Final Sistemas Informáticos Industriales

Programación de PLC's mediante Sequential Function Charts

Alumno: ***Santiago Carello***

Fecha: ***4/6/2020***

ÍNDICE

1) ¿Qué es un PLC?

Un CONTROLADOR LÓGICO PROGRAMABLE (PLC) es un sistema de control industrial por computadora que monitorea continuamente el estado de los dispositivos de entrada y toma decisiones basadas en un programa personalizado para controlar el estado de los dispositivos de salida.

Casi cualquier línea de producción, función de máquina o proceso puede mejorarse enormemente utilizando este tipo de sistema de control. Sin embargo, el mayor beneficio de usar un PLC es la capacidad de cambiar y replicar la operación o proceso mientras se recolecta y comunica información vital.

Otra ventaja de un sistema PLC es que es modular. Es decir, puede mezclar y combinar los tipos de dispositivos de entrada y salida que mejor se adapten a su aplicación.

2) ¿Cómo funciona un PLC?

El PLC recibe información de los sensores o dispositivos de entrada conectados, procesa los datos y activa las salidas basándose en parámetros preprogramados.

Dependiendo de las entradas y salidas, un PLC puede monitorear y registrar datos de tiempo de ejecución como la productividad de la máquina o la temperatura de funcionamiento, iniciar y detener automáticamente los procesos, generar alarmas si una máquina no funciona correctamente, y más. Los controladores lógicos programables son una solución de control flexible y robusta, adaptable a casi cualquier aplicación

Hay algunas características clave que distinguen a los PLCs de los PCs industriales, microcontroladores y otras soluciones de control industrial:

