respectiva hasta obtener la tarjeta final, en la figura 3.20 se indica la tarjeta electrónica utilizada para el contador.

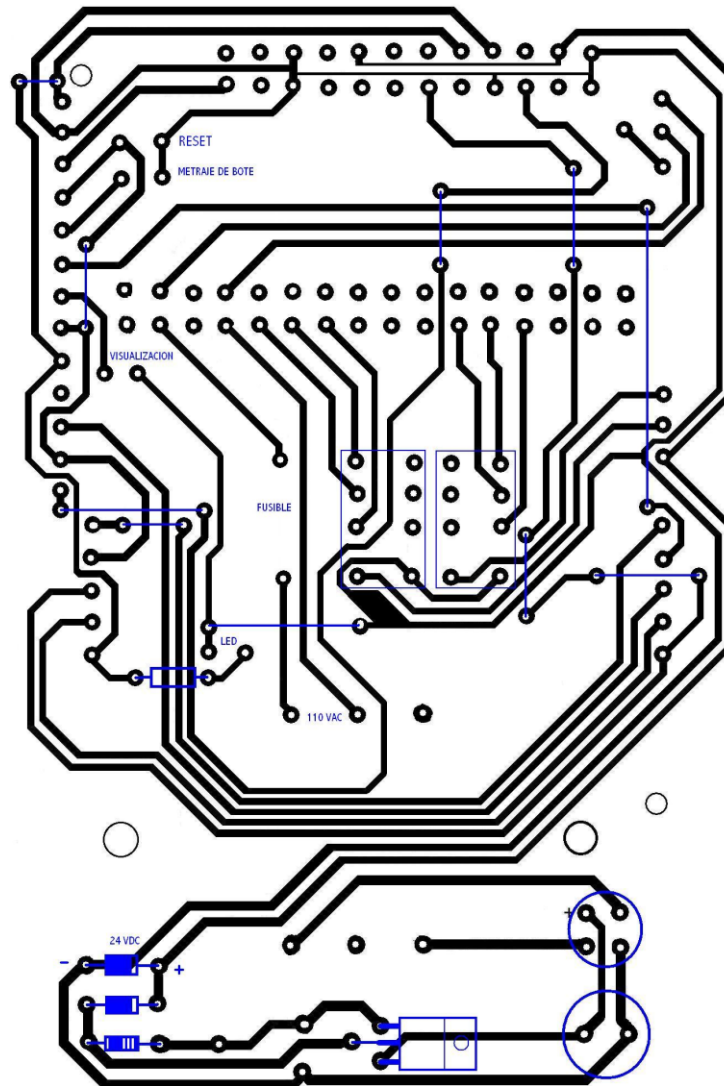


Figura 3.20. Diseño de tarjeta electrónica.

Una vez realizado el diseño se procede a la fabricación de la misma, empezando a dibujar con marcador indeleble sobre la parte conductora de la placa de baquelita, una vez que se han trazado todas las pistas, se introduce la placa en una solución de ácido perclórico el mismo que corroe y elimina todo el cobre que no está protegido por el marcador indeleble, una vez terminado este proceso, se realiza una completa limpieza de la placa de todos los restos ácidos para posteriormente realizar el montaje de los elementos electrónicos y la suelda de los mismos (ver figura 3.21), una vez terminada la tarjeta se procede a la conexión de las entradas y salidas del PLC (ver figura 3.22) terminado el montaje del montaje del PLC sobre la tarjeta electrónica se realiza el montaje del sistema contador en el chasis de iguales dimensiones al chasis original del contador EZA-1.

Una vez terminado todo el contador se procede al montaje en la máquina, así como se observa en la figura 3.23, para realizar las respectivas pruebas y calibraciones.

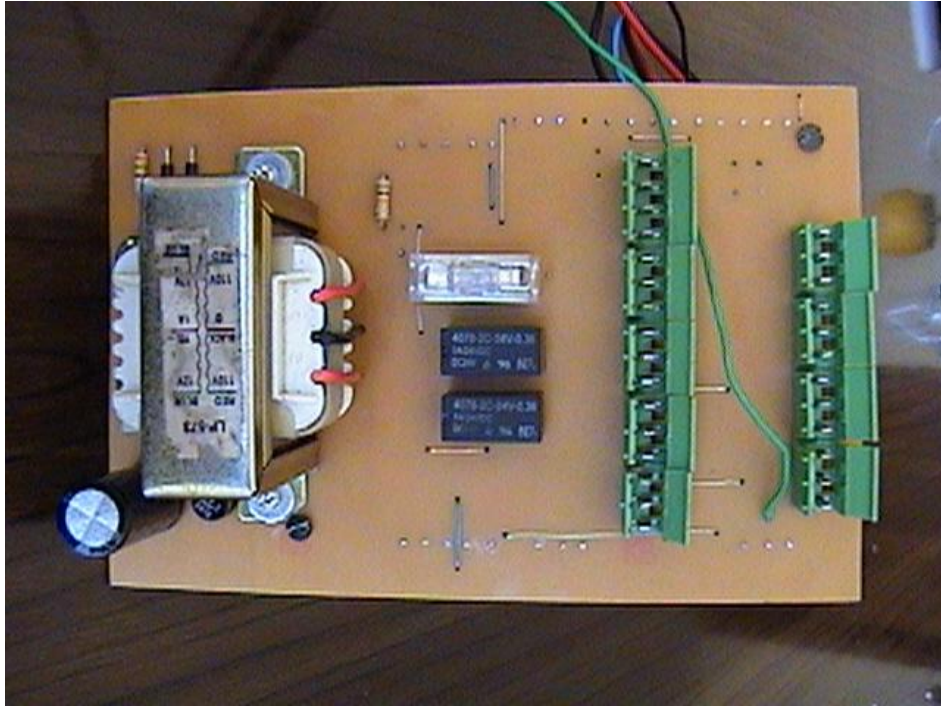


Figura 3.21. Vista final de tarjeta electrónica

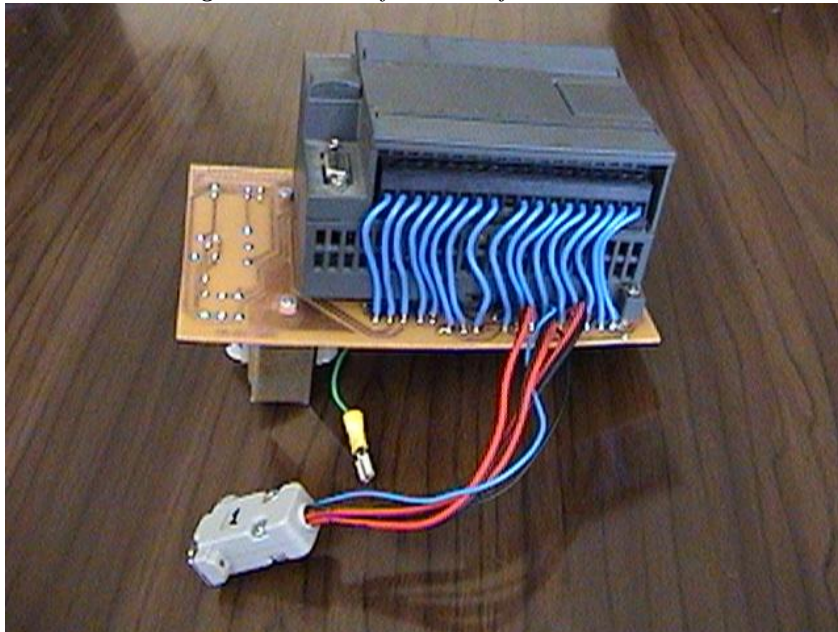


Figura 3.22. Conexión de PLC y tarjeta electrónica



Figura 3.23. Contador finalizado e instalado en la maquina.

3.3 PRUEBAS DE FUNCIONAMIENTO.

Las pruebas de funcionamiento se las realizó en cada una de las partes que está conformado el contador:

3.3.1 Tacómetro.

Para realizar el tacómetro se empezó realizando un tren de pulsos que genere una función impulso cada segundo, esta señal inicializa un contador el mismo que se encarga de totalizar un número determinado de pulsos contemplados dentro de 1 Seg., si analizamos lo que significa pulsos por segundo podemos concluir que un pulso es un ciclo completo lo que equivale a decir que pulsos por Seg. es igual que ciclos por segundo equivalente a Hertz que es una unidad de frecuencia, por lo tanto el tacómetro diseñado también es un frecuencímetro.

Al realizar una contrastación con un frecuencímetro FLUKE, y la ayuda de un osciloscopio nos damos cuenta que el frecuencímetro diseñado funciona a la perfección, ya que las lecturas visualizadas en el display TD-200 es la misma que registra el multímetro FLUKE al igual que la frecuencia que se aprecia en el osciloscopio.

Al realizar estas pruebas surge el primer problema, el frecuencímetro registra lecturas de hasta 80Hz, las entradas de los contadores del PLC se saturan desde este valor, manteniendo la entrada en estado alto "1", el problema radica en que la frecuencia entregada por los sensores de las máquinas opera entre 80Hz y 100Hz, este problema se solucionó utilizando contadores rápidos, los mismos se los programa de acuerdo a la necesidad y aplicación que se requiere realizar.

La especificación del fabricante nos indica que la frecuencia de entrada de un contador rápido es de máximo 30Khz, una vez configurado y programado el contador rápido se realizó las respectivas pruebas de contrastación obteniendo resultados favorables, el valor indicado en la pantalla del TD-200 es la misma indicada en el multímetro FLUKE, y la medición en la pantalla del osciloscopio es la misma, el frecuencímetro es una parte fundamental en el funcionamiento del contador, ya que en base al frecuencímetro podemos obtener el tacómetro en el que se pueda medir en revoluciones por minuto (RPM), mediante un cálculo matemático ingresado en el programa del PLC, el calculo introducido en el programa esta basado en la siguiente fórmula.

$$RPM = \frac{Frecuencia \times 60}{ND}$$

Donde: Frecuencia= Pulsos/Segundo
60= 1 minuto tiene 60 Seg.
ND= Número de dientes ó agujeros.

3.3.2 Selección de Turno.

Inicialmente se tenia diseñado usar un switch de cuatro posiciones, cada una de ellas selecciona un turno para tener un totalizado de material entregado, aquí se da el segundo problema, al usar un contador rápido por cada turno y de acuerdo a la configuración de entradas de cada uno, las entradas disponibles en el PLC son apenas 2, para solucionar este problema se buscó un medio de utilizar el teclado del TD-200, al tener cuatro teclas F1, F2, F3 y F4, se realizo el respectivo estudio de las configuraciones para utilizar las teclas del display TD-200, con la respectiva programación de las variables asignadas para manejar el teclado del TD-200 se solucionó el problema para escoger cada uno de los turnos.

También se implementó un reset de conteo de bote, esta opción la realiza el contador EZA-1, la tecla reset inicializa el contador sin realizar el cambio de bote, en el contador en base al PLC realiza la misma función pero de diferente manera, al presionar la tecla SHIFT y luego la tecla del turno en operación se inicializa el contador sin realizar el cambio de bote de igual manera que lo realiza el contador EZA-1.

