

UNIVERSIDAD AUTONOMA DE NUEVO LEON  
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA



AUTOMATIZACION DE TOLVAS  
Y GRAFCET

TESINA PROFESIONAL  
QUE EN OPCION AL TITULO DE  
Ingeniero en Control y Computación  
PRESENTA:  
GASPAR ALFREDO SOSA ESQUIVEL

CD. UNIVERSITARIA

ENERO DE 1996

T

TJ223

.P76

S6

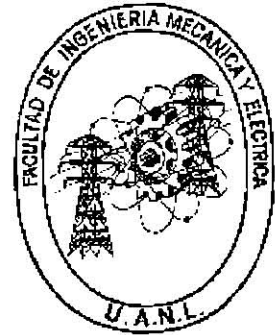
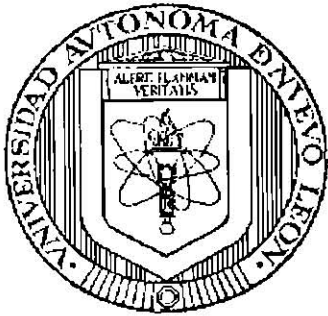
C.1



1080064425

**UNIVERSIDAD AUTONOMA DE NUEVO LEON**

**FACULTAD DE INGENIERIA MECANICA Y  
ELECTRICA**



***AUTOMATIZACION DE TOLVAS Y GRAFCET***

**TESINA PROFESIONAL  
QUE EN OPCION AL TITULO DE:**

***INGENIERO EN CONTROL Y COMPUTACION***

**PRESENTA:**

***GASPAR ALFREDO SOSA ESQUIVEL***

*ENERO DE 1996*

*CD. UNIVERSITARIA.*

T  
TJ223  
.P7  
S6



*Alfonsi*

BUR U  
UAWL  
FONDO  
TEMS LICENCIATURA

*A todos aquellos que me han brindado  
su apoyo para llegar al lugar en  
donde hoy estoy.*

*A mis padres.*

## INDICE

<b>AUTOMATIZACION DE UNA TOLVA MOVIL .....</b>	<b>2</b>
<b>Descripción de la Planta .....</b>	<b>2</b>
<b>La Necesidad .....</b>	<b>2</b>
<b>La Propuesta .....</b>	<b>2</b>
<b>Descripción del Sistema de Control Lógico .....</b>	<b>2</b>
<b>Diagrama de la Planta .....</b>	<b>4</b>
<b>Descripción del Tablero .....</b>	<b>5</b>
<b>Funcionamiento. ....</b>	<b>6</b>
<b>LOS PLC'S Y EL CONTROL LOGICO SECUENCIAL.....</b>	<b>7</b>
<b>Introducción al Control Lógico.....</b>	<b>7</b>
<b>Historia y Desarrollo de los Plc's .....</b>	<b>8</b>
<b>Definición del Controlador Lógico Programable .....</b>	<b>8</b>
<b>GRAFCET.....</b>	<b>10</b>
<b>Introducción .....</b>	<b>10</b>
<b>Nociones Básicas.....</b>	<b>11</b>
<b>Etapas y Transiciones.....</b>	<b>11</b>
<b>Ejemplos de Implementación de Grafcet .....</b>	<b>13</b>
<b>Descripción .....</b>	<b>13</b>
<b>Ejemplo 1: Automatización de una Máquina Barrenadora .....</b>	<b>14</b>
<b>Ejemplo 2: Control de un Elevador de Dos Pisos.....</b>	<b>19</b>
<b>Ejemplo 3: Puesto Automático de Alimentación de Piezas .....</b>	<b>22</b>
<b>Importancia Actual de Grafcet .....</b>	<b>27</b>
<b>APENDICE .....</b>	<b>33</b>
<b>Diagrama Escalera del Proyecto "Automatización de una Tolva Móvil" .....</b>	<b>33</b>
<b>Codificación del Proyecto .....</b>	<b>42</b>
<b>Bibliografía.....</b>	<b>46</b>

## **PROYECTO**

### **AUTOMATIZACION DE UNA TOLVA MOVIL**

#### **DESCRIPCIÓN DE LA PLANTA**

El proyecto esta diseñado para trabajar en una planta de fundición de acero, enfocándose principalmente a la parte donde se realiza la recolección de los materiales combustibles. La planta consta de cuatro tolvas donde se encuentran contenidos los materiales a usarse en la combustión, como son: carbón mineral, carbón coke, piedra caliza, carbón vegetal, etc. También se tiene una tolva móvil en la cual se van depositando cada uno de los materiales antes mencionados. La tolva móvil se desplaza por debajo de cada una de las tolvas. Al terminar de recolectar el material la tolva móvil procede a descargar su contenido en un molino.

#### **LA NECESIDAD**

La justificación para automatizar la planta se fundamenta en la necesidad de contar con las mezclas necesarias para la combustión en una forma automática, más rápida y segura, también para que, automáticamente, se transportaran dichas mezclas hasta el molino. otra de las necesidades era la de contar con alarmas para estar monitoreando el nivel de las tolvas.

#### **LA PROPUESTA**

Se propuso realizar la automatización a base de PLC's por los siguientes motivos:

- 1.- Mediante un análisis se pudo constatar que las variables eran del tipo encendido apagado (on - off). Por lo tanto se llegó a la conclusión que era necesario utilizar Control Lógico.
- 2.- Se decidió controlar la planta por medio de un PLC, tomando en cuenta el bajo costo de los PLC's comparado con otros dispositivos de control.

#### **DESCRIPCIÓN DEL SISTEMA DE CONTROL LÓGICO**

Se necesitaba que el control fuera capaz de proveer tres tipos de mezclas. La primera mezcla debería ser de la siguiente manera:

Producto A	25 %
Producto B	25 %
Producto C	25 %
Producto D	25 %



La segunda mezcla se muestra a continuación:

Producto A	10 %
Producto B	20 %
Producto C	30 %
Producto D	40 %

Y por ultimo la tercera mezcla es la siguiente:

Producto A	40 %
Producto B	30 %
Producto C	20 %
Producto D	10 %

Haciendo un estudio del proceso se logro hacer que el material se descargara de las tolvas a una razón de 10 Kgs /seg del material. Se necesitaba que la mezcla final tuviera un peso de cien kilogramos.

Usando la razón de descarga del material se pudo establecer una relación entre el peso del material y el tiempo de descarga quedando la siguiente relación :

Para la primer mezcla tenemos

Tolva	Peso Requerido	Tiempo
A	25 Kg	2.5 seg
B	25 Kg	2.5 seg
C	25 kg	2.5 seg
D	25 Kg	2.5 seg

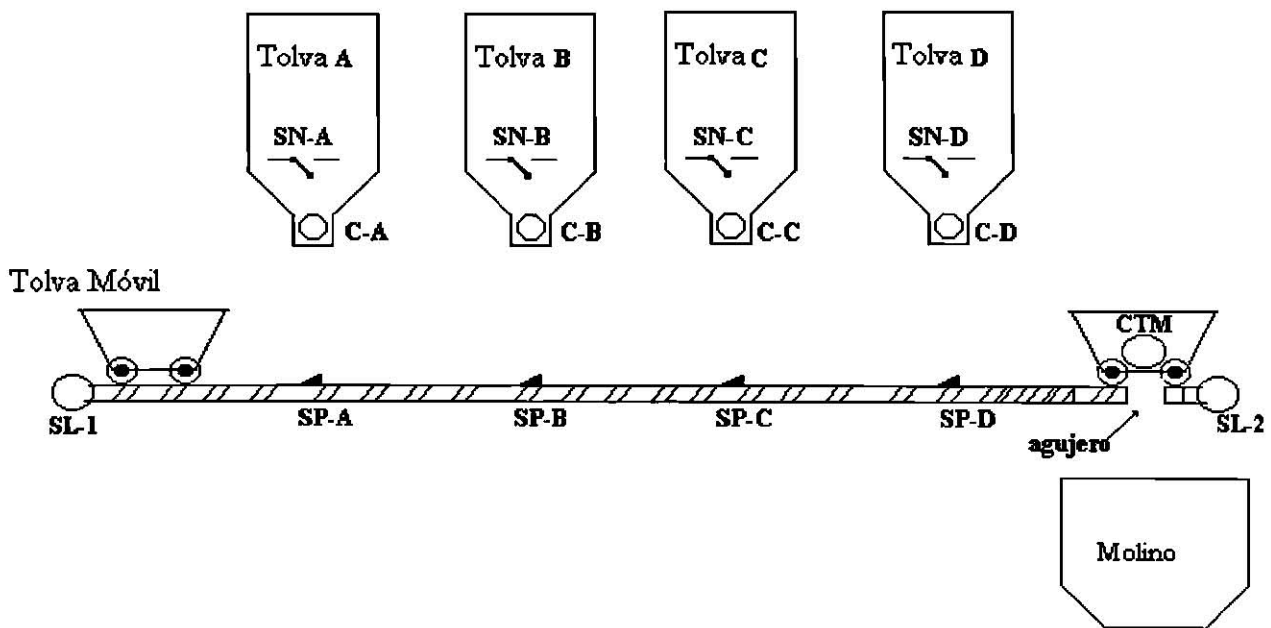
Para la segunda mezcla tenemos :

Tolva	Peso Requerido	Tiempo
A	10 Kg	1 seg
B	20 Kg	2 seg
C	30 Kg	3 seg
D	40 Kg	4 seg

Para la tercera mezcla tenemos :

Tolva	Peso Requerido	Tiempo
A	40 Kg	4 seg
B	30 Kg	3 seg
C	20 Kg	2 seg
D	10 Kg	1 seg

### DIAGRAMA DE LA PLANTA



### Simbología usada en el dibujo

<b>SN-A</b>	Sensor de nivel de la Tolva A
<b>SN-B</b>	Sensor de nivel de la Tolva B
<b>SN-C</b>	Sensor de nivel de la Tolva C
<b>SN-D</b>	Sensor de nivel de la Tolva D
<b>C-A</b>	Compuerta Tolva A
<b>C-B</b>	Compuerta Tolva B
<b>C-C</b>	Compuerta Tolva C
<b>C-D</b>	Compuerta Tolva D
<b>SL-1</b>	Sensor de limite 1
<b>SL-2</b>	Sensor de limite 2
<b>SP-1</b>	Sensor de Posición de la Tolva A
<b>SP-B</b>	Sensor de Posición de la Tolva B
<b>SP-C</b>	Sensor de Posición de la Tolva C
<b>SP-D</b>	Sensor de Posición de la Tolva D
<b>CTM</b>	Compuerta de la Tolva Móvil

## DESCRIPCIÓN DEL TABLERO

<input type="radio"/> ARRANQUE	<input type="radio"/> FORMULA 1	<input checked="" type="radio"/> CORRIENDO	<input checked="" type="radio"/> ADEL.
<input type="radio"/> PAUSA	<input type="radio"/> FORMULA 2	<input checked="" type="radio"/> FIN	<input checked="" type="radio"/> ALTO
<input type="radio"/> ABORTO	<input type="radio"/> FORMULA 3	<input checked="" type="radio"/> ERROR	<input checked="" type="radio"/> REV.

En el tablero es donde se encuentran los controles del sistema y las lámparas indicadoras que son las siguientes :

- 1.- Arranque; Este botón sirve para dar inicio al ciclo de trabajo del sistema.
- 2.- Pausa; Este botón sirve para detener durante un momento nuestro sistema de control.
- 3.- Aborto; Este botón detiene todo el sistema en cualquier punto del proceso, después de usar esta opción el sistema deberá arrancarse desde el principio.
- 4.- Correr; Esta lampara nos indica que el programa esta corriendo.
- 5.- Fin; Esta lampara nos indica que el sistema ha terminado su rutina y esta en espera de arrancar otra vez.
- 6.- Error; Esta lampara nos indica que ha habido un error en el proceso.
- 7.- Reversa; Esta lampara nos indica que la tolva móvil se esta moviendo hacia su posición de inicio.
- 8.- Alto; Esta lampara nos indica que la tolva móvil se encuentra en estado de reposo.
- 9.- Adelante; Esta lampara nos indica que la tolva móvil se esta desplazando hacia adelante.
- 10.- Fórmula 1; Por medio de este botón seleccionamos la mezcla # 1.
- 11.- Fórmula 2; Por medio de este botón seleccionamos la mezcla # 2.
- 12.- Fórmula 3; Por medio de este botón seleccionamos la mezcla # 3.