Al igual que en el contador EZA-1, el nuevo contador dispone de un contacto el cual da señal a la máquina indicando que el contador está listo para funcionar, es decir el contador está energizado, en caso de que el sensor de proximidad esté dañado o sea no entrega pulsos se visualiza en el TD-200 el mensaje “falla en sensor n1”, y de igual manera si uno de los componentes de la máquina no se encuentra listo para operar, mediante una entrada del PLC se recibe señal de la máquina indicando que la máquina no está lista, esto se visualiza el display TD-200 con un mensaje “Falla en la máquina”

3.4 MANUAL DE USUARIO.

3.4.1 Identificación de partes.

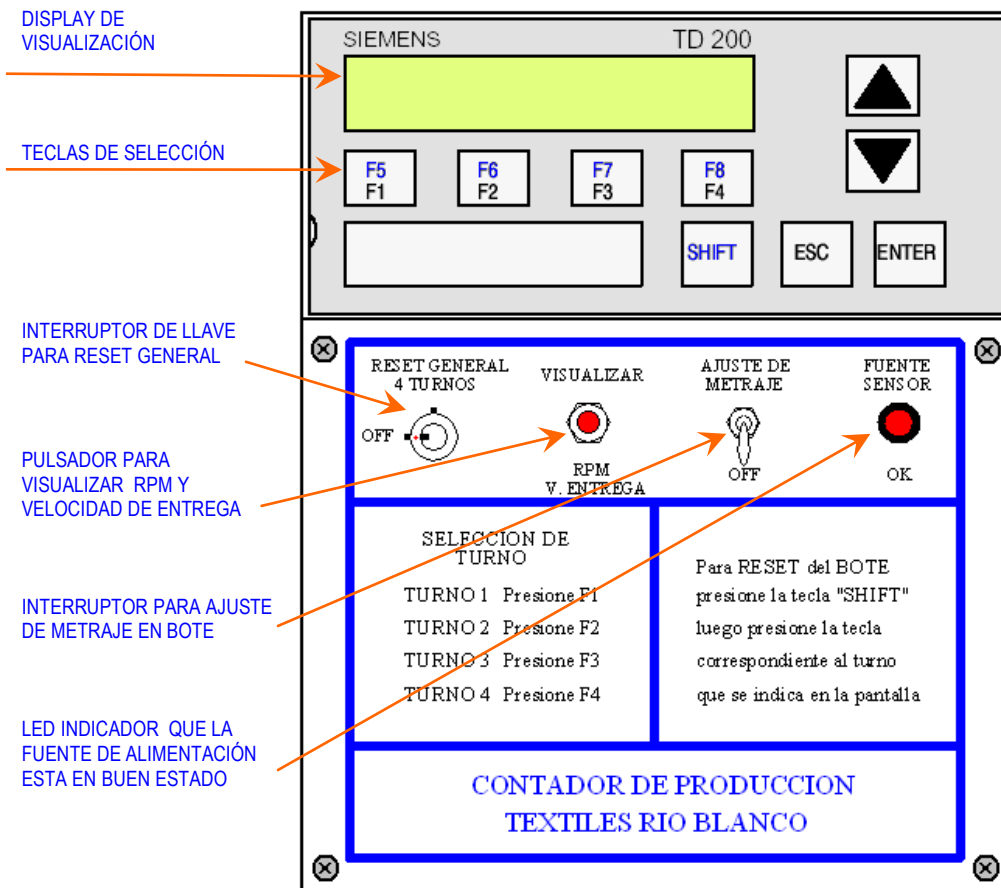


Figura 3.24. Identificación de partes del panel frontal del nuevo contador.

En la figura 3.24 se puede apreciar el panel frontal del nuevo contador, en este panel se localiza al visualizador de textos TD-200, el mismo que consta de una pantalla LCD de iluminación verde que permite visualizar textos en dos líneas, además tiene 9 teclas las cuales se detallarán posteriormente.

Siguiendo con el panel de control se observa que tiene un interruptor con llave que se utiliza para el reset general.

También dispone de un pulsador para la visualización de otros textos.

Un interruptor para realizar el ajuste de metraje de bote.

Finalmente un indicador LED rojo, el cual permanece encendido cuando la fuente de alimentación está en perfecto estado.

3.4.2 Selección de Turno.

Para la selección de turno debemos presionar las teclas que están marcadas como F1, F2, F3 y F4 de esta manera se escoge el Turno 1, Turno 2, Turno 3 ó Turno 4 respectivamente, por ejemplo si presionamos la tecla F1, como se indica en la figura 3.25

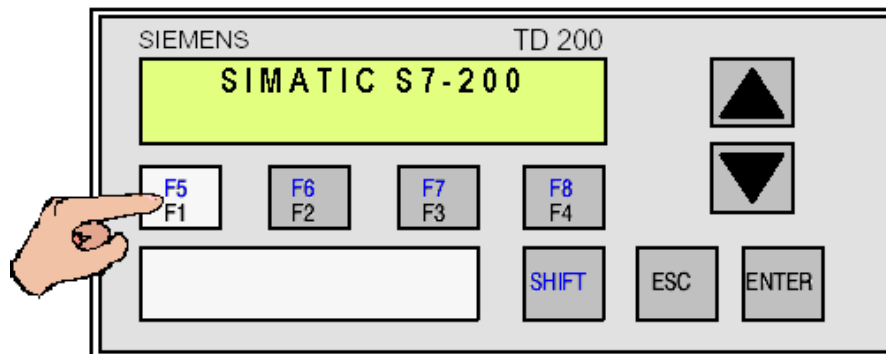


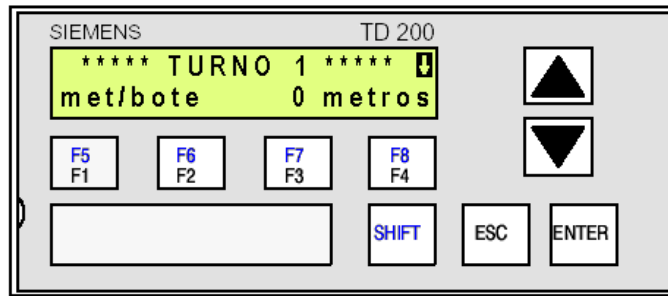
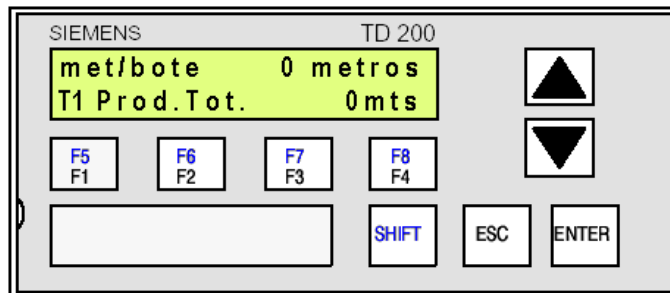


Figura 3.25 Pantalla inicial cuando se enciende el contador por primera vez, tecla F1 presionada

Hemos seleccionado el turno 1, en la pantalla nos aparece la información del estado del metraje del bote, también podemos observar el símbolo intermitente , este nos indica que hay más mensajes por visualizar para lo cual se debe presionar la tecla  para observar todos los mensajes hasta que el símbolo intermitente desaparezca, en este caso se puede observar la cantidad totalizada del material entregado en el turno seleccionado, en la figura 3.26 se observa las pantallas con la información del turno seleccionado, en este caso el turno 1.



a) Pantalla después de selección de turno



b) Visualización de estado de bote y producción totalizada

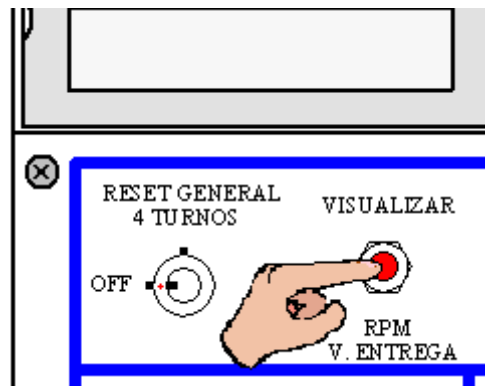
Figura 3.26. Selección e información de turno seleccionado.

Para realizar un cambio de turno solamente se debe presionar la tecla correspondiente al deseado, en la siguiente tabla se indica los turnos con la respectiva tecla de selección, al momento de cambiar de turno toda la información se queda guardada, en caso de retornar al mismo turno el contador continúa con su función desde el último valor en el cual se quedó totalizado.

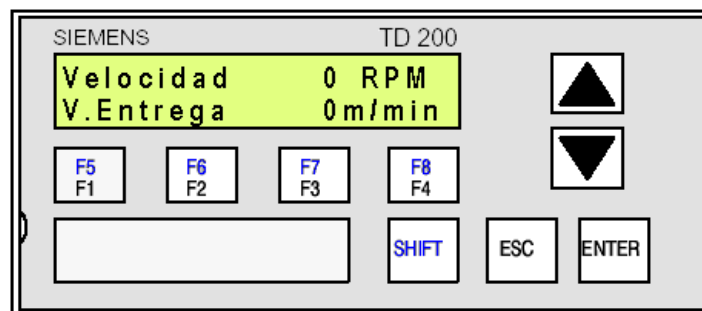
TURNO A SELECCIONAR	TECLA DE SELECCIÓN
TURNO 1	F5 F1
TURNO 2	F6 F2
TURNO 3	F7 F3
TURNO 4	F8 F4

3.4.3 Visualización de RPM y Velocidad de entrega.

Para visualizar los valores de Velocidad (RPM) y velocidad de entrega se debe presionar el pulsador “VISUALIZAR”, como se indica en la figura 3.27, al realizar esta acción en la pantalla se oculta la información de metrajes y se visualiza los datos de Velocidad de máquina (RPM) y la velocidad de entrega de material, esto ocurre mientras se tenga presionado el pulsador, una vez que se deje de pulsar en la pantalla se visualiza los datos del turno seleccionado, es decir el metraje de bote y el totalizado de material entregado.



a)



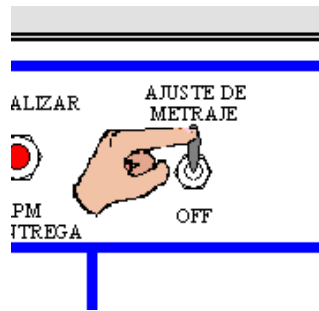
b)

Figura 3.27. a) Pulsador “VISUALIZAR” presionado b) Visualización en la pantalla.

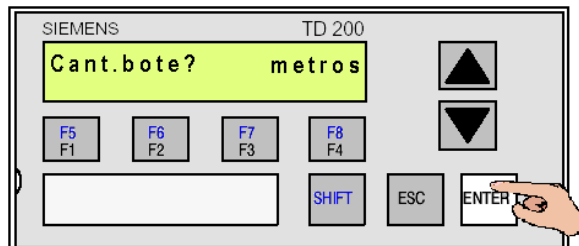
3.4.4 Programación para el metraje de bote.

Para programar la cantidad de metros que debe contener un bote se acciona el interruptor “AJUSTE DE METRAJE”, en la pantalla se despliega el mensaje “Cant. Bote?”.

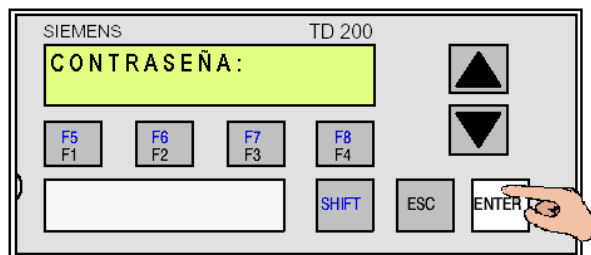
Seguido presionamos la tecla , en la pantalla se aparece el mensaje “CONTRASEÑA”, se ingresa la contraseña con la ayuda de las teclas y seleccionamos los números que forman parte de la contraseña que consta de cuatro dígitos, una vez ingresada presionamos la tecla , el mensaje que aparece en la pantalla nos dice “Cant. Bote?” y el cursor que aparece en forma intermitente, con la ayuda de las flechas del teclado fijamos el valor al cual queremos que se llene el bote de material, una vez seleccionado presionamos la tecla “ENTER”, posteriormente se debe poner el interruptor “AJUSTE DE METRAJE” en la posición OFF, en la figura 3.28 se observa todo el procedimiento para programar un valor de metraje de bote de material.



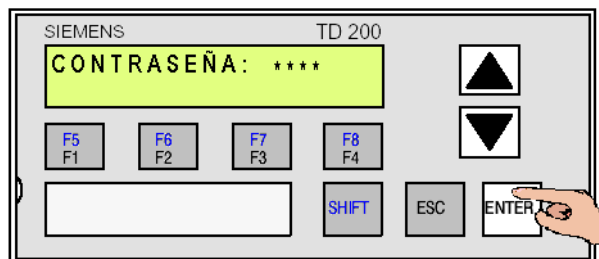
a) Interruptor "Ajuste de Metraje" en posición ON



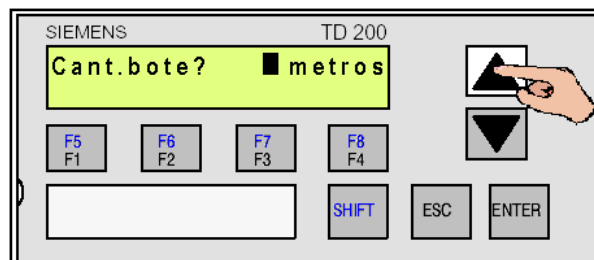
b) Visualización en pantalla, seguido presionar ENTER



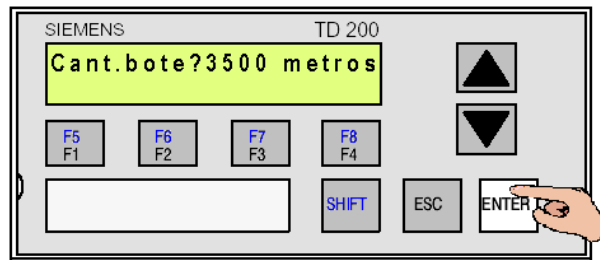
c) Visualización de pedido de contraseña en pantalla, seguido presionar ENTER



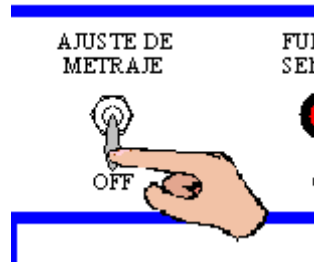
d) Mediante las teclas de flechas ingresar la contraseña y presionar ENTER



e) Por medio de las teclas de flechas fijar el valor deseado




f) Una vez fijado el valor deseado presionar ENTER

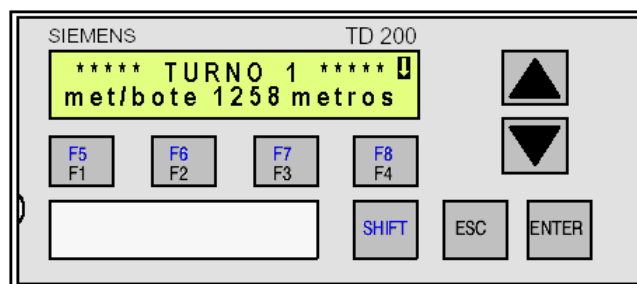


g) Finalmente el interruptor "AJUSTE DE METRAJE" se debe poner en la posición OFF
 Figura 3.28. Procedimiento para ajuste de metraje.

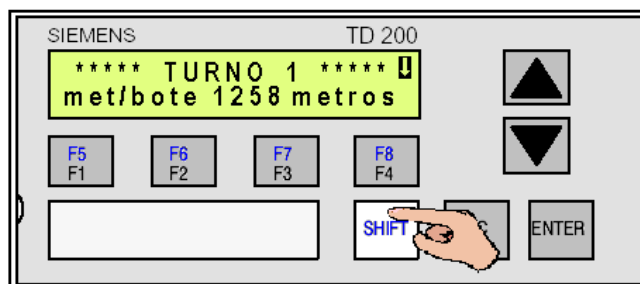
3.4.5 Reiniciación de conteo de bote.

Cuando el contador de bote está funcionando es posible inicializar nuevamente, es decir que empiece su cuenta regresiva desde el valor programado, sin que el contador de la señal de cambio de bote.

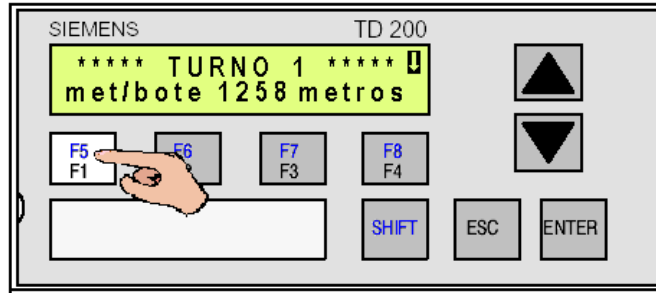
Para realizar esta operación se debe presionar la tecla  seguido de la tecla correspondiente al turno que está operando (F1, F2, F3 ó F4), el procedimiento gráfico se lo muestra en la figura 3.29.



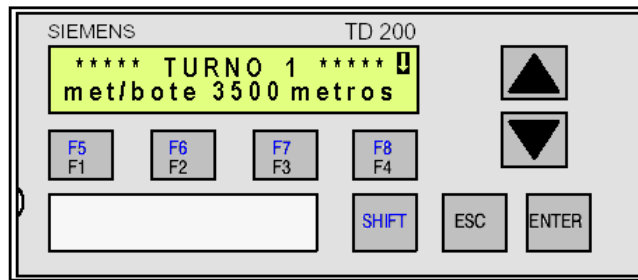
a) Pantalla en normal operación



b) Para inicializar el contador presione "SHIFT"



c) Observe en la pantalla el turno que está operando, en este caso Turno 1, presione F1



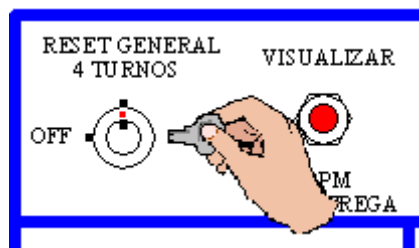
d) Se observa que el contador se inicializó a su valor programado.

Figura 3.29 Pasos para inicialización de contador sin realizar cambio de bote.

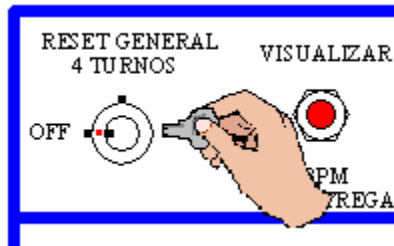
3.4.6 Puesta a cero de los contadores totalizadores.

Para borrar todos los datos totalizados de los conteos, es decir iniciar desde cero la medida de producción total, es necesario activar el interruptor de llave a la posición “RESET GENERAL 4 TURNOS” y posteriormente a la posición “OFF”, así se indica en la figura 3.30, una vez realizada esta acción los valores totalizados de los contadores de los turnos 1, 2, 3 y 4 se ponen en cero, esta opción se la realizará para tener una apreciación de la producción obtenida en un turno, en un día, en una semana, etc.

La llave del interruptor la debe tener solo el personal autorizado.



a) Para poner a cero el totalizado de los contadores el interruptor debe ponerse en la posición “RESET GENERAL 4 TURNOS”



b) Para continuar con el conteo totalizado el interruptor se debe poner en la posición OFF.

Figura 3.30 Procedimiento para poner a cero el totalizado de los contadores.

3.4.7 Estado de la fuente.

El estado de la fuente de alimentación del sistema contador se puede visualizar mediante el diodo LED rojo que se encuentra en el panel de control (Ver figura 3.31), cuando el indicador está encendido nos indica que la energía está en perfecto estado, caso contrario se procederá a revisar el fusible ubicado en la tarjeta electrónica en la parte posterior del contador.

Si el fusible se encuentra en buen estado, se procederá a revisar el suministro de energía de 110Vac en los terminales 13 y 16 de la tarjeta electrónica.

Si el suministro de energía es correcto se procederá a revisar el correcto funcionamiento del PLC.



a) Fuente en mal estado



b) Operación Normal.

Figura 3.31 Diodo indicador de estado de fuente de alimentación.

CAPITULO IV

VENTAJAS Y DESVENTAJAS DEL NUEVO SISTEMA

4.1 Ventajas.

Dentro de las ventajas que se tiene con el nuevo sistema de conteo podemos citar las siguientes:

- El nuevo sistema está basado en un PLC con amplia funcionalidad, el nuevo contador se le puede agregar operaciones o modos de control para un mejor desempeño y optimización de operación de las máquinas.
- El sistema nuevo puede enlazarse con otros sistemas de monitoreo, sistemas SCADA, redes, lo que facilita la operación, monitoreo y control.
- Monitoreo, edición del programa cargado en el PLC.
- Los costos de construcción del nuevo sistema son mucho menores que un contador original EZA-1.
- El nuevo contador es de fácil manejo.
- Dispone de partes reemplazables en caso de daño en alguna de ellas.
- No se ha realizado ninguna adaptación a las máquinas para que funcione el nuevo sistema.
- Mejor resolución de mensajes, ya que se indica la variable que se visualiza, la cantidad y la unidad de medida.

4.2 Desventajas.

- El espacio físico es más grande que el sistema original, debido al panel de visualización TD-200.
- La programación de parámetros se la realiza desde el computador.
- Los mensajes desplegados el en TD-200 solo se visualiza de uno en uno.
- La programación del metraje de bote es un poco lento, ya que se realiza por medio de las teclas y el valor va incrementándose o decrementándose en pasos de 1, luego en pasos de 10 y por último en pasos 100.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES.

5.1 CONCLUSIONES.

- El nuevo contador cumple con los requerimientos de la empresa Textiles Río Blanco al igual que lo hace el contador EZA-1.
- La empresa Textiles Río Blanco dispone de equipos de última tecnología donde la intervención humana es mínima.
- El proceso de cardado es uno de los que necesita mayor control de parámetros textiles.
- Se ha logrado construir un contador en base a un PLC SIEMENS Simatic S7-200
- Los costos para la realización de este proyecto son mucho más bajos de lo que implicaría una importación de contador EZA-1.
- Las operaciones que ofrece el PLC Simatic S7-200, son muy amplias y se puede dar usos mucho más complejos que el proyecto realizado.
- No tiene problemas de operabilidad en las distintas máquinas que realizan el proceso de producción.
- El nuevo contador no ha dañado la estética de las máquinas, es decir no hay ninguna modificación en el espacio físico asignado para el contador EZA-1, en el mismo espacio puede encajar el nuevo contador o el contador EZA-1 sin ningún problema.
- El nuevo sistema realizado, en hardware funciona de la misma manera que el contador EZA-1, es decir que con este contador no se realizó ninguna adaptación o supresión de elementos de la máquina.

- Con la ayuda del computador se puede realizar un diagnóstico de funcionalidad del programa, revisar parámetros, mientras el PLC está operando en forma normal.
- La corrección de fallas o defectos del programa se lo puede corregir, editar o agregar con la ayuda de un computador, en el contador EZA-1 un error de programa dejaría lo fuera de servicio.
- Todas las máquinas que se utilizan en los diferentes procesos de producción tiene en común utilizar el mismo sensor de proximidad, lo que facilitó el desarrollo de este proyecto.
- El sistema desarrollado es de fácil operación, ya que el manejo del TD-200 no es complejo.
- Por la discontinuidad de elementos electrónicos, varios equipos se dan de baja ya que no es posible su reparación por falta de repuestos.
- La empresa Textiles Río Blanco es partícipe del desarrollo de nuevos equipos conforme va evolucionando la tecnología
- El nuevo equipo es capaz de realizar funciones adicionales dependiendo de alguna necesidad que se presente en el funcionamiento de las mismas.
- La frecuencia de operación de los contadores normales se satura a los 80Hz, para lo cual se utiliza contadores rápidos (alta velocidad) que soportan frecuencias de 30Khz.
- La visualización de textos es mucho mejor que la del contador EZA-1.
- El nuevo sistema no se ve muy afectado por la humedad en la que trabaja la planta que va desde el 45% al 52% de humedad relativa.
- El nuevo contador es una solución práctica que se puede implementar en todas las áreas de producción.