## **FUNCIONAMIENTO.**

Al encender el PLC y ponerlo en modo de correr (Run), lo primero que hace el programa es revisar que las tolvas estén a un nivel suficiente de no ser así el programa marcará error y no permite continuar con el proceso una vez revisado esto el programa se queda en espera de que se seleccione una de las fórmulas en dado caso de que se llegue a seleccionar dos al mismo tiempo por accidente el programa marcará error.

Una vez que se haya seleccionado una de las tres fórmulas el siguiente paso es presionar el botón de arranque. Al comenzar la rutina la tolva móvil debe de estar en su posición inicial, de no ser así automáticamente el controlador manda la señal a la tolva móvil para que se active la reversa hasta llegar a dicha posición.

Ya estando en posición de inicio la tolva móvil se dispone a avanzar hacia adelante, al llegar a la tolva "A" el sensor de proximidad (SP-A) detecta la tolva móvil y manda parar a esta e inmediatamente la compuerta de la tolva A (C-A) comienza a descargar durante algunos segundos lo que llamamos "Tiempo de descarga"; una vez concluida esta acción se cierra la compuerta y la tolva móvil continúa su camino hacia la siguiente tolva.

Este procedimiento se sigue repitiendo en las siguientes tres tolvas B, C y D; una vez que las tolvas han vaciado su contenido la tolva móvil se dispone a viajar hasta el otro extremo del carril donde se encuentra el molino al llegar a donde esta este la tolva móvil es detectada por el sensor de proximidad (SP-2) se detiene y se acciona una compuerta que esta ubicada en la parte inferior de la tolva móvil llamada compuerta de la tolva móvil (CTM) depositando su contenido en el molino para que este haga una mezcla homogénea.

Una vez que concluye esto la tolva móvil regresa al inicio del recorrido en espera de otra rutina.

Para la implementación de este proyecto se usó el PLC Micro - 1 de IDEC Corp. En el apéndice se localiza la codificación y el diagrama escalera correspondientes a este proyecto

## **LOS PLC'S Y EL CONTROL LOGICO SECUENCIAL**

### **INTRODUCCIÓN AL CONTROL LÓGICO**

Actualmente existen muchas aplicaciones de control en las cuales es necesario manipular variables de salida del tipo prendido - apagado (on - off), de acuerdo a condiciones del mismo tipo, en los dispositivos que constituyen las variables de entrada. A este tipo de control se le conoce como control lógico y se utiliza en un sin número de máquinas automáticas y en la automatización de procesos en los que sus variables son tales que el proceso sigue una secuencia de operaciones que se basa en un tiempo o un evento determinado.

Entre los ejemplos de procesos que utilizan el control secuencial podemos mencionar las líneas de transferencia utilizadas en la producción en serie de productos en donde cada estación realiza una cierta operación sobre una parte, después, esta parte procesada es transferida a la siguiente estación y esta a su vez, se ve remplazada por una parte no procesada de la estación anterior. Otro ejemplo, es un proceso de manejo de materiales en el cual varios materiales se pesan, se mezclan se transportan y se descargan. Otros ejemplos lo constituyen las máquinas herramientas automáticas, procesos secuenciales en la operación de plantas químicas, las máquinas utilizadas en la producción de alimentos, etc. El uso del control lógico para hacer que una máquina o proceso realice una determinada secuencia de operaciones da lugar al término control lógico secuencial.

Una planta productora de cristal automotriz, provee un clásico ejemplo de la necesidad de control secuencial. El proceso necesario para fabricar un parabrisas debe seguir a otro en un orden particular. La descripción de la fabricación de el medallón de un automóvil es la siguiente:

- La pieza de vidrio plano, se pasa por las cortadoras para darle las dimensiones requeridas.
- Una vez cortada la pieza es pulida en sus cantos.
- La pieza es barrenada.
- Se le aplica la pintura cerámica.
- La pieza es templada en hornos.
- Se le colocan los conectores de los desempañadores a la pieza.

**Control Secuencial** Es un proceso que dicta el orden correcto de los eventos y asegura que un evento ocurrirá solo al final de otro evento. También se le puede calificar como control discreto en donde se tienen entradas y salidas en forma binaria, como un encendido y apagado (on / off)

## **HISTORIA Y DESARROLLO DE LOS PLC'S**

Una parte muy importante en el desarrollo de los PLC actuales fue la introducción del circuito integrado (ic) en 1959. Esto logro reducir el control de estado sólido, reduciendo así los costos, pero llevo 10 años para que los circuitos integrados se hayan vuelto mas sofisticados, con una mayor disponibilidad y a un costo cada vez menor.

El primer controlador programable fue diseñado por la Hydra-Matic División de General Motors en 1968. Antes de esa fecha, los relevadores electromecánicos que habían sido utilizados requerían un gran cantidad de cableado, y al momento de cambiar los modelos de los carros se gastaba una gran cantidad de horas/hombre en recablear los sistemas de control adaptándolos para los nuevos modelos.

General Motor estaba buscando una forma de ahorrar dinero en este costoso proceso. Los nuevos controladores fueron empleados con bases limitadas y tomó un tiempo para trabajar en los "bugs" (detalles) y hacer este tipo de control mas confiable y a menor costo.

Aun verdadero avance se dio en 1971 cuando Intel introdujo el microprocesador. El microprocesador posee una unidad central de procesamiento en un chip sencillo y hace posible el tener una pequeña computadora en un pequeño circuito impreso al agregar memoria y puertos de entradas o salida (I/O). Los microprocesadores se han vuelto cada vez mas poderosos a través del tiempo al incrementarse el numero de dígitos de datos que ellos pueden procesar a la vez. El primero que se introdujo podía manejar solo 4 dígitos o bits. Hoy los microprocesadores de 32 bits son totalmente disponibles y tienen precios accesibles. Al igual la capacidad de memoria de los circuitos integrados se ha ido incrementando con el paso del tiempo.

La introducción de un pequeño y económico PLC, comúnmente llamado micro - PLC ha hecho mas practico el uso de controladores programables para control en pequeña escala. Un control que utiliza media docena de relevadores puede ser reemplazado económicamente por uno de estos micro Plc's. Estos aparatos tienen un precio bajo, y con unos pocos cientos de dólares se puede tener un control flexible, sofisticado y versátil.

## **DEFINICIÓN DEL CONTROLADOR LÓGICO PROGRAMABLE**

Un PLC una pequeña computadora de propósito especial que ha sido diseñada especialmente para trabajar en ambientes rudos. Puede tomar decisiones lógicas e interconectarse con elementos de control lógico tanto del tipo sensores como actuadores, esta equipada además con puertos especiales de salida/entrada y con un lenguaje de programación especial para control.

En resumen podemos decir que los controladores lógicos programables realizan las siguientes funciones básicas:

- Aceptan y almacenan un conjunto de instrucciones llamada programa.

- **Aceptan información (Entradas y salidas) acerca del estado de los elementos de la máquina o proceso controlado.**
- **Efectúan cambios en las salidas bajo control de acuerdo a la información recibida y al programa de control almacenado**

Una característica sumamente importante es que su programación y operación es sumamente simple. Todos los dispositivos que son comunes en sistemas de relevadores se incluyen y su implementación se logra por medio de herramienta física (hardware) o de lógico (software). Aquellos controladores programables con capacidades aritméticas incluyen generalmente la posibilidad de entradas y salidas analógicas, lo que implica la inclusión de conversión D/A y A/D. Algunos de estos PLC's más sofisticados pueden incluir: estrategias programadas para el control PID de algunas variables analógicas, servo controladores de posición utilizando servomotores de CD o incluso motores de pasos. Todos estos agregados resultan de los procesos y/o máquinas automáticas en donde los PLC's se han aplicado sin embargo el núcleo principal lo constituye el control lógico. Desde esta perspectiva se puede decir que los PLC's se encuentran localizados entre lógica de estado sólido y programadores de secuencia (mecánicos o de otro tipo), y microcomputadoras (cuyas capacidades de programación y aplicaciones son más amplias que la sola aplicación al proceso de hacer decisiones lógicas).

La herramienta matemática para el estudio de los controladores lógicos es el álgebra booleana. Los controladores resuelven en forma seriada, ecuaciones booleanas o instrucciones lógicas dadas en forma de diagrama de escalera. A diferencia de circuitos de control secuencial implementados con relevadores o elementos de estado sólido que resuelven las ecuaciones booleanas o el diagrama escalera en paralelo, esto es, todas las ecuaciones booleanas a la vez, el PLC las resuelve una por una, y por cada condición lógica de cada ecuación.

## **GRAF CET**

### **UNA PODEROSA HERRAMIENTA PARA LA ESPECIFICACIÓN DE CONTROLADORES LÓGICOS**

#### **INTRODUCCIÓN**

El objetivo del controlador lógico es reaccionar a los cambios en las entradas modificando el estado de las salidas como una función de las reglas de control (o ecuaciones booleanas), las cuales están especificadas por el diseñador.

El caballo de batalla en el control lógico industrial es el controlador lógico programable (PLC). Los PLC's son usados extensivamente y algunas veces exclusivamente para automatizar la producción de energía, tratamientos químicos y síntesis, procesos de manufactura y sistemas de transportación.

Antes de la introducción del primer PLC, el control lógico era implementado por lógica alamburada, en donde los relevadores electromagnéticos, neumáticos, y compuertas electrónicas y flip flops eran usados. Con estas tecnologías, los problemas de automatización complejos eran difíciles de manejar o de darles mantenimiento y el hacer modificaciones en el sistema de control era un dura tarea. Cuando aparecieron los microprocesadores y tiempo después cuando estos se abarataron y se hicieron mas poderosos fue cuando inicio el auge de los PLC's.

Hoy, los PLC's son usados extensivamente en muchas clases de industrias, y son capaces de manejar cientos de entradas y salidas. Los modelos mas grandes pueden intercomunicarse y enviar datos a través de redes y algunos tienen funciones como, cálculo y regulación.

En el tiempo en que los controladores lógicos eran sistemas pequeños, los modelos clásicos eran suficientes para especificar las reglas de control lógico (ecuaciones). Existían dos clases de modelos. La mas representación mas famosa son los diagramas escalera (RLL en ingles), la cual es una representación gráfica basada en una analogía con los sistemas de relevadores físicos.

En los inicios de los años 70's la necesidad para describir complejos sistemas de control, aunado al avance de la tecnología de los microprocesadores que permitían guardar mas instrucciones era evidente. Con esto para describir un complejo sistema de control era necesario describir el comportamiento de muchas entradas y salidas lo que producía diagramas escalera demasiado grandes.

Mas precisamente las necesidades eran las siguientes:

- Describir la secuencia de los estados de sistemas que contenían un gran cantidad de estados.
- Tomar en cuenta la concurrencia (para una simplificación y un fácil entendimiento). Esto proviene del hecho que algunos subsistemas son parcialmente independientes.



- Generalmente dado el estado de un sistema, solo unas pocas entradas afectan el estado y solo unas pocas salidas pueden cambiar. Entonces describir el comportamiento de dichos cambios.
- Tener un claro entendimiento del comportamiento de las entradas y salidas de un sistema de control lógico.

El modelo de Grafcet , proviene de estas necesidades. Creado de la filosofía de Redes de Petri es una muy buena herramienta para la especificación de controladores lógicos. Las **Redes de Petri** son una herramienta gráfica que permiten un modelo de concurrencia y sincronización (necesidades 1 y 2). Un lugar en la Red de Petri representa un componente de el estado. Con esto, cuando un componente cambia, el estado cambia. En Grafcet, el comportamiento de entradas/salidas es especificado sin ambigüedad.

El modelo de Grafcet fue definido por un grupo francés. La mitad de las 24 personas que firmaron el reporte final, de fecha de agosto de 1977, son personal académico (incluyendo el autor) y la otra mitad pertenecen a la industria. Así, en 1987, un estándar francés se volvió internacional. Los comités internacionales de varios países, entre ellos, Japón, Canadá y el Reino Unido, declararon la aprobación explícita de la publicación. Posteriormente Grafcet fue adoptado como estándar en Estados Unidos pero bajo el nombre de Sequential Function Chart (SFC).

Un sistema en Grafcet se comporta básicamente como una Red de Petri con solo unas pequeñas diferencias, pero la principal es que propone una interpretación sencilla de entradas-salidas.

## **NOCIONES BÁSICAS.**

Un *grafcet* es una gráfica que tiene dos tipos de nodos: etapas y transiciones (un *grafcet* contiene al menos una etapa y una transición).