5.2 RECOMENDACIONES.

- Se recomienda tomar muy en cuenta la estética de un equipo a realizarse.
- Aprovechar de la mejor manera todas las entradas o salidas del PLC.

- Realizar con cuidado las tarjetas electrónicas, revisar minuciosamente cada una de las pistas antes de proceder al montaje de los elementos electrónicos.
- Realizar una codificación de cables para no producir confusiones y posteriormente conexiones mal realizadas que implicaría el daño de uno o varios equipos.
- Dar mantenimiento periódico a los componentes electrónicos para un mejor funcionamiento, ya que el equipo está expuesto a un ambiente de polvo y pelusa.
- Tener una idea clara del funcionamiento de maquinaria y equipos para dar un criterio de diseño y construcción de equipos que sustituyan a otros.
- Se debe tomar en cuenta las normas de seguridad y avisos de advertencia para evitar descargas o accidentes de trabajo.
- Capacitar al personal que maneja directamente el equipo, para que se familiarice y realice la operación en forma correcta.
- Es recomendable aislar la parte de control de la parte de potencia.
- Marcar las zonas por las cuales se energiza con AC o DC, e indicar la cantidad que debe tener el suministro.
- Tener el manual de operación hasta ambientarse con el nuevo equipo.
- Realizar las funciones y operaciones necesarias optimizando el programa que las va a realizar.

BIBLIOGRAFIA

- **SIEMENS**, “SIMATIC Sistema de automatización S7-200”
Manual de Sistema Edición 03/2000
- **RIETER**, Manual de operación de Carda C4.
- **RIETER**, Manual de operación de Peinadora E75.
- **RIETER**, Manual de operación de Hila G51.
- <http://www.rieter.com>
- <http://www.siemens.com/S7-200>

GLOSARIO.

AEROFEED.- Aspira las fibras de la última máquina las transporta a una tubería de alimentación.

BIT.- Es la unidad más pequeña de información y la unidad base en comunicaciones.

Byte.- Conjunto de bits continuos mínimos que hacen posible un direccionamiento de información en un sistema computarizado. Está formado por 8 bits.

Cilindro Calandrador.- Conjunto de dos cilindros utilizados para trasportar material textil.

CPU.- (Center Process Unit). Unidad central de proceso.

Hardware.- Conjunto de componentes electrónicos, eléctricos y mecánicos que soportan la información y realizan operaciones para las cuales fueron diseñados.

LIKERIN.- Es un elemento de forma cilíndrica cuya superficie se encuentra recubierta de púas de acero en forma de sierras llamadas guarniciones fijas

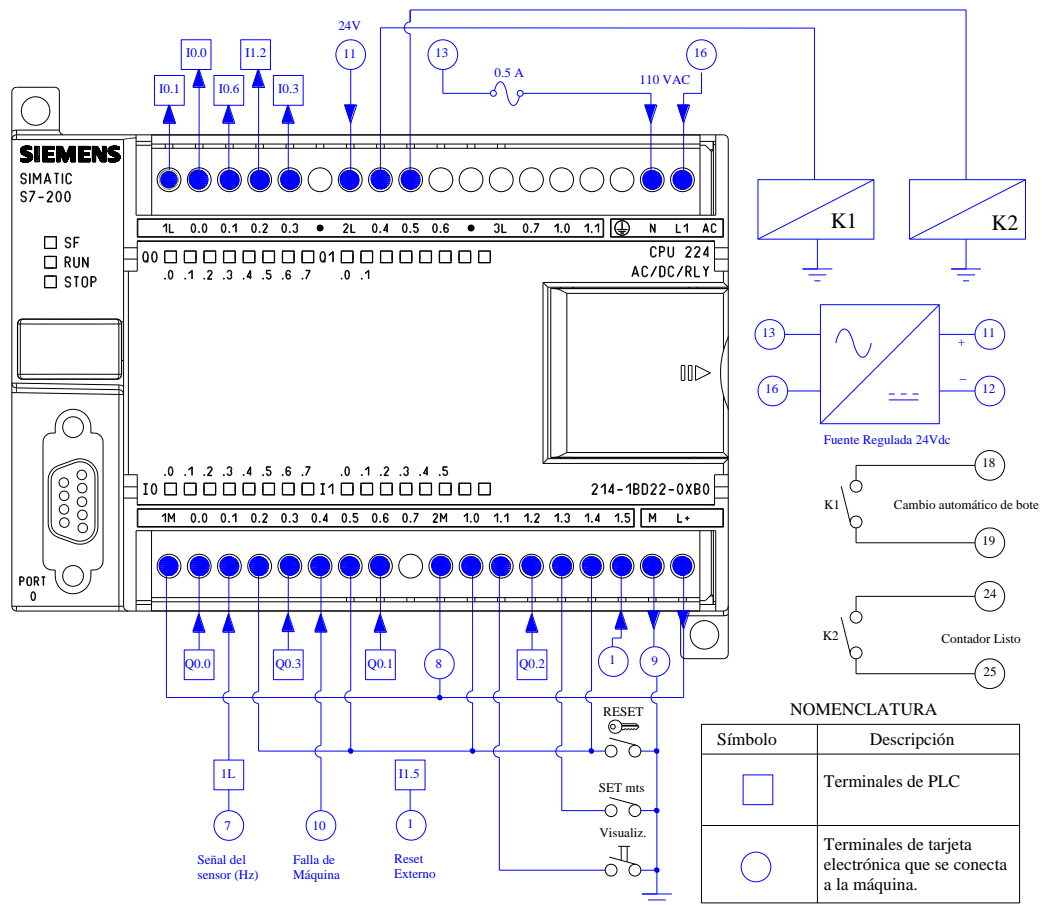
PLC.- Controlador Lógico Programable, sistema lógico que combina hardware y software, de modo que sus instrucciones codificadas puedan modificarse (reprogramarse); proporciona automatización “flexible”.

PPI.- (Point to point interface). Protocolo de comunicación utilizado por el PLC Siemens.

SCADA.- (Supervisory Control And Data Acquisition). Control por Supervisión y Adquisición de Datos.

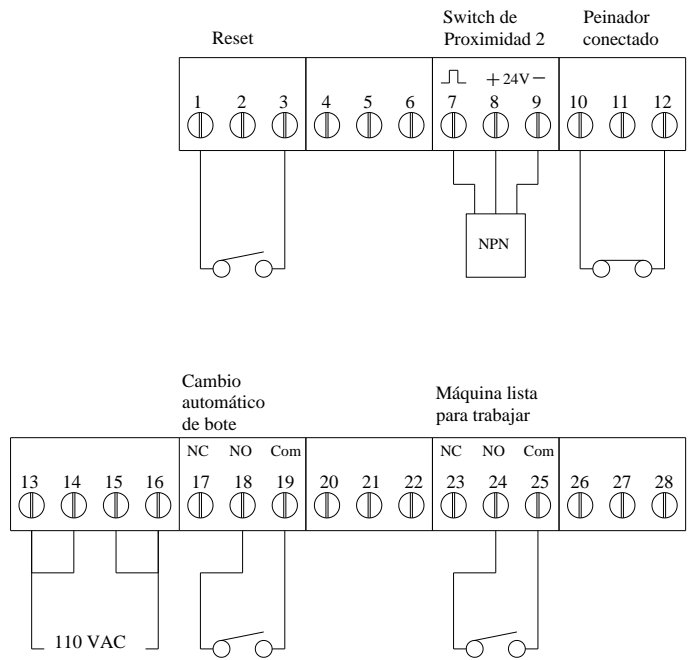
Software.- Conjunto de instrucciones (programas) y datos que, almacenados en la memoria de la máquina, describen el trabajo a realizar.

ANEXO A.



Anexo A. Conexiones entre el PLC y tarjeta electrónica.

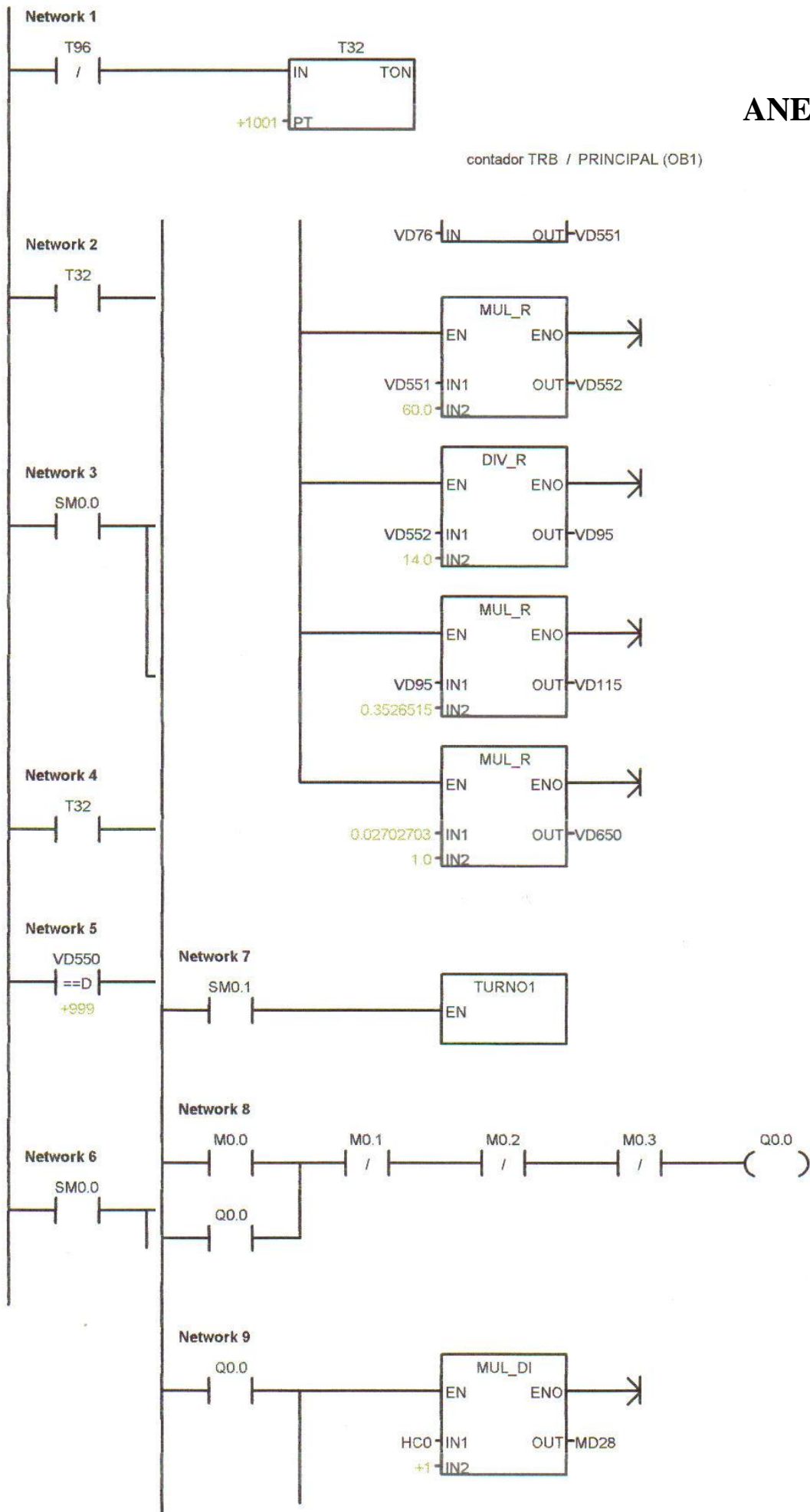
ANEXO B.



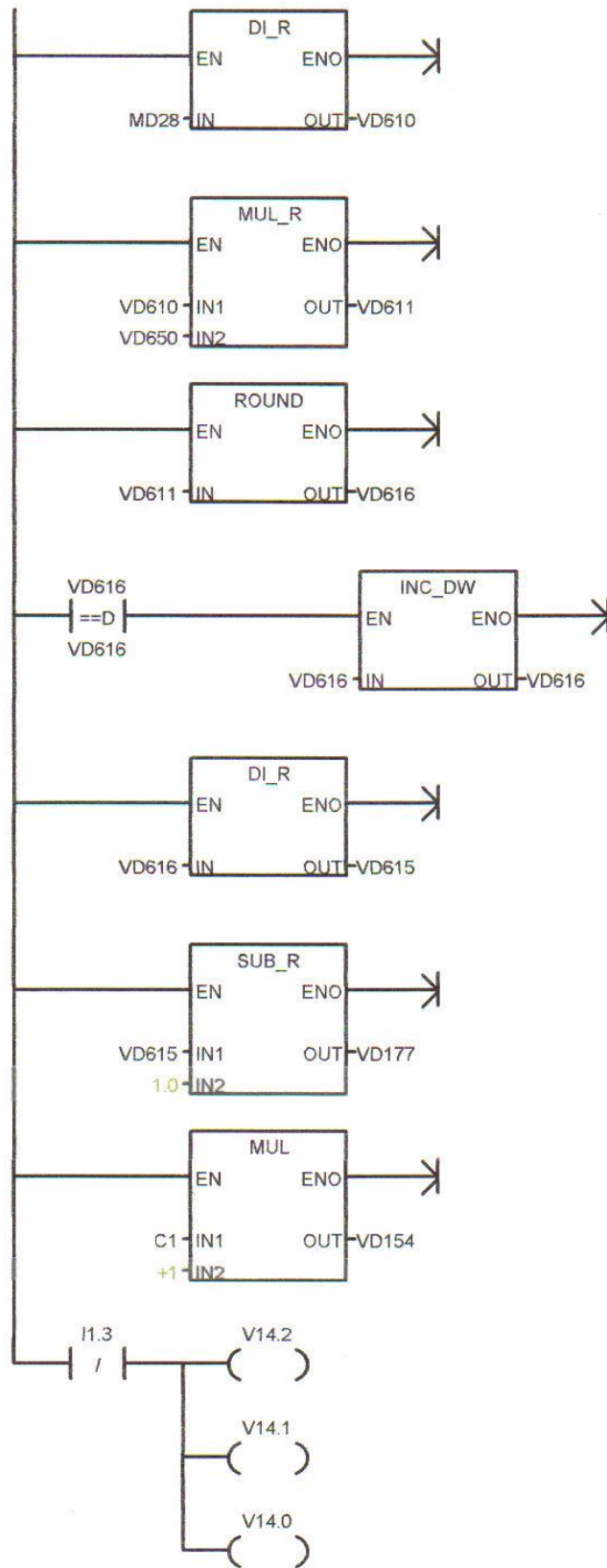
Anexo B. Conexión de terminales de contador EZA-1

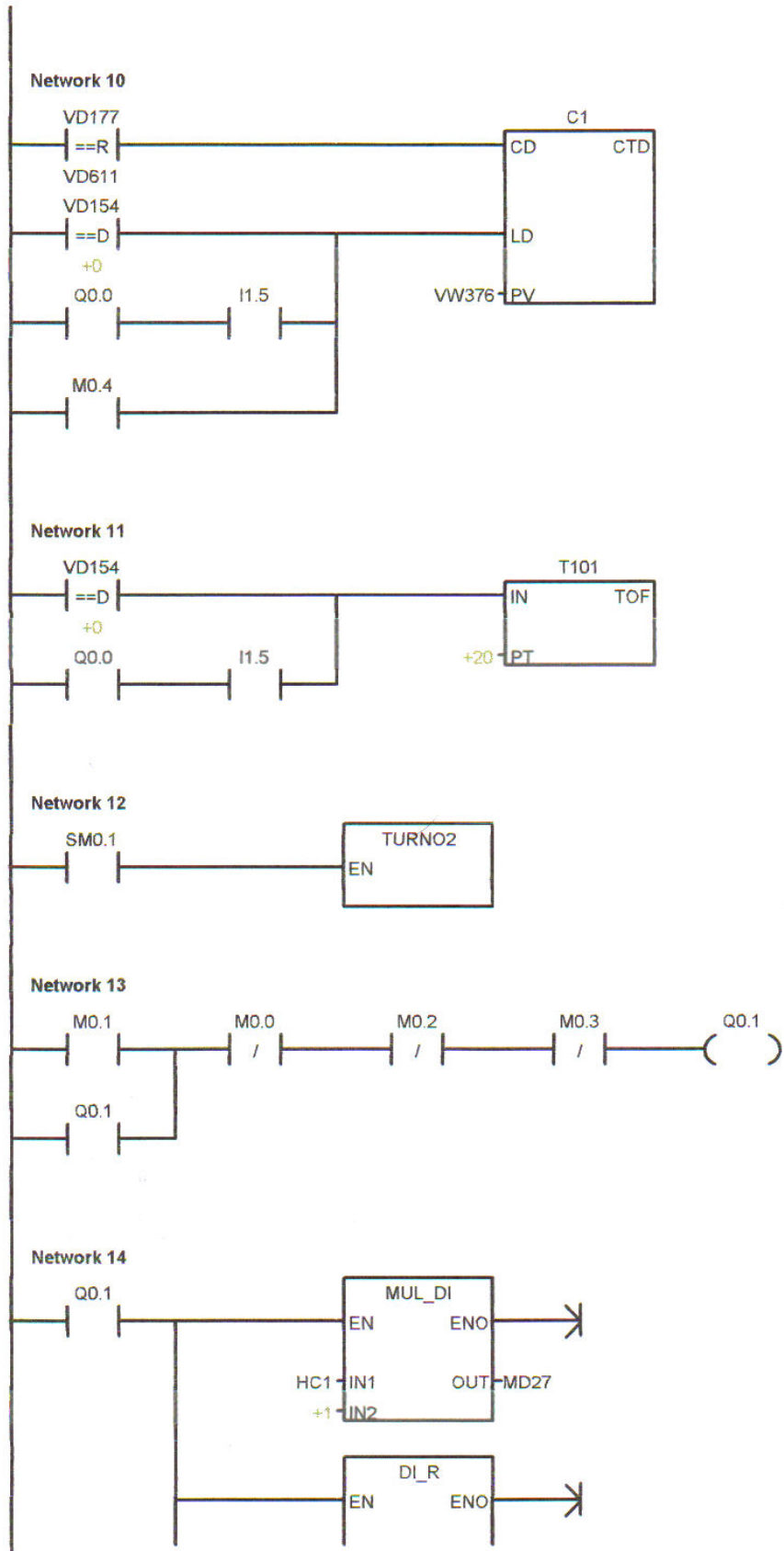
contador TRB / PRINCIPAL (OB1)

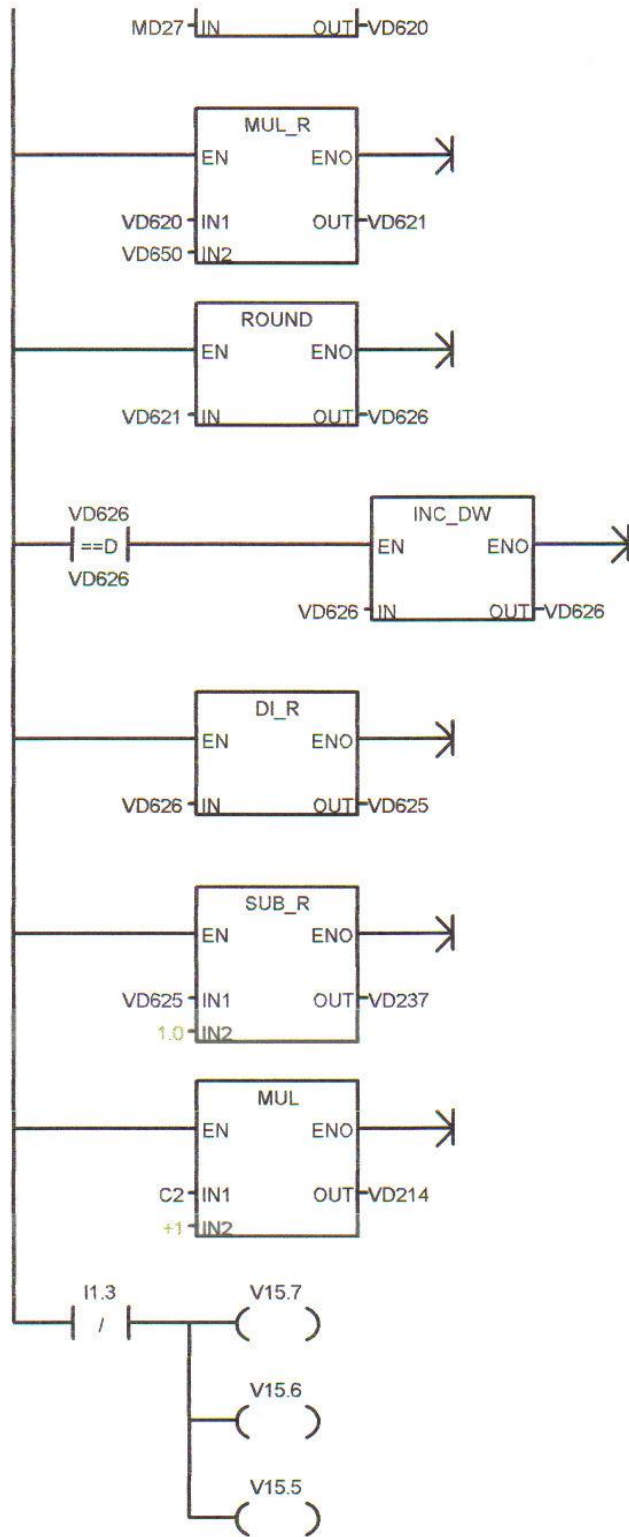
ANEXO C.