### **ETAPAS Y TRANSICIONES**

*Etapas:* Una etapa se representa por un cuadro y esta puede tener dos estados: activa (esta es representada por un token) o inactiva. La representación de las etapas se muestra a continuación:

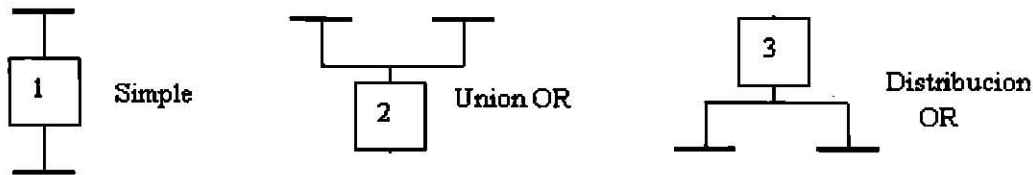


Una etapa que debe estar activa cuando el sistema es iniciado se representa por un doble cuadro y se conoce como etapa inicial. Las acciones son asociadas con las etapas y estas son las salidas de el *grafcet*.

**Transiciones:** Para que una etapa se lleve a cabo es necesario que se presenten un grupo de condiciones. A estas se les llama *Condiciones de Transición* o *Transiciones*. La representación de las transiciones es la siguiente:



**Los lazos directos:** Se le llama así a las líneas que unen las transiciones con las etapas. La representación de los lazos directos es ilustrada a continuación:



Un lazo directo debe correr siempre de una etapa a una transición o de una transición a una etapa.

Algunas partes del diseño del control lógico pueden ser descritas separadamente usando macropasos (subrutinas) para simplificar el modelo. Esto permite refinamientos que ya habían sido usados en las Redes de Petri. Gracias a las macroacciones se puede modelar un sistema a través de distintos modelos teniendo una relación jerárquica. Esto es particularmente útil para tomar en cuenta dispositivos de seguridad.

Un grafcet o SFC puede ser implementado ya sea por un lenguaje literal o por lenguaje gráfico. Actualmente existen algunos PLC's basados en Grafcet que poseen un editor gráfico. La estructura del grafcet o SFC es introducida en forma de dibujos en donde el usuario dibuja el grafcet directamente en la pantalla, y agrega acciones o condiciones en el lenguaje que el ha escogido. Es mucho mas sencillo de entender su comportamiento, y de depurar.

Sistemas de un tamaño razonable pueden ser descritos en Grafcet, y al parecer, este modelo es usado extensamente en la industria francesa debido a que es muy fácil entenderlo.

Ahora que el uso de Grafcet ha crecido considerablemente en los últimos años existe ahora un problema: Debido a que muchas compañías, (particularmente en los Estados Unidos), han ya desarrollado muchos programas (algunos de ellos muy grandes) basados en especificaciones de diagramas escalera (RLL). Desde que estos programas están funcionando y deben tener mantenimiento, es difícil transportarlos a una especificación Grafcet. El problema en si, es la transportación de especificaciones de sistemas de automatización basados en diagramas escaleras a sistemas basados en Grafcet. El hecho de la lógica de escalera (diagramas escaleras) haya sido tan popular, es debido a el papel que el relevador ha desempeñado en el desarrollo del control secuencial. Actualmente se están

haciendo grandes investigaciones en donde se han intentado elaborar algoritmos que conviertan automáticamente un diagrama escalera RLL a Grafcet.

## EJEMPLOS DE IMPLEMENTACIÓN DE GRAFCET

### DESCRIPCION

A continuación se muestran algunos ejemplos de la implementación de Grafcet para la resolución de problemas comunes de automatización.

Cabe señalar que estos ejemplos han sido probados en un laboratorio, además debe tenerse en cuenta que la implementación se realizó a través de un PLC modelo TSX-17 de Telemecanique, de tal forma que los mnemónicos u comandos de programación del PLC solo funcionan para los modelos Telemecanique, de cualquier forma mediante el entendimiento del diagrama de Grafcet, se puede hacer la implementación en cualquier otro PLC, conociendo sus comandos para Grafcet.

Debe tomarse en cuenta que la programación en Grafcet para cada tipo de PLC es distinta, al igual que conjunto de instrucciones.

En esta sección, se utilizó el PLC de Telemecanique Modelo TSX -17, y su lenguaje de programación para elaborar los ejemplos que aquí se muestran. En el PLC de Telemecanique un Grafcet se programa mediante 3 secciones: Sección Preliminar, Sección Grafcet, y la Sección Posterior. Estas 3 secciones se verán para cada ejemplo.

La programación de Grafcet en el PLC Telemecanique TSX-17 cuenta de 3 Partes:

#### 1) Sección preliminar.

En esta sección se declara el uso de Grafcet en el PLC, se programa la ejecución del Bit de Grafcet así como las características de arranque en frío o caliente de Grafcet.

#### 2) Sección Grafcet.

En esta sección se indica el número de etapas y las condiciones de transición entre cada etapa. (la etapa inicial esta en recuadro).

Además, para programar en Grafcet en el PLC es necesario también el uso de ciertos símbolos:

- = \* = Indica el inicio de la programación en Grafcet. (Etapa No 1).
- \* - Indica el inicio de cualquier otra etapa que no sea la inicial
- # Indica cual es la etapa siguiente para la cual se definieron las condiciones de transición.

#### 3) Sección Posterior.

En esta sección se indica que etapa se va a realizar y esa etapa a cuantas etapas va a activar.

Para efectuar la programación de Grafcet en el PLC de Telemecanique Modelo TSX-17, es necesario el uso de los bits especiales siguientes:

<i>SY21</i>	Indica el uso de Grafcet
<i>SY23</i>	Efectúa una pausa en la ejecución de Grafcet
<i>SY00</i>	Bit para arranque en frío de Grafcet
<i>SY01</i>	Bit para arranque en caliente de Grafcet

Cabe mencionar el significado de los arranques:

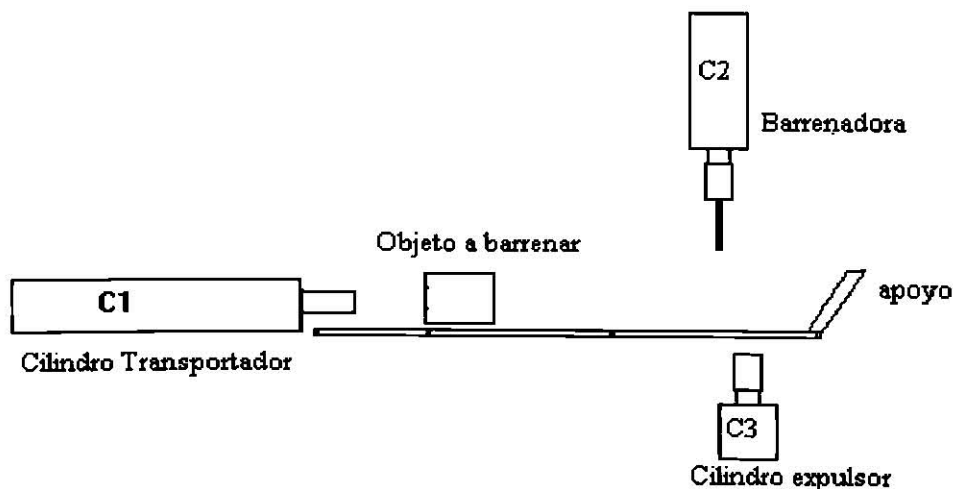
*Arranque En Frío* : Provoca una interrupción en la secuencia y la inicia de nuevo.

*Arranque En Caliente*: Continúa la ejecución de la secuencia en donde se detuvo con el bit sy23.

### **EJEMPLO 1: AUTOMATIZACION DE UNA MÁQUINA BARRENADORA**

#### **Descripción:**

Se desea controlar el proceso del siguiente dibujo:



En donde el actuador C1 deberá extenderse para transportar al objeto hacia la zona de barrenado. Al llegar a esta zona (hasta el apoyo), una barrenadora empujada por el actuador 2 comienza a bajar y a perforar el objeto. Una vez perforada la pieza la barrenadora sube por la acción del actuador 2. Posteriormente el actuador 3 se extiende para expulsar la placa del apoyo y este actuador regresa a su posición original al igual que el actuador 1. Así el proceso se vuelve a repetir.

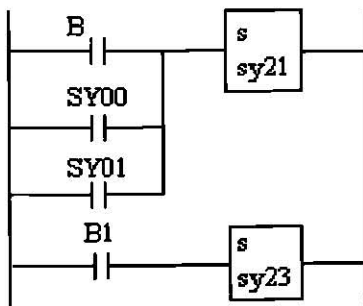
Cabe mencionar que cada actuador cuenta con sensores magnéticos de avance, y retroceso:

Sensores	Descripción
S1	sensor que indica que el actuador 1 esta retraído
S2	sensor que indica que el actuador 1 esta extendido
S3	sensor que indica que el actuador 2 esta retraído
S4	sensor que indica que el actuador 2 esta extendido
S5	sensor que indica que el actuador 3 esta retraído
S6	sensor que indica que el actuador 3 esta extendido

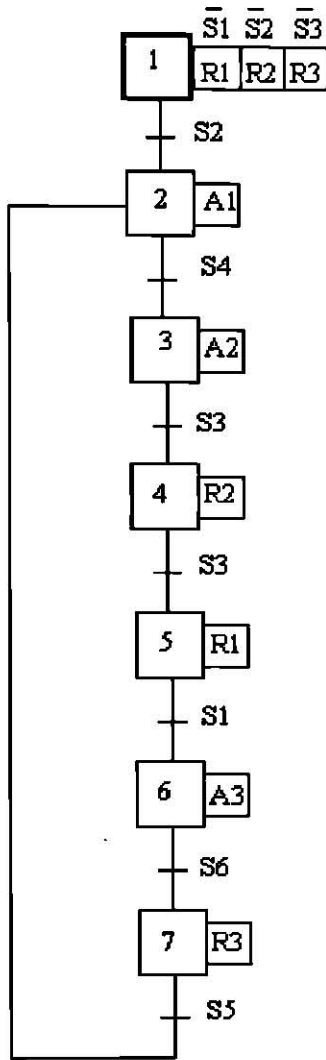
Tabla de definición de las variables.

Sensores	Entradas (PLC)	Botones	Entradas (PLC)	Accionamientos	Salidas (PLC)
S1	I1	B (Arranque )	I7	A1	O1
S2	I2	P (Paro )	I8	R1	O2
S3	I3	A (Reset a la secuencia)	I9	A2	O3
S4	I4			R2	O4
S5	I5			A3	O5
S6	I6			R3	O6
				Foco VERDE	O7
				Foco ROJO	O8

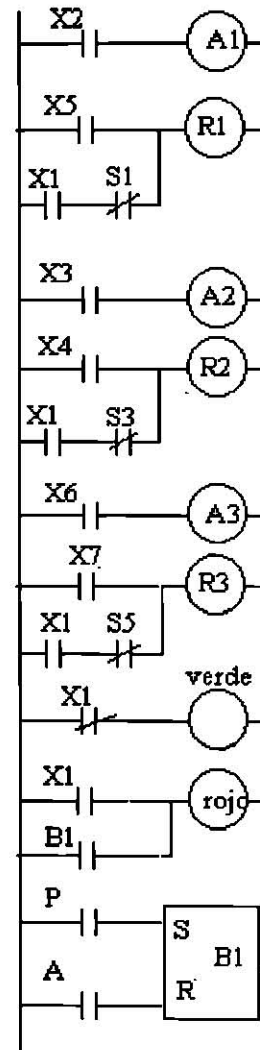
Sección preliminar: (dibujo)



**Sección Grafcet: (dibujo)**



**Sección Posterior (Dibujo)**



**Descripción de la sección preliminar**

Se puede apreciar que mediante la activación de B inicia la secuencia, y mediante la activación de P ocurre una pausa.

Para salir de esta pausa debe activarse el botón A o B.

Si se pulsa el botón A después de una pausa, el proceso regresa al estado inicial y ocurre un arranque en caliente.

Si se pulsa el botón B después de una pausa, el proceso continúa desde el lugar donde ocurrió la pausa y ocurre un arranque en frío.

SY00 y SY01 son bits internos del PLC de Telemecanique que configurados de la forma que se muestra en la sección preliminar tienen las siguientes funciones:

- \* SY00 Continúa desde el lugar donde ocurrió la pausa (arranque en frío)
- \* SY01 Inicia la secuencia desde el principio (arranque en caliente)

### ***Descripción de la sección Grafcet***

Esta sección se interpreta de la siguiente manera:

En la etapa 1 deben retroceder los 3 actuadores. Esto debe ocurrir cuando se cumplan las condiciones: S1'S2'S3'.

S2 o S5 son las la condición de transición para que ocurra la etapa 2.

En la etapa 2 ocurre el avance del actuador 1

S4 es la condición para la etapa 3

En la etapa 3 avanza el actuador 2

S3 es la condición de transición para que ocurra la etapa 2

En la etapa 4 retrocede el actuador 2

S3 es la condición para la etapa 5

En la etapa 5 retrocede el actuador 2

S1 es la condición para la etapa 6

En la etapa 6 avanza el actuador 1

S1 es la condición para la etapa 7

En la etapa 7 retrocede el actuador 2

### ***Descripción de la sección Posterior***

Esta sección muestra las acciones que deben ejecutarse al cumplirse las condiciones para tal etapa.

A las etapas se les da ( por especificaciones del lenguaje del PLC ) la variable Xn. Así la etapa 1 es nombrada X1; la etapa 2, X2 y así sucesivamente.

Por ejemplo en el primer peldaño podemos observar que el actuador No. 2 debe avanzar al cumplirse la etapa 2 (X2).

En el segundo y tercer peldaño podemos observar que para que el actuador 1 retroceda, debe cumplirse la etapa 5 (X5), o la 1er etapa (X1) y el sensor S1 no debe estar presente (S1').

### ***Programación en el PLC***

#### **Sección Preliminar :**

S000	L	I9
S001	O	SY00
S002	O	SY01
S003	S	SY21
S004	L	B1
S005	S	SY23

**Sección Grafcet :**

S006	==*=	O1
S007	L	I7
S008	A	I1
S009	A	I3
S010	A	I5
S011	#	O2
S012	-*-	O2
S013	L	I2
S014	#	O3
S015	-*-	O3
S016	L	I4
S017	#	O4
S018	-*-	O4
S019	L	I3
S020	#	O5
S021	-*-	O5
S022	L	I1
S023	#	O6
S024	-*-	O6
S025	L	I6
S026	#	O7
S027	-*-	O7
S028	L	I5
S029	#	O2

**Sección Posterior :**

S030	==*=	POST
S031	L	X2
S032	=	O1
S033	L	X1
S034	AN	I1
S035	O	X5
S036	=	O2
S037	L	X3
S038	=	O3
S039	L	X1
S040	AN	I3
S041	O	X4
S042	=	O4
S043	L	X6
S044	=	O5
S045	L	X1
S046	AN	I5
S047	O	X7



S048	=	O6
S049	LN	X1
S050	=	O7
S051	L	X1
S052	O	B1
S053	=	OO
S054	L	I8
S055	S	B1
S056	L	I7
S057	R	B1
S058	EP	

### ***Comentarios***

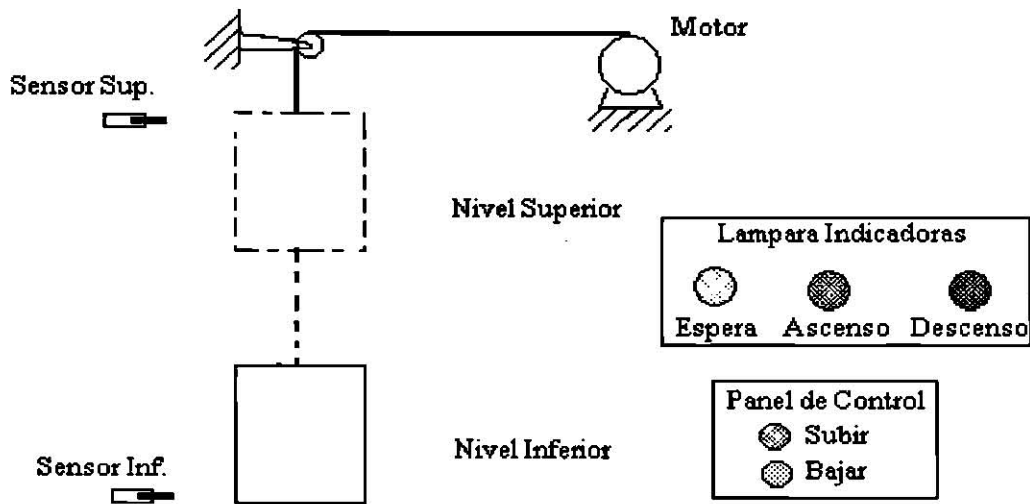
Con este ejemplo se pueden apreciar las ventajas de la implementación de un sistema de Control Lógico Secuencial en un PLC por medio de Grafcet y todo lo que esta implica. Se entendió la forma de implementar un sistema, la cual se realiza de una forma "mas natural" definiendo las secuencias deseadas y las condiciones para cada secuencia. De esta forma ahora conocemos una forma de programar que para algunas aplicaciones es muy sencilla y mas practica que el programar por diagrama escalera sobre todo en aquellas aplicaciones donde se tienen ya sea un gran numero de variables en un peldaño o donde existen condiciones que se cumplen para diferentes etapas.

### ***EJEMPLO 2: CONTROL DE UN ELEVADOR DE DOS PISOS***

#### ***Descripción:***

Se trata de controlar la operación de un elevador de dos pisos. Se tienen dos funciones en este elevador: ascenso y descenso. Se desea que en el momento que la cabina se encuentre en el nivel inferior y se oprima el botón de ascenso, la cabina se eleve hasta el segundo nivel, y permanecer en dicho nivel hasta que el botón de descenso sea presionado.

Se puede esquematizar el problema como sigue:

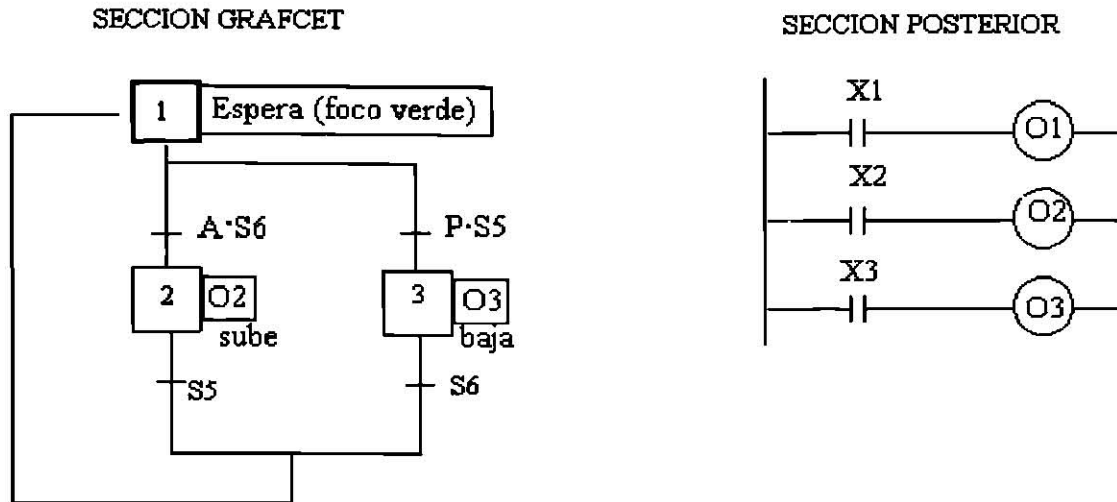


Para la implementación, se optó sustituir el motor por un actuador lineal, también se implementaron tres luces indicadoras del estado en el que se encontraba dicho elevador.

Ello implica que las acciones en el motor sean las siguientes:

Estado	Acción	Luz Indicadora
1.- Espera	Motor apagado	Verde
2.- Ascenso	Motor en sentido directo	Azul
3.- Descenso	Motor en sentido inverso	Rojo

A continuación se representan las dos secciones que corresponden a la programación del PLC en Grafcet. Nótese que la sección preliminar es omitida, ya que las condiciones del problema hacen innecesario el restablecimiento y a pausa.



A continuación se escriben las variables que se utilizaron en la programación del PLC, posteriormente se presenta la programación del PLC que se utilizó para automatizar el proceso.

Descripción de Entradas	PLC	Dibujo	Descripción de Salidas	PLC
Sensor de primer nivel	I 02	S5	Espera	O 01
Botón de ascenso	I 03	A	Ascenso	O 02
Sensor de segundo nivel	I 04	S6	Descenso	O 03
Botón de descenso	I 05	P		

**Programa Implementado**

S000	==	01	S013	==	POST
S001	L	I 02	S014	L	X1
S002	A	I 03	S015	=	O 01
S003	#	02	S016	L	X2
S004	L	I 04	S017	=	O 02
S005	A	I 05	S018	L	X3
S006	#	03	S019	=	O 03
S007	-*-	03	S020	EP	
S008	L	I 02			
S009	#	01			
S010	-*-	02			
S011	L	I 04			
S012	#	01			

**Comentarios**

Con la implementación de este problema, observamos una aplicación práctica de la bifurcación en la programación de Grafcet del PLC. Al realizar esto constatamos que es un método sencillo para resolver un problema aparentemente complejo.

La bifurcación es una herramienta muy útil para la toma de decisiones en la secuencia de los eventos, ello convierte al PLC en un elemento "inteligente" en el control lógico digital. Aunque esto ilustra un potencial de Grafcet, realmente la aplicación en la industria, requeriría tomar en cuenta otras consideraciones, tales como la velocidad de desplazamiento de los motores, sensores en las puertas, paro de emergencia, seguros, etc.

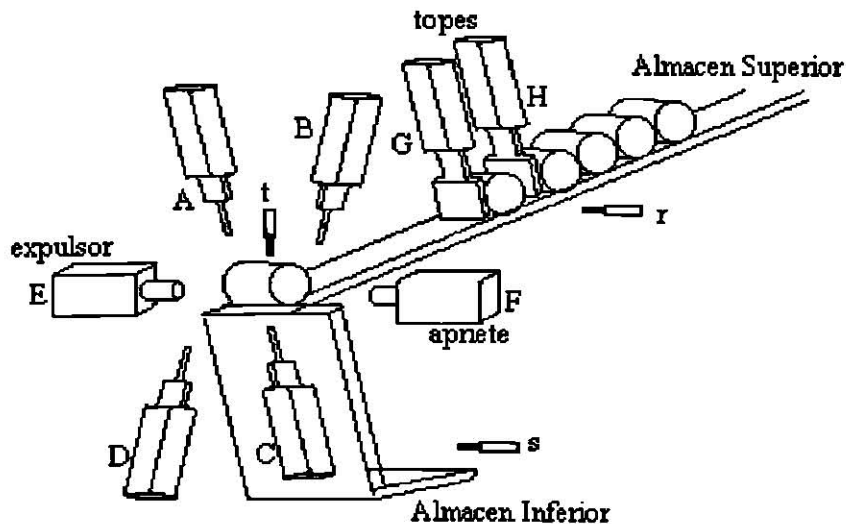
**EJEMPLO 3: PUESTO AUTOMÁTICO DE ALIMENTACIÓN DE PIEZAS**

**Descripción:**

Se busca automatizar una cadena de fabricación que consiste en puestos separados por topes de almacenamiento de piezas. El ciclo de maquinado se repite sin intervención humana, siempre que hayan piezas en el almacén superior, no haya una pieza en la posición de maquinado y haya sitio en el almacén inferior.

Desde el almacenamiento superior, un tope mecánico libera una pieza al comienzo de cada ciclo. Al final del ciclo de trabajo, la pieza se evacua al almacenamiento que alimenta al puesto siguiente. Unos detectores controlan el nivel de los almacenes y la presencia de piezas.

El problema se puede plantear con la cadena de fabricación que se ilustra:



Siendo este el caso, implica que las acciones de los actuadores sean:

1.- Subida del tope inferior	G-		
2.- Apriete de la pieza y bajada del tope inferior	F+	G+	
3.- Avance de las unidades de trabajo	(A+B+C+D+)	G+	
4.- Retroceso de unidades de trabajo	(A-B-C-D-)		
5.- Desapriete, expulsión y subida del tope superior	F-	E+	H-
6.- Retroceso del expulsor y bajada del tope superior	E-	H+	

Nota: Un símbolo " - " significa retroceso y un " + " significa avance.

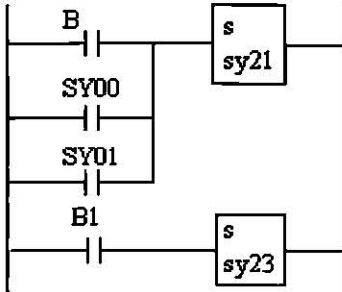
A diferencia del problema original, en el que se sugería que el retroceso del tope superior fuese simultáneo con el retroceso de las unidades de trabajo, se situó este retroceso simultáneo con el desapriete y expulsión, es decir, se retrasó un estado. Esto permite que se utilice una sola salida para los actuadores H y E, aunque se tenga que utilizar un *relevador* o una salida del PLC extra para que la acción de E fuese la inversa de H.