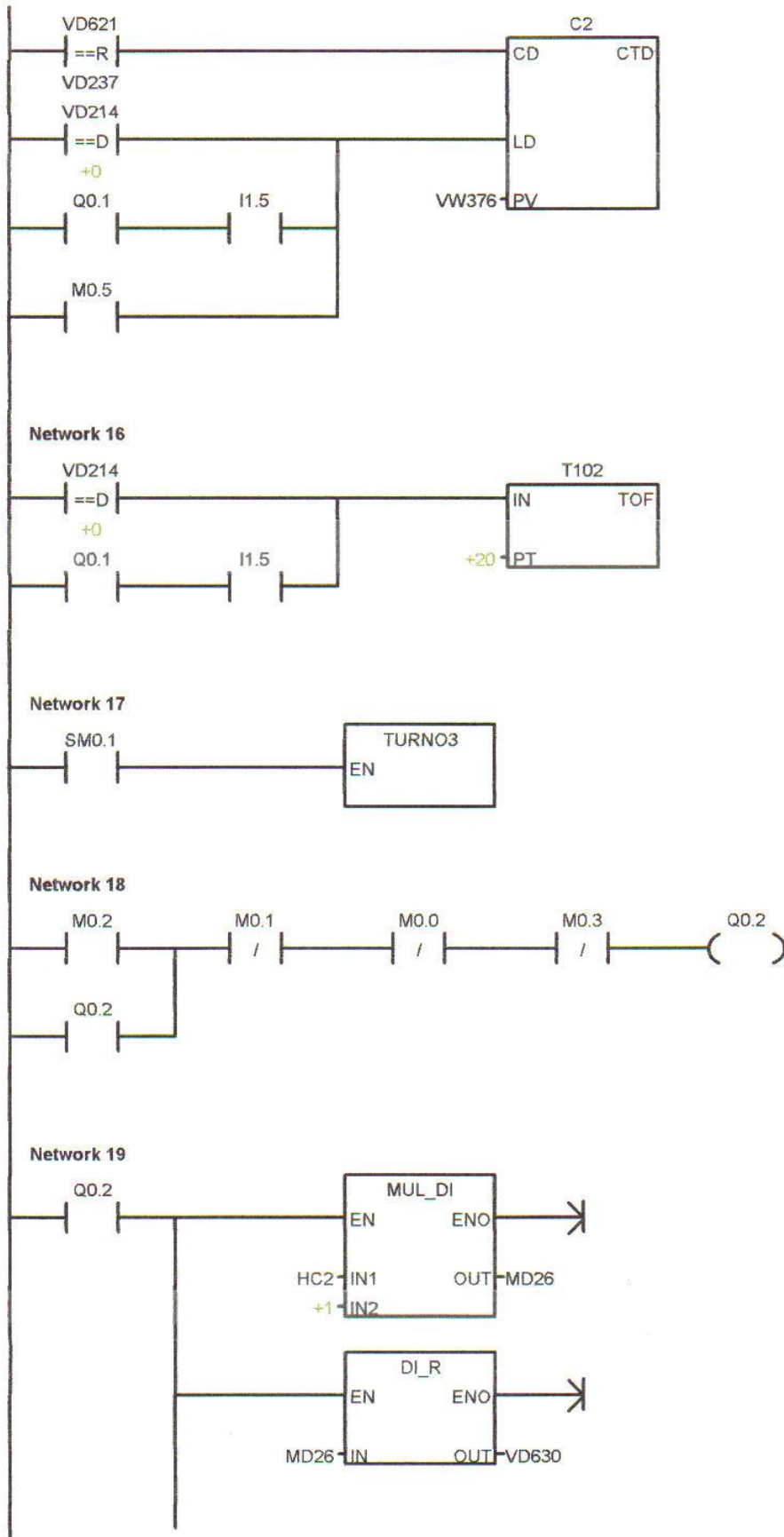
contador TRB / PRINCIPAL (OB1)