Por otra parte, la última acción se *elimina*, ya que las condiciones que tiene la primera permite que se haga innecesaria. Se consideraron a los actuadores A, B, C, D como *uno solo* que correspondía al actuador C1, como se mencionó anteriormente, E y H como C4, F como C2 y G como C3. Estas modificaciones se hicieron en aras de usar un solo PLC que tiene solamente 12 entradas y 8 salidas. Además, para poder implementar y simular el problema, se utilizaron botones pulsadores que simulan a los detectores de piezas.

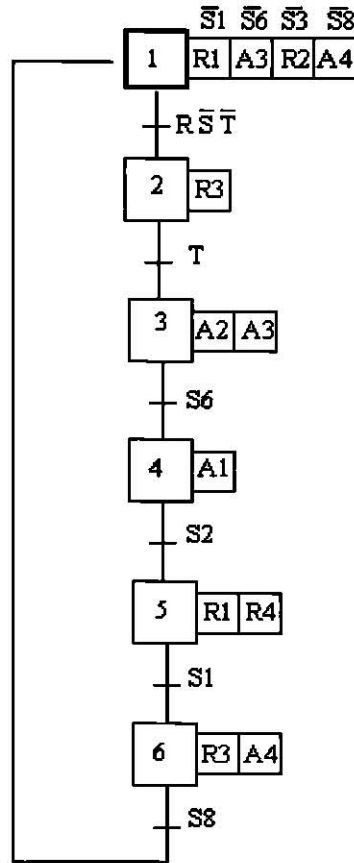
Sensores	Descripción
S1	sensor que indica que el actuador 1 esta retraído
S2	sensor que indica que el actuador 1 esta extendido
S3	sensor que indica que el actuador 2 esta retraído
S4	sensor que indica que el actuador 2 esta extendido
S5	sensor que indica que el actuador 3 esta retraído
S6	sensor que indica que el actuador 3 esta extendido
S7	sensor que indica que el actuador 4 esta retraído
S8	sensor que indica que el actuador 4 esta extendido

A continuación se representan las tres secciones de que corresponde la programación del PLC en Grafcet.

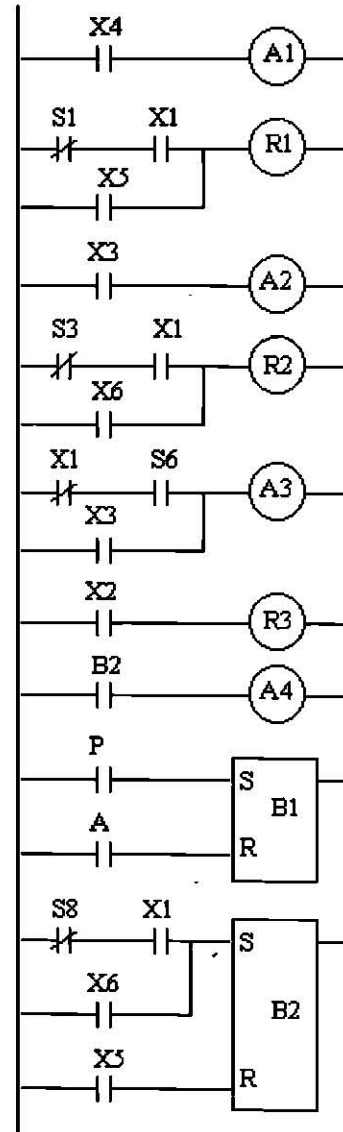
SECCION PRELIMINAR



SECCION GRAFCET



SECCION POSTERIOR



A continuación se describe a detalle la programación del PLC que se utilizó para automatizar el proceso. La primera sección corresponde a la "sección preliminar", la segunda a la "sección Grafcet" y la tercera a la "sección posterior".

#### Descripción de Entradas

--	I 00	
R	I 01	
S	I 02	
T	I 03	
A	I 04	arranque
P	I 05	paro
B	I 06	restablecer
S62	I 07	
S11	I 08	
S82	I 09	
S21	I 10	
S32	I 11	

#### Programa Implementado

S000	L	I 06
S001	O	SY00
S002	O	SY01
S003	S	SY21
S004	L	B1
S005	S	SY23
S006	=*=	01
S007	L	I 01
S008	AN	I 02
S009	AN	I 03
S010	#	02
S011	-*-	02
S012	L	I 03
S013	#	03
S014	-*-	03
S015	L	I 07
S016	#	04
S017	-*-	04
S018	L	I 10
S019	#	05

S020	-*-	05
S021	L	I08
S022	#	06
S023	-*-	06
S024	L	I 09
S025	#	01
S026	==*=	POST
S027	L	X01
S028	AN	I 08
S029	O	X05
S030	=	O 00
S031	L	X02
S032	=	O01
S033	L	X03
S034	=	O 02
S035	L	X01
S036	AN	I 07
S037	O	X03
S038	=	O03
S039	L	X04
S040	=	O04
S041	L	X01
S042	AN	I 11
S043	O	X06
S044	=	O06
S045	L	I 05
S046	S	B1
S047	L	O 00
S048	R	B1
S049	L	B2
S050	=	O 07
S051	L	X01
S052	AN	I 09
S053	O	X06
S054	S	B2
S055	L	X05
S056	R	B2
S057	EP	

---



## Comentarios

El problema de esta forma resuelto, funciona con todas las especificaciones del problema propuesto pero con las limitaciones de la implementación en un laboratorio.

Si se tuviese el caso de la implementación real en la industria, se optaría por accionar a los cuatro actuadores con unidades de trabajo A, B, C, D con la misma salida del actuador 1 (C1) utilizado aquí, sin embargo, se utilizarían los sensores límite de carrera en "AND", (probablemente implementada en transistores, o circuitos integrados o, por ser la opción aparentemente mas económica, en caso de no contar con un PLC con un número sobrado de entradas), para poder asegurar que todos los actuadores están retraídos, o se encuentran extendidos. Lógicamente, también se optaría por utilizar los sensores reales, en lugar de utilizar botones.

El hecho de haber colocado a una misma salida los actuadores H y E, implica que si, en la industria se tiene un PLC de mayor número de salidas, se puedan manejar por separado. En caso contrario, el hecho de manejarlos juntos y utilizar un relevador extra, no es una mala opción.

## IMPORTANCIA ACTUAL DE GRAFCET

A continuación se muestran algunos resúmenes de documentos de conferencias e investigaciones concernientes al uso actual de Grafcet y Redes de Petri con el fin de demostrar la creciente importancia que han tenido estas herramientas en los últimos años:

### *Analysis of function charts for control systems using Petri nets*

Author: El Rhalibi, A.; Prunet, F.; Durante, C.

Corporate Source: Universite Montpellier, Montpellier, Fr

Conference Title: Proceedings of the 1994 IEEE Symposium on Emerging Technologies & Factory Automation

Publication Year: 1994

**Abstract:** A technical process is the controlled flow of matter, energy, and/or information. As opposed to the widely investigated continuous control, this paper concentrates on event-related control modelling and validation. In this case, process and related process control can be described by steps and transitions, which can be represented very concisely by Function Charts for Control Systems model (Grafcet in French).

As it is difficult to validate the Grafcet model, we propose to transform a Grafcet in Petri net model that is a rigid mathematical tool, and which can be used for a very general approach to event-related process control in specification, simulating, checking, debugging, and evaluating performance of any given process control system.

**Descriptors:** \*Petri nets; Process control; Control systems; Systems analysis; Flowcharting; Mathematical models; Sensors; State estimation; Computer

simulation; Automation

***From modelling using function charts for control systems to analysis using Petri nets***

Author: El Rhalibi, A.; Prunet, F.; Durante, C.

Corporate Source: Universite Montpellier II/CNRS, Montpellier, Fr

Conference Title: Proceedings of the 3rd International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems

Conference Location: Durham, NC, USA

Conference Sponsor: IEEE

**Abstract:** A technical process is the controlled flow of matter, energy, and/or information. As opposed to the widely investigated continuous control, this paper concentrates on event-related control modelling and validation. In this case, process and related process control can be described by steps and transitions, which can be represented very concisely by a Functions Charts for

Control Systems model (Grafcet in French). As it is difficult to validate the Grafcet model, we propose to transform a Grafcet in Petri net model that is a rigid mathematical tool, and which can be used for a very general approach to event-related process control in design and analysis.

***Development environment for the SSC control system***

Author: Murray, Doug; Martinsen, Garth; Wang, Judy

Corporate Source: SSC Lab, Dallas, TX, USA

Conference Title: Proceedings of the 3rd International Conference on Accelerator and Large Experimental Physics Control Systems

**Abstract:** The SSC is developing a design environment for control system development within the context of EPICS. The environment is aimed at developers of applications using the EPICS Input/Output Controller (IOC). The unique aspect of this effort is our emphasis on providing a simple and intuitive development environment compared with tools currently available. Our most important goal in this effort has been to hide the complexity of EPICS IOC development from the developers.

This paper describes two tools which are under development; The Function Block Editor (FBE) and the State Machine Editor (SME). The FBE provides a visual editing environment for graphically describing control processes as a set of related functional blocks. The configuration of functional blocks is then translated into IOC records and executed. SME allows the user to visually construct a sequence using a notation we are modeling after Grafcet. State machines are translated into EPICS State Notation Language programs and executed. Future extensions will be described. (Author abstract) 2 Refs.

***Grafcet for intelligent supervisory control applications***

Author: Arzen, Karl-Erik

Corporate Source: Lund Inst of Technology, Lund, Sweden

Source: Automatica 30 10 Oct 1994, p 1513-1525

Publication Year: 1994

**Abstract:** The paper describes the application of Grafcet to supervisory, knowledge-based applications. Grafchart, a grafcet-based toolbox has been implemented in the real-time expert system environment G2. The toolbox supports steps, transitions, parallel paths, macro steps, and exception transitions. Several applications of the toolbox are presented. One of these is a full-scale industrial application where Grafchart is used to implement a hydrogen balance advisory control system.

*Automatic creation of sequential control schemes in polynomial time*

Authors: Klein, I.; Lindskog, P.

Conf. Title: Proceedings of the 32nd IEEE Conference on Decision and Control p. 211-16 vol.1

Publisher: IEEE

Language: English

Treatment: Practical

**Abstract:** Of all hard- and software developed for industrial control purposes, the majority is devoted to sequential, or binary valued, control and only a minor part to classical linear control.

The sequential parts of the controller are typically invoked during startup or shut-down phases to bring the system either into its normal operating region or into some safe standby region. Despite its

importance, fairly little theoretical research has been devoted to this area, and sequential control programs are still created manually without much support for a systematic approach.

We propose a method to create sequential control programs automatically and online upon request, for example when a plant fault has occurred. The main idea is to spend some effort off-line on modeling the process, and from this model generate the control strategy, i.e. the plan. Here we present a planning tool implemented in a real-time expert system called G2. The planning system contains algorithms for creating plans in form of minimal GRAFCET charts that show maximal parallelism. The algorithms can handle a restricted class of problems and for this class the complexity only increases polynomially with the number of state variables.

*G++: a graphical language to specify real-time parallel applications*

Authors: Brams, H.; Lobelle, M.; Detroz, G.; April, A.

Affiliation: Unite d'Inf., Univ. Catholique de Louvain, Belgium

Conf. Title: Proceedings. Euromicro Workshop on Parallel and Distributed Processing

p. 185-93

Publisher: IEEE Comput. Soc. Press

**Treatment:** Practical

**Abstract:** The G++ language belongs to the same family as Statecharts and GRAFCET. Like these languages, it is based on the state/transition model. The state and the transition concepts are

presented in this paper. Temporal constraints, including timeouts and deadlines, can be expressed in G++. Like Statecharts, G++ allows a hierarchical organisation of states in two different ways: parallel decomposition (AND-state) and sequential decomposition (OR-state) of states. But G++, unlike Statecharts, also provides graphical constructs to express synchronisation and resource sharing between concurrent entities. In most applications, concurrent entities need

to communicate. Communications can be built on top of the two basic interaction mechanisms. A small example (a respiration monitor) is used throughout the paper to illustrate the concepts and constructs.