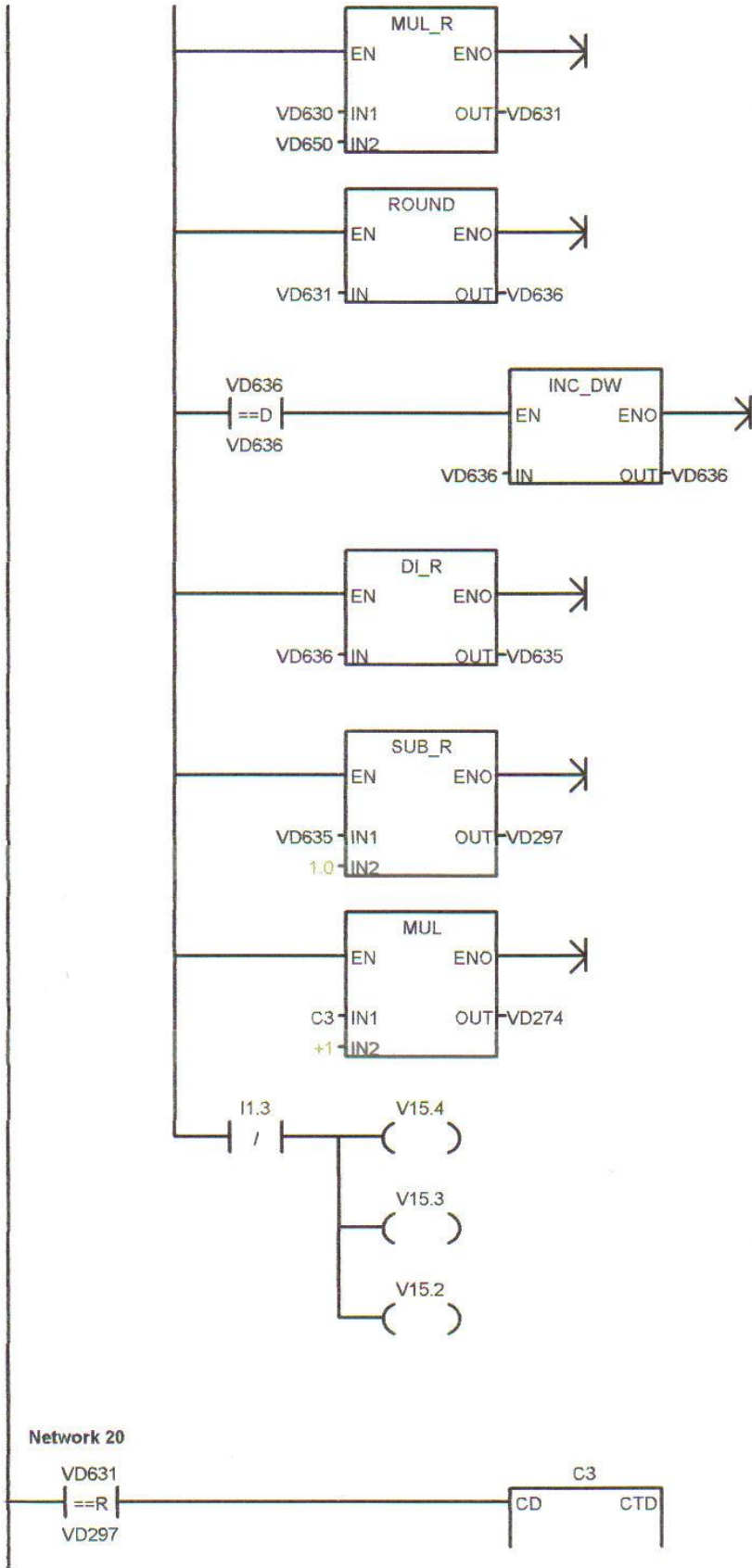




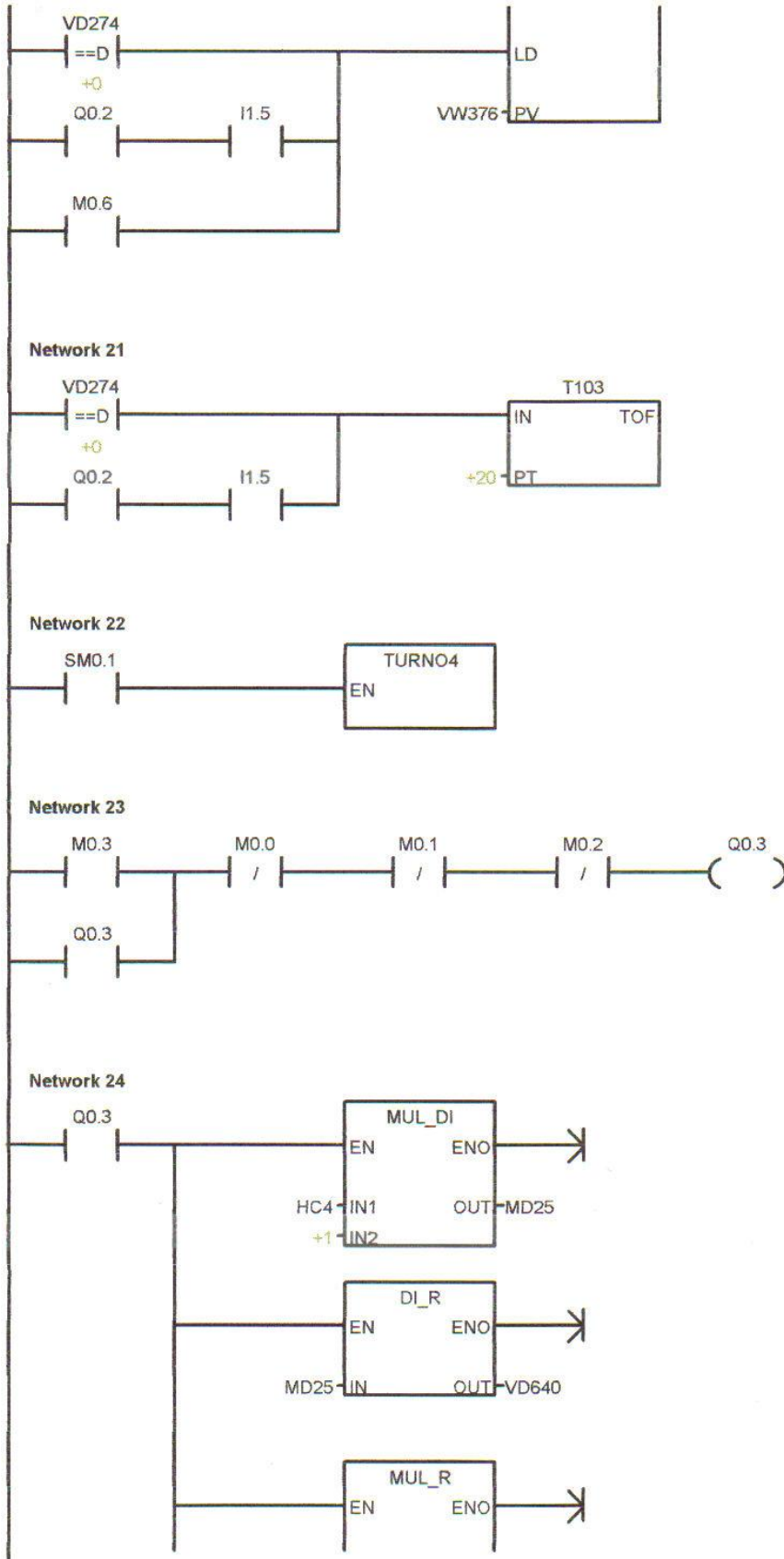
Network 15



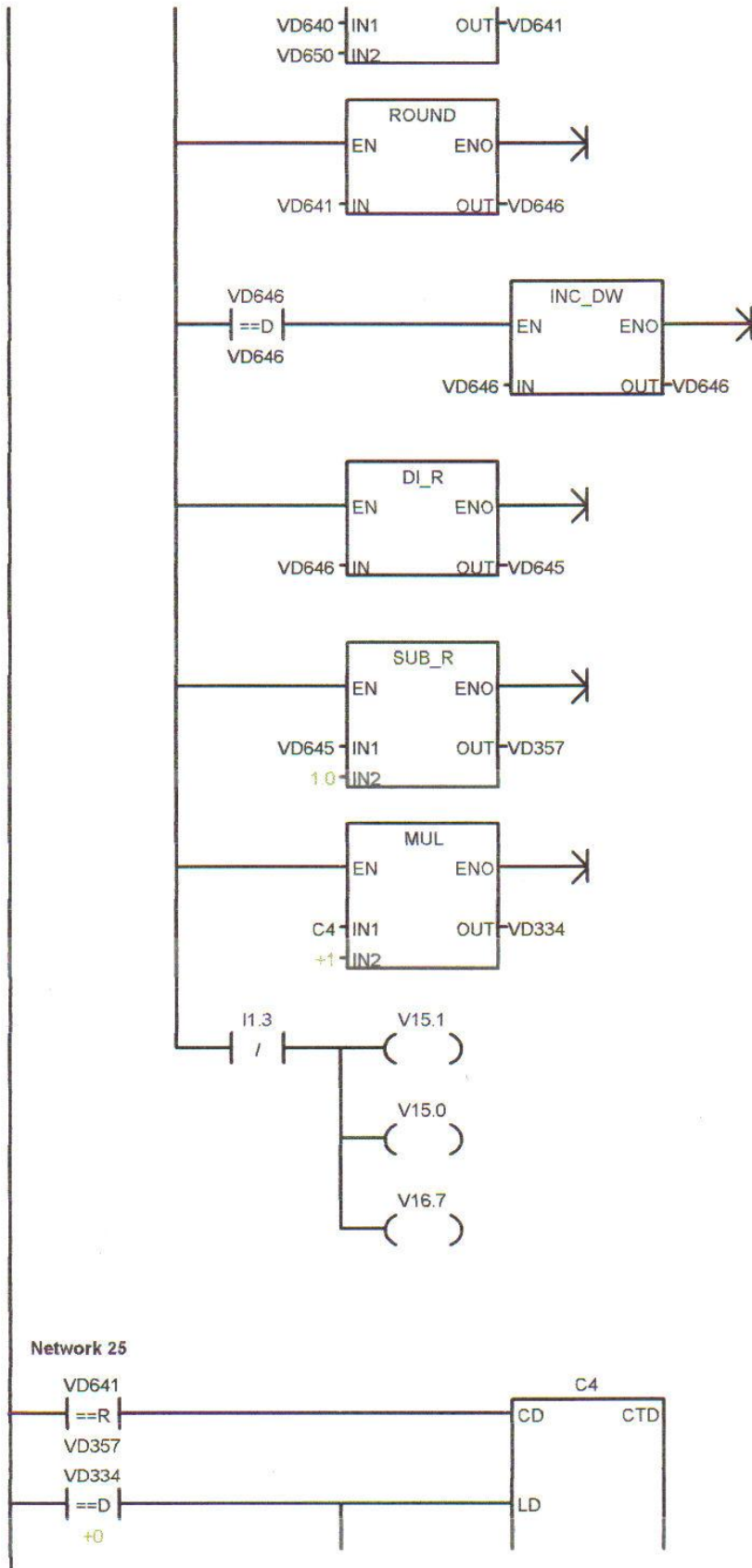
contador TRB / PRINCIPAL (OB1)



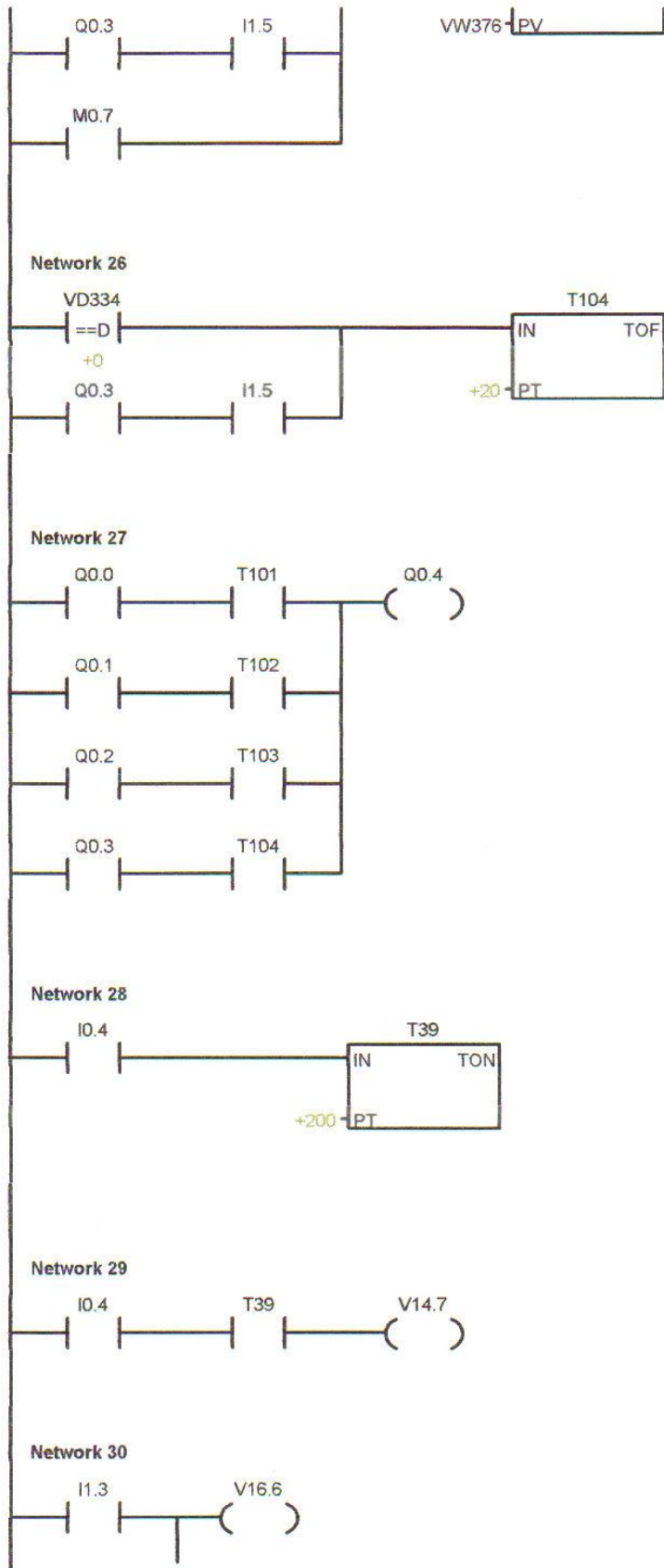
contador TRB / PRINCIPAL (OB1)



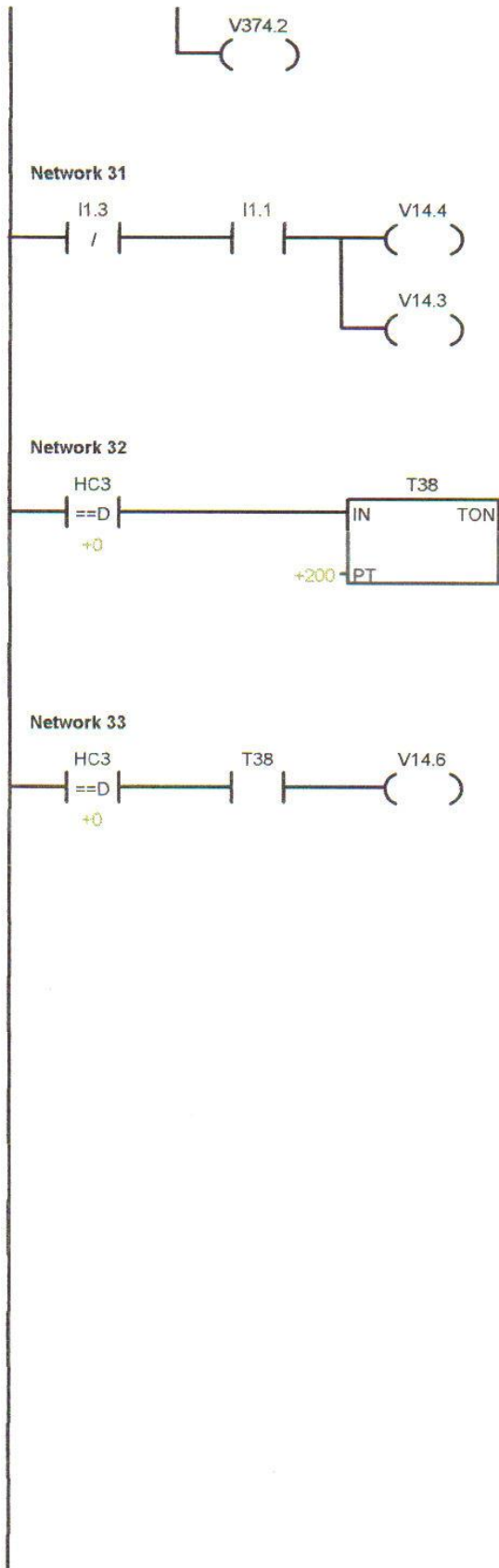
contador TRB / PRINCIPAL (OB1)



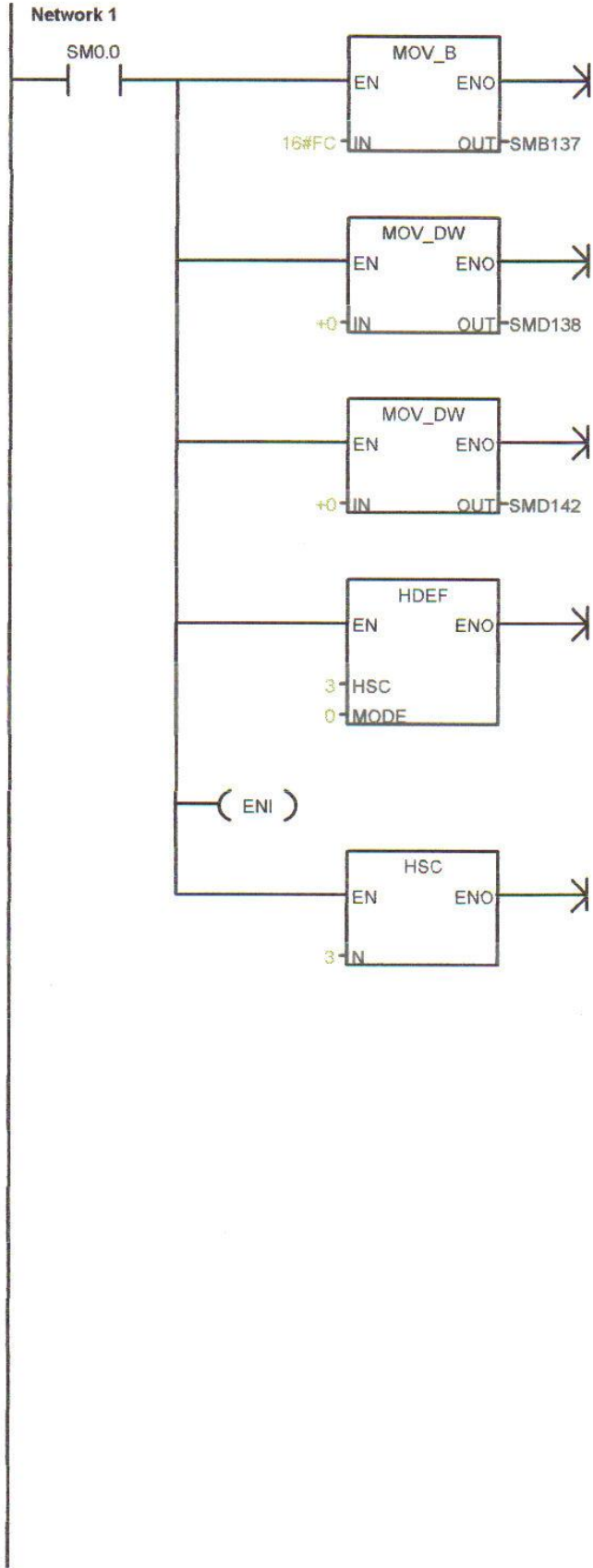
contador TRB / PRINCIPAL (OB1)