### *Design recovery for relay ladder logic*

**Authors:** Falcione, A.; Krogh, B.H.

**Affiliation:** Dept. of Electr. & Comput. Eng., Carnegie Mellon Univ.,  
Pittsburgh, PA, USA

**Journal:** IEEE Control Systems Magazine

Vol: 13 Iss: 2 p. 90-8

Date: April 1993

Country of Publication: USA

**Language:** English

**Treatment:** Application; Practical

**Abstract:** An algorithm for converting relay ladder logic (RLL) programs for programmable logic controllers (PLCs) into sequential function chart (SFC) programs is introduced. Since an SFC (a standardization of Grafset) better represents the sequential flow of control logic, it is considerably easier to understand and modify SFC programs. The practical motivation for this study is to provide a tool for converting existing RLL programs into SFC programs by

recovering the underlying sequential control logic. A basic design recovery algorithm is presented and applied to a simple RLL program for one stage of a batch chemical process. Several fundamental issues in discrete control logic representation and analysis are identified

as fertile areas for future research.

### *Specification of real time control systems by means of sequential function charts*

**Authors:** Mallaband, S.

**Affiliation:** Bass Brewers Ltd., Burton-on-Trent, UK

**Conf Title:** Third International Conference on Software Engineering  
for Real Time Systems (Conf. Publ. No.344)  
p. 57-62

**Publisher:** IEE

**Language:** English

**Treatment:** Practical

**Abstract:** Very many specification techniques have been proposed for real time control systems, each with its own advantages and disadvantages. This paper considers the use of one of these

techniques-sequential function charts (SFCs). SFCs are not new-GRAF CET, a standard for such charts, was presented in 1982-but until recently they were not widely known. This has now changed and the technique has received considerable coverage in the technical literature. In addition, several control system manufacturers have adopted SFCs as a programming language. The paper begins with a brief survey of specification techniques for real time control systems; it moves on to trace the origins of SFCs and then describes them in some detail. Finally, the practical production of SFCs is discussed and the need for an SFC diagram editor is identified. Current options in this field are presented.

***GEMMA, the complementary tool of the GRAFCET***

**Authors:** Cloutier, G.; Paques, J.-J.

**Affiliation:** Chemin de l'Ecole Polytech., Que., Canada

**Conf. Title:** Conference Proceedings. Fourth Annual Canadian  
Programmable Control and Automation Technology Conference  
and Exhibition p. 12A1-5/1-10

**Publisher:** IEEE

**Language:** English

**Treatment:** Practical

**Abstract:** The basic features of GEMMA, a design guide for start and stop modes, are described. It is noted that the completed GEMMA takes the form of a macro-GRAF CET and can be programmed as such on some programmable controllers. It is an active tool, as it prompts

design-oriented questions very early in the project. It is applicable as an analytical checklist, a standardized communication/specification tool for training or contractual agreements, and a programming approach if the appropriate material is available. Its macro-representation allows the designer to zoom to detailed hardware and software specifications progressively and to consider possible human interventions from general functional requirements. This feature minimizes the number of iterations in the design phase, helps to decouple functional principles from mechanical fine-tuning, and fosters safe operation by providing a logical framework to make the most of the conclusions of various safety analysis techniques or plain common sense. It is concluded that GEMMA is well suited to a wide variety of cases in the field of programmable automation. (4 Refs.)

***Graphical and iconic programming languages for distributed process control: an object oriented approach***

**Authors:** Coote, S.; Gallagher, J.; Mariani, J.; Rodden, T.; Scott, A.; Shepherd, D.

**Affiliation:** Dept. of Comput., Lancaster Univ., UK

**Conf. Title:** 1988 IEEE Workshop on Visual Languages (Cat. No.TH0229-5)  
p. 183-90

**Publisher:** IEEE Comput. Soc. Press

**Language:** English

**Treatment:** Practical

**Abstract:** Distributed process control has gained in impetus over recent years as an alternative to central control systems. The object-oriented paradigm lends itself well to the description of such

distributed systems. The Oscar model of active objects embodies this paradigm, and a support environment for the design and implementation of process control systems built on this model is outlined. A graphical system description language based on an extended GRAFCET notation is described. This language is used to specify the interobject messaging and action required for a particular control task. The parsing of the diagram provides the code for controlling a

system. A higher-level iconic approach to the description of control strategies is also described. By restricting themselves to the domain of process control, the authors have been able to make substantial progress in the provision of an integrated support environment. The environment incorporates the notions of visual programming techniques and combines iconic and object-oriented paradigms to produce a consistent interface metaphor.

***Using Grafcet to design generic controllers***

**Authors:** Thomas, B.H.; McLean, C.

**Affiliation:** NBS, Gaithersburg, MD, USA

**Conf. Title:** 1988 International Conference on Computer Integrated  
Manufacturing  
p. 110-19

**Publisher:** IEEE Comput. Soc. Press

**Language:** English

**Treatment:** Practical

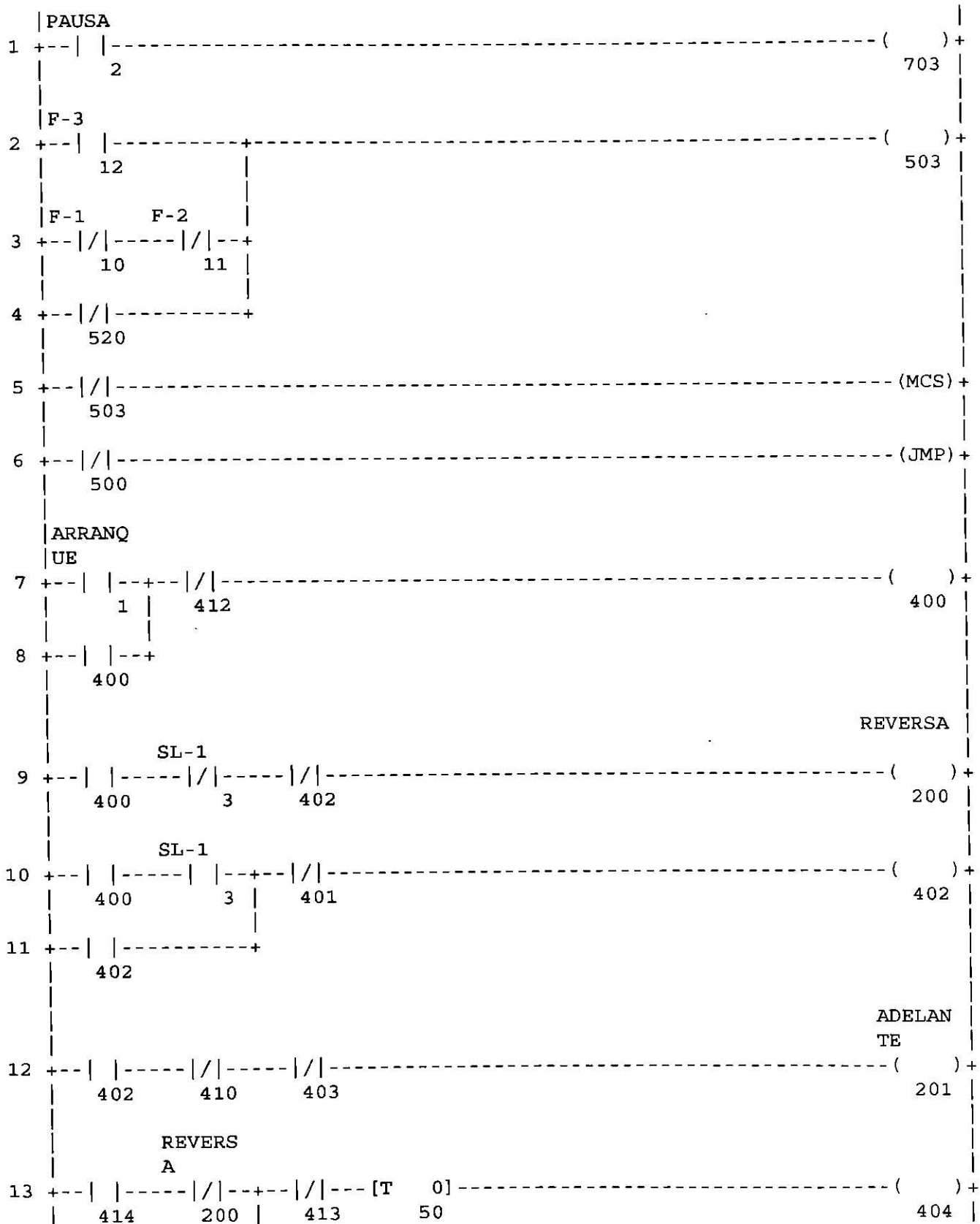
**Abstract:** Ways in which Grafcet is being used to help design generic controllers within the US National Bureau of Standards' Automated Manufacturing Research Facility are outlined. Grafcet is a powerful graphic language for expressing control flow and allows the expression of both parallel and sequential control logic. The rationale is described for using Grafcet as a design tool for

expressing software for a generic controller. Grafcet helps the designer determine: modularization of the code, functions that can be performed in parallel, communication between parallel processes, and problems in control flow.

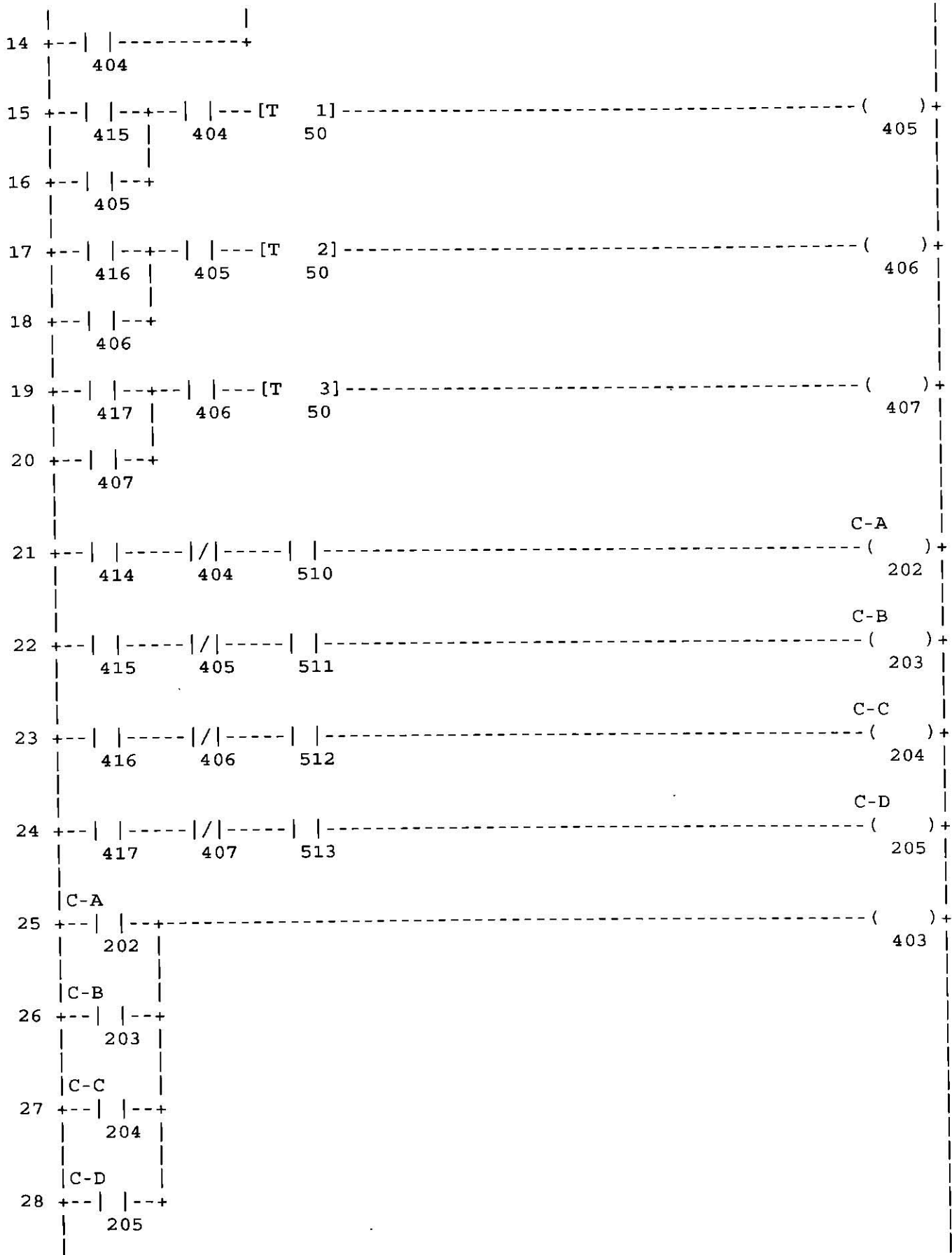
---

APENDICE

DIAGRAMA ESCALERA DEL PROYECTO "AUTOMATIZACIÓN DE UNA TOLVA MÓVIL"

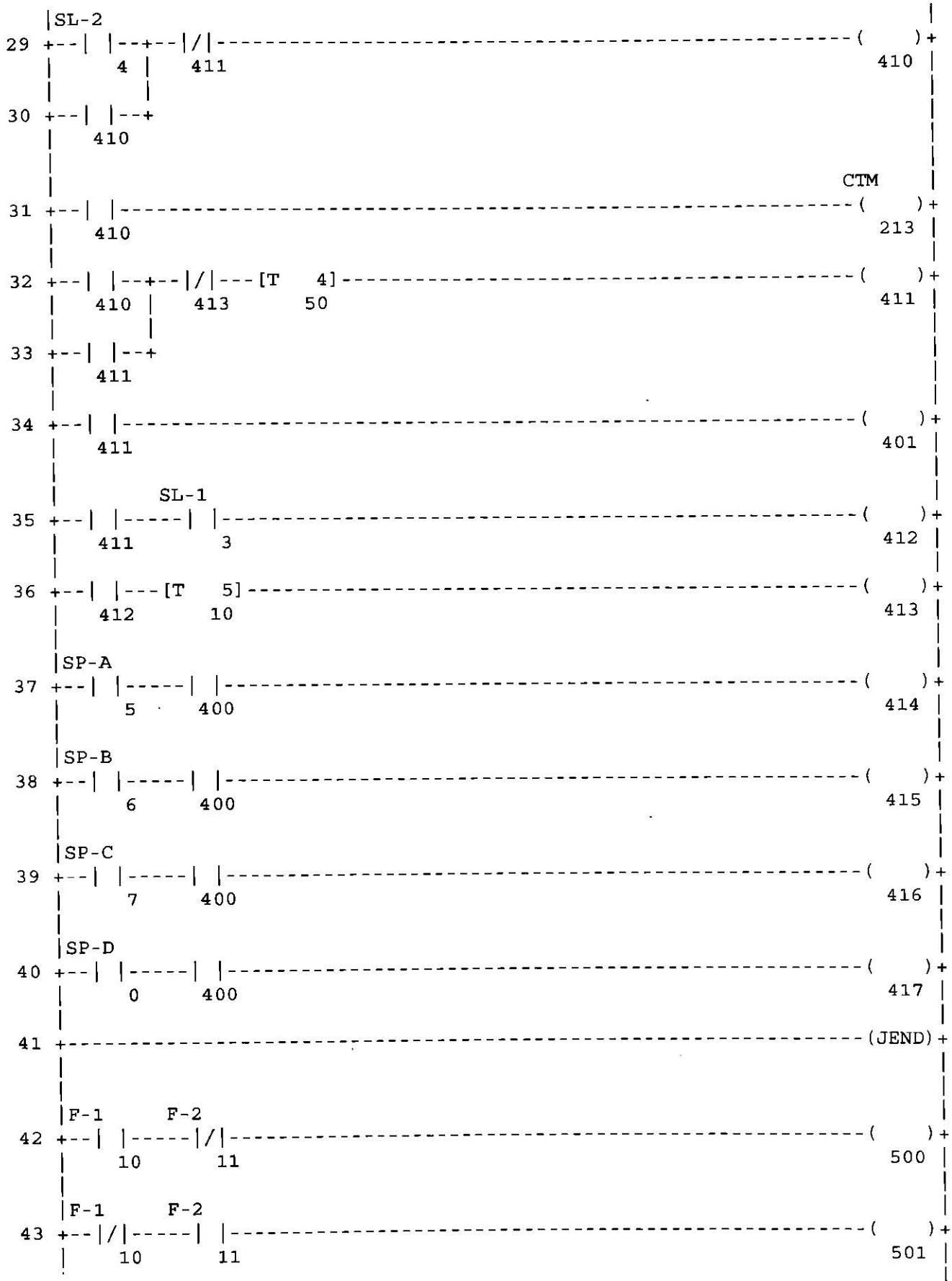


Automatización de una Tolva Móvil y Grafcet



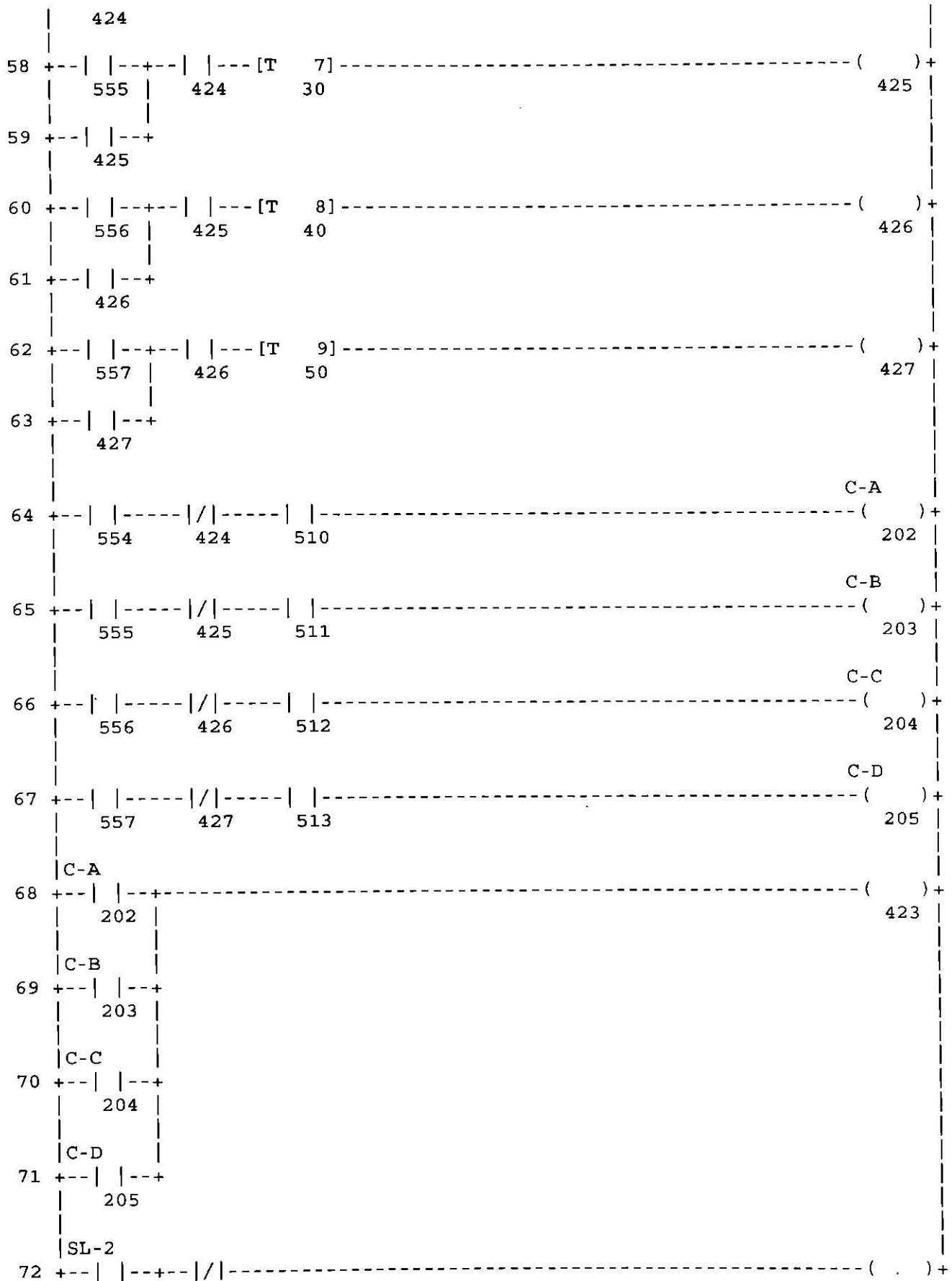


Automatización de una Tolva Móvil y Grafcet



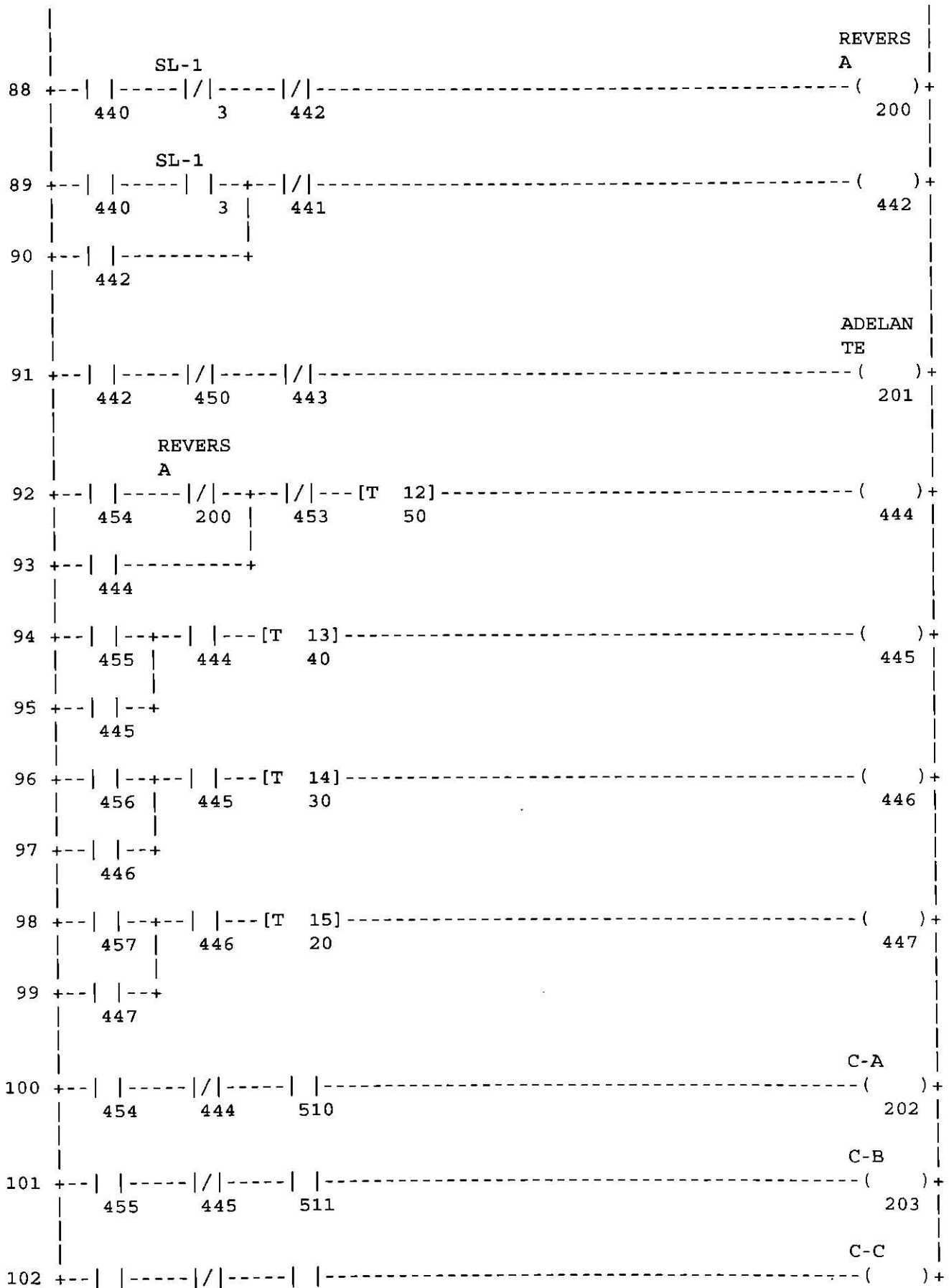


Automatización de una Tolva Móvil y Grafset

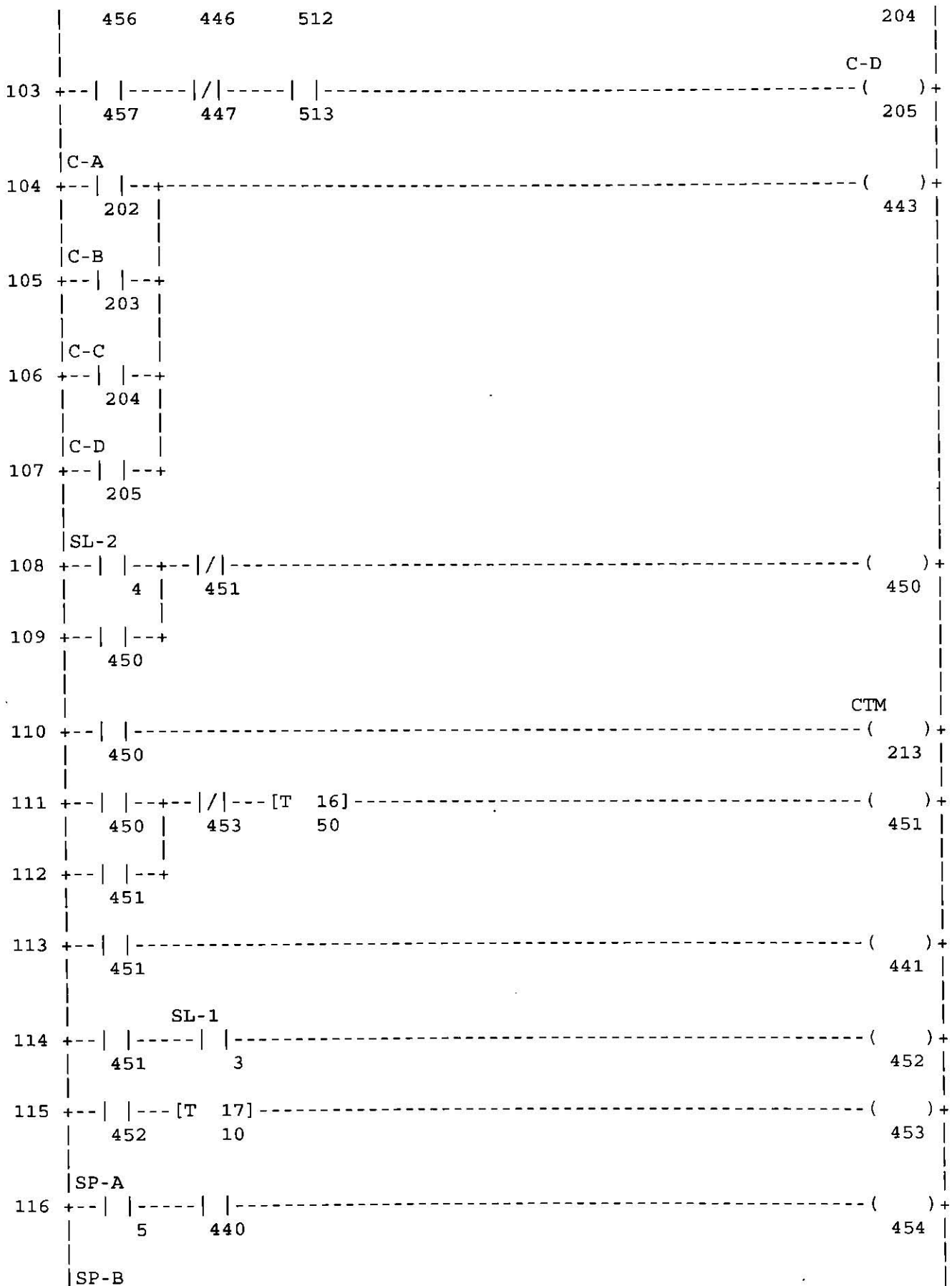




Automatización de una Tolva Móvil y Grafcet



Automatización de una Tolva Móvil y Grafset



Automatización de una Tolva Móvil y Grafset

117	+	---		-----		-----		-----	(	)	+		
				6		440				455			
				SP-C									
118	+	---		-----		-----		-----	(	)	+		
				7		440				456			
				SP-D									
119	+	---		-----		-----		-----	(	)	+		
				0		440				457			
120	+	-----									(JEND)	+	
121	+	-----									(MCR)	+	
				F-1		F-2				ERROR			
122	+	---		/		-----		/		-----	(	)	+
				10		11				210			
				F-3									
123	+	---		-----		-----		-----	(	)	+		
				12									
124	+	---		/		-----		/		-----	(	)	+
				520									
				REVERS									
				SN-A A									
125	+	---		-----		/		-----	(	)	+		
				13		200				510			
				REVERSA									
				SN-B									
126	+	---		-----		/		-----	(	)	+		
				14		200				511			
				REVERSA									
				SN-C									
127	+	---		-----		/		-----	(	)	+		
				15		200				512			
				REVERSA									
				SN-D									
128	+	---		-----		/		-----	(	)	+		
				16		200				513			
				SN-A	SN-B	SN-C	SN-D						
129	+	---		-----		-----		-----		-----	(	)	+
				13	14	15	16			520			
130	+	-----									(END)	+	

**CODIFICACIÓN DEL PROYECTO**

00000	LOD	2	00041	OUT	405
00001	OUT	703	00042	LOD	416
00002	LOD	12	00043	OR	406
00003	LOD N	10	00044	AND	405
00004	AND N	11	00045	TIM	2
00005	OR LOD		00046		50
00006	OR N	520	00047	OUT	406
00007	OUT	503	00048	LOD	417
00008	LOD N	503	00049	OR	407
00009	MCS		00050	AND	406
00010	LOD N.	500	00051	TIM	3
00011	JMP		00052		50
00012	LOD	1	00053	OUT	407
00013	OR	400	00054	LOD	414
00014	AND N	412	00055	AND N	404
00015	OUT	400	00056	AND	510
00016	LOD	400	00057	OUT	202
00017	AND N	3	00058	LOD	415
00018	AND N	402	00059	AND N	405
00019	OUT	200	00060	AND	511
00020	LOD	400	00061	OUT	203
00021	AND	3	00062	LOD	416
00022	OR	402	00063	AND N	406
00023	AND N	401	00064	AND	512
00024	OUT	402	00065	OUT	204
00025	LOD	402	00066	LOD	417
00026	AND N	410	00067	AND N	407
00027	AND N	403	00068	AND	513
00028	OUT	201	00069	OUT	205
00029	LOD	414	00070	LOD	202
00030	AND N	200	00071	OR	203
00031	OR	404	00072	OR	204
00032	AND N	413	00073	OR	205
00033	TIM	0	00074	OUT	403
00034		50	00075	LOD	4
00035	OUT	404	00076	OR	410
00036	LOD	415	00077	AND N	411
00037	OR	405	00078	OUT	410
00038	AND	404	00079	LOD	410
00039	TIM	1	00080	OUT	213
00040		50	00081	LOD	410



00082	OR	411	00127	OR	420
00083	AND N	413	00128	AND N	432
00084	TIM	4	00129	OUT	420
00085		50	00130	LOD	420
00086	OUT	411	00131	AND N	3
00087	LOD	411	00132	AND N	422
00088	OUT	401	00133	OUT	200
00089	LOD	411	00134	LOD	420
00090	AND	3	00135	AND	3
00091	OUT	412	00136	OR	422
00092	LOD	412	00137	AND N	421
00093	TIM	5	00138	OUT	422
00094		10	00139	LOD	422
00095	OUT	413	00140	AND N	430
00096	LOD	5	00141	AND N	423
00097	AND	400	00142	OUT	201
00098	OUT	414	00143	LOD	554
00099	LOD	6	00144	AND N	200
00100	AND	400	00145	OR	424
00101	OUT	415	00146	AND N	433
00102	LOD	7	00147	TIM	6
00103	AND	400	00148		20
00104	OUT	416	00149	OUT	424
00105	LOD	0	00150	LOD	555
00106	AND	400	00151	OR	425
00107	OUT	417	00152	AND	424
00108	JEND		00153	TIM	7
00109	LOD	10	00154		30
00110	AND N	11	00155	OUT	425
00111	OUT	500	00156	LOD	556
00112	LOD N	10	00157	OR	426
00113	AND	11	00158	AND	425