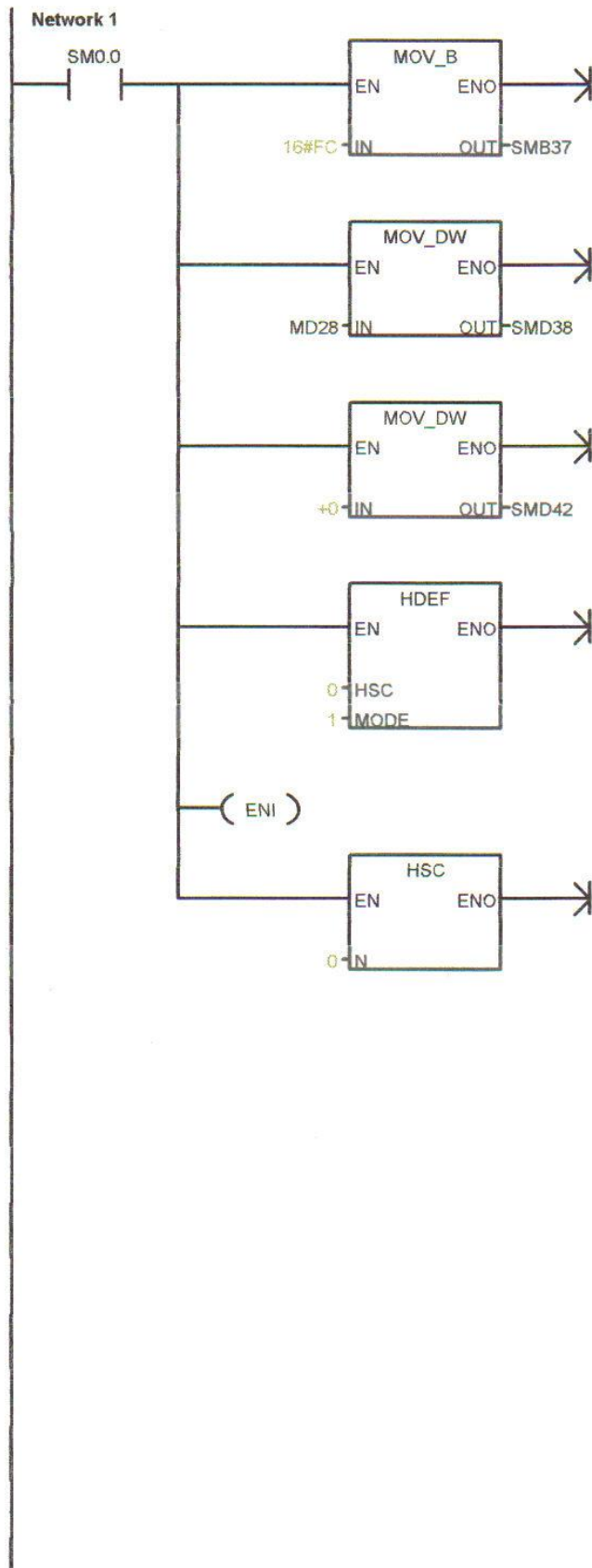
contador TRB / PRINCIPAL (OB1)



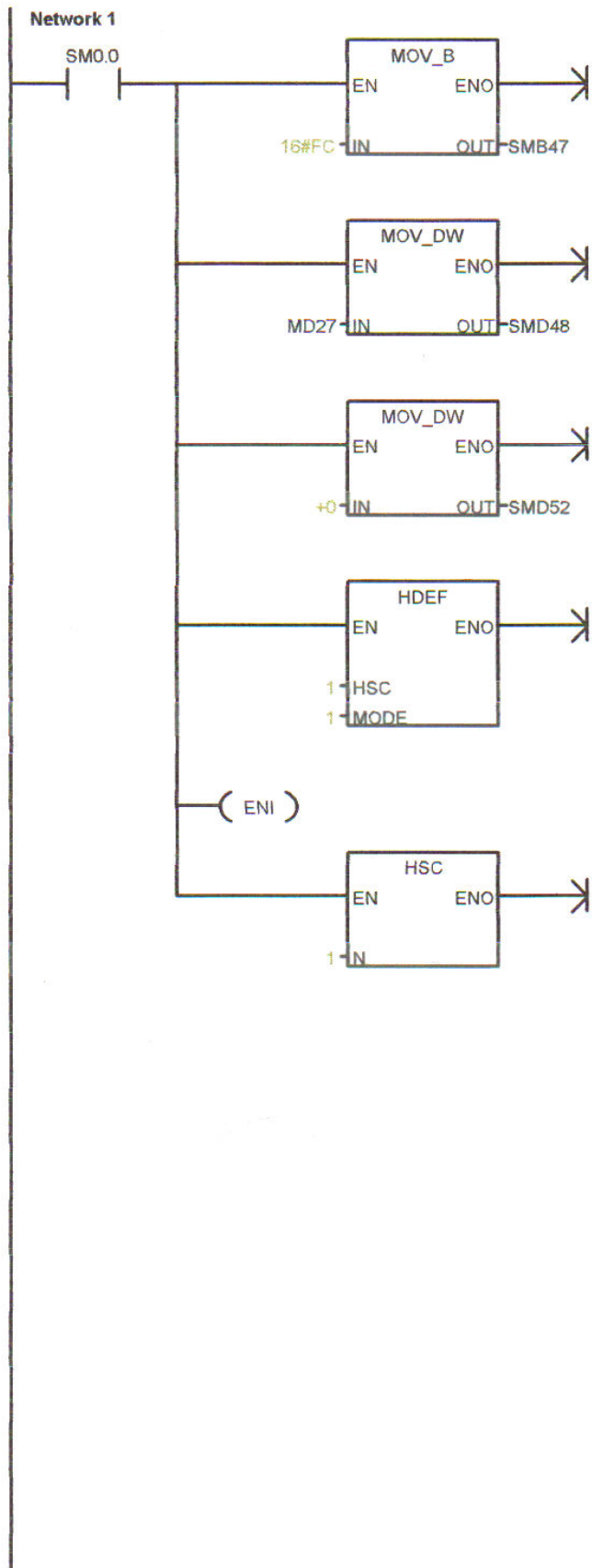
contador TRB / TACOMETRO (SBR0)



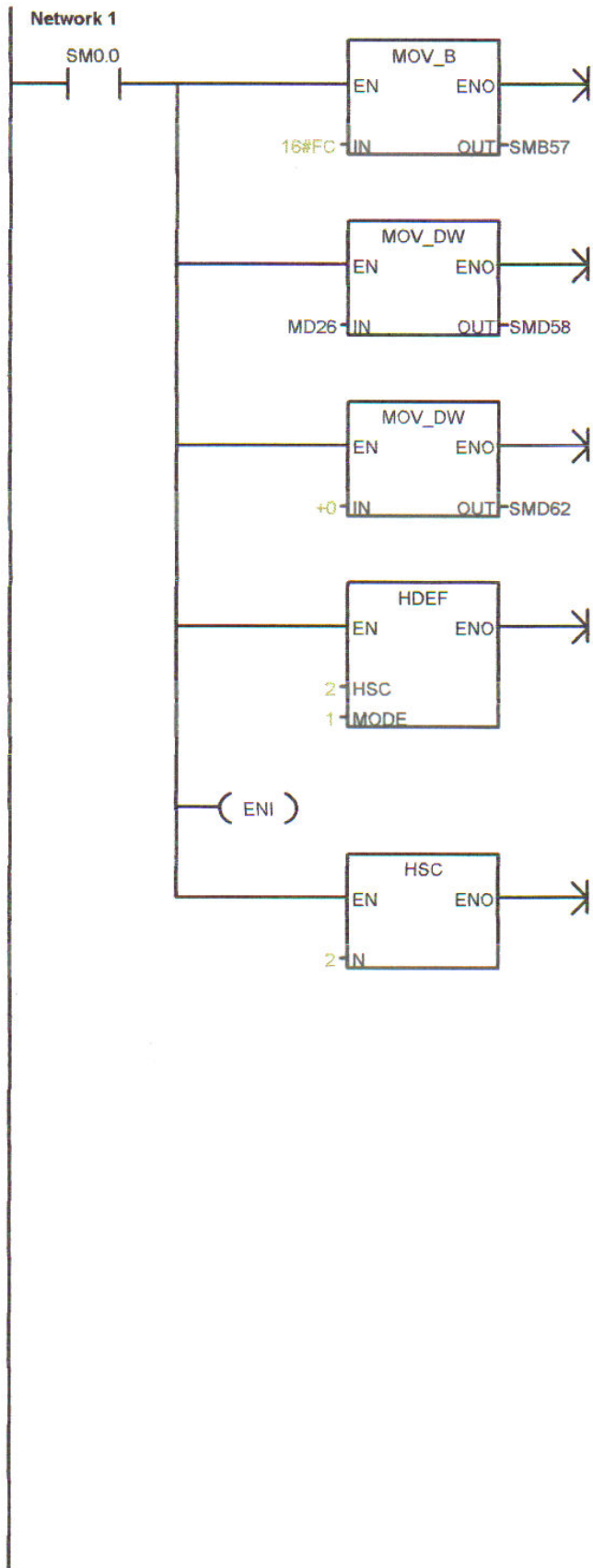
contador TRB / TURNO1 (SBR1)



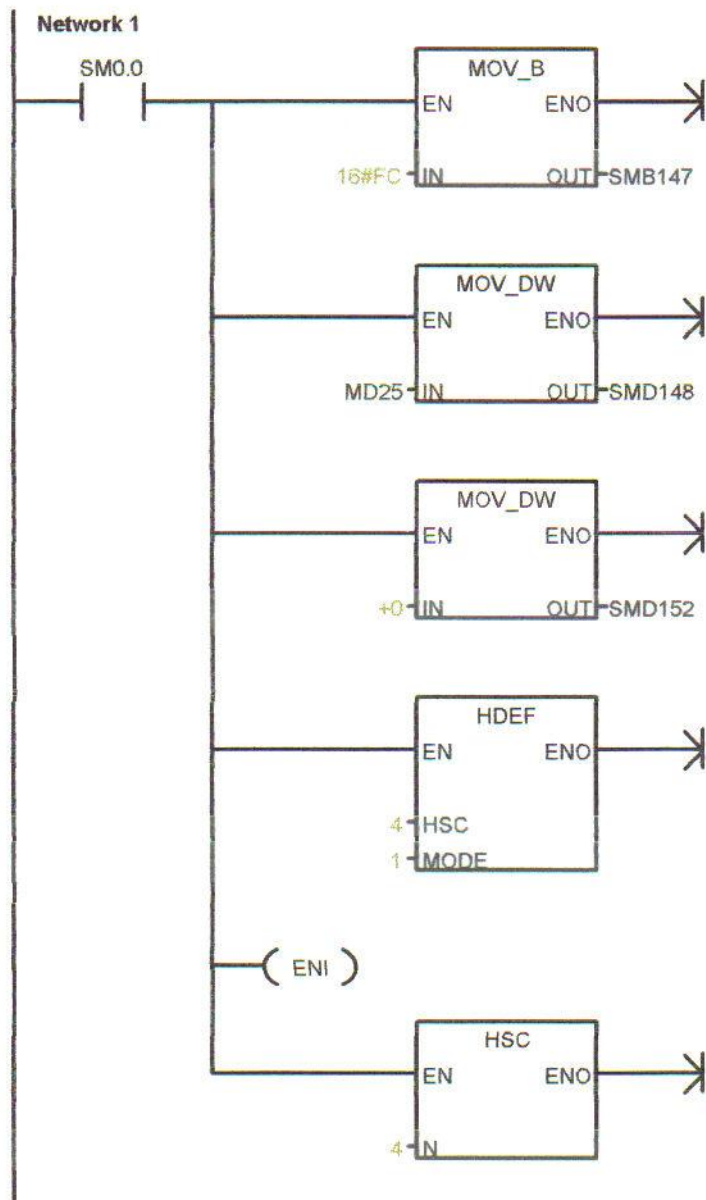
contador TRB / TURNO2 (SBR2)



contador TRB / TURNO3 (SBR3)



contador TRB / TURNO4 (SBR4)



Anexo C. Diagramas del programa realizado

LATACUNGA, OCTUBRE DEL 2006

ELABORADO POR:

JAVIER MOLINA

**ING. ARMANDO ÁLVAREZ
COORDINADOR DE CARRERA**

**DR. EDUARDO VÁSQUEZ ALCÁZAR
SECRETARIO ACADÉMICO**