00114	OUT	501	00159	TIM	8
00115	LOD	10	00160		40
00116	AND	11	00161	OUT	426
00117	OUT	502	00162	LOD	557
00118	LOD	400	00163	OR	427
00119	OR	420	00164	AND	426
00120	OR	440	00165	TIM	9
00121	OUT	211	00166		50
00122	LOD N	211	00167	OUT	427
00123	OUT	212	00168	LOD	554
00124	LOD N	501	00169	AND N	424
00125	JMP		00170	AND	510
00126	LOD	1	00171	OUT	202

00172	LOD	555	00217	AND	420
00173	AND N	425	00218	OUT	556
00174	AND	511	00219	LOD	0
00175	OUT	203	00220	AND	420
00176	LOD	556	00221	OUT	557
00177	AND N	426	00222	JEND	
00178	AND	512	00223	LOD N	502
00179	OUT	204	00224	JMP	
00180	LOD	557	00225	LOD	1
00181	AND N	427	00226	OR	440
00182	AND	513	00227	AND N	452
00183	OUT	205	00228	OUT	440
00184	LOD	202	00229	LOD	440
00185	OR	203	00230	AND N	3
00186	OR	204	00231	AND N	442
00187	OR	205	00232	OUT	200
00188	OUT	423	00233	LOD	440
00189	LOD	4	00234	AND	3
00190	OR	430	00235	OR	442
00191	AND N	431	00236	AND N	441
00192	OUT	430	00237	OUT	442
00193	LOD	430	00238	LOD	442
00194	OUT	213	00239	AND N	450
00195	LOD	430	00240	AND N	443
00196	OR	431	00241	OUT	201
00197	AND N	433	00242	LOD	454
00198	TIM	10	00243	AND N	200
00199		50	00244	OR	444
00200	OUT	431	00245	AND N	453
00201	LOD	431	00246	TIM	12
00202	OUT	421	00247		50
00203	LOD	431	00248	OUT	444
00204	AND	3	00249	LOD	455
00205	OUT	432	00250	OR	445
00206	LOD	432	00251	AND	444
00207	TIM	11	00252	TIM	13
00208		10	00253		40
00209	OUT	433	00254	OUT	445
00210	LOD	5	00255	LOD	456
00211	AND	420	00256	OR	446
00212	OUT	554	00257	AND	445
00213	LOD	6	00258	TIM	14
00214	AND	420	00259		30
00215	OUT	555	00260	OUT	446
00216	LOD	7	00261	LOD	457

Automatización de una Tolva Móvil y Grafcet

00262	OR	447	00307		10
00263	AND	446	00308	OUT	453
00264	TIM	15	00309	LOD	5
00265		20	00310	AND	440
00266	OUT	447	00311	OUT	454
00267	LOD	454	00312	LOD	6
00268	AND N	444	00313	AND	440
00269	AND	510	00314	OUT	455
00270	OUT	202	00315	LOD	7
00271	LOD	455	00316	AND	440
00272	AND N	445	00317	OUT	456
00273	AND	511	00318	LOD	0
00274	OUT	203	00319	AND	440
00275	LOD	456	00320	OUT	457
00276	AND N	446	00321	JEND	
00277	AND	512	00322	MCR	
00278	OUT	204	00323	LOD N	10
00279	LOD	457	00324	AND N	11
00280	AND N	447	00325	OR	12
00281	AND	513	00326	OR N	520
00282	OUT	205	00327	OUT	210
00283	LOD	202	00328	LOD	13
00284	OR	203	00329	AND N	200
00285	OR	204	00330	OUT	510
00286	OR	205	00331	LOD	14
00287	OUT	443	00332	AND N	200
00288	LOD	4	00333	OUT	511
00289	OR	450	00334	LOD	15
00290	AND N	451	00335	AND N	200
00291	OUT	450	00336	OUT	512
00292	LOD	450	00337	LOD	16
00293	OUT	213	00338	AND N	200
00294	LOD	450	00339	OUT	513
00295	OR	451	00340	LOD	13
00296	AND N	453	00341	AND	14
00297	TIM	16	00342	AND	15
00298		50	00343	AND	16
00299	OUT	451	00344	OUT	520
00300	LOD	451			
00301	OUT	441			
00302	LOD	451			
00303	AND	3			
00304	OUT	452			
00305	LOD	452			
00306	TIM	17			

**BIBLIOGRAFÍA**

**Programable Controllers (selected applications vol. 1)**

L.A. Bryan

E.A. Bryan

Industrial Text Publishing

**Programable Controllers and Designing Sequential Logic**

Robert Filer & George Leinonen

Saunders College Publishing

**Servicio de Compact Disk del Banco de Informacion Dialog**

**Servicio de Compact Disk IPO de la IEEE**

Centro de Información de Negocios del ITESM Campus Monterrey.

**IEEE Control Systems**

Vol. 13, Número 2, Abril 1993

**IEEE Transactions on Control Systems Technology**

Vol. 3, Número 3, Septiembre 1995

**Control Lógico Industrial**

J.J. Rodríguez

ITESM

**MICRO - 1, Micro Programmable Controller Users Manual**

Idec Corporation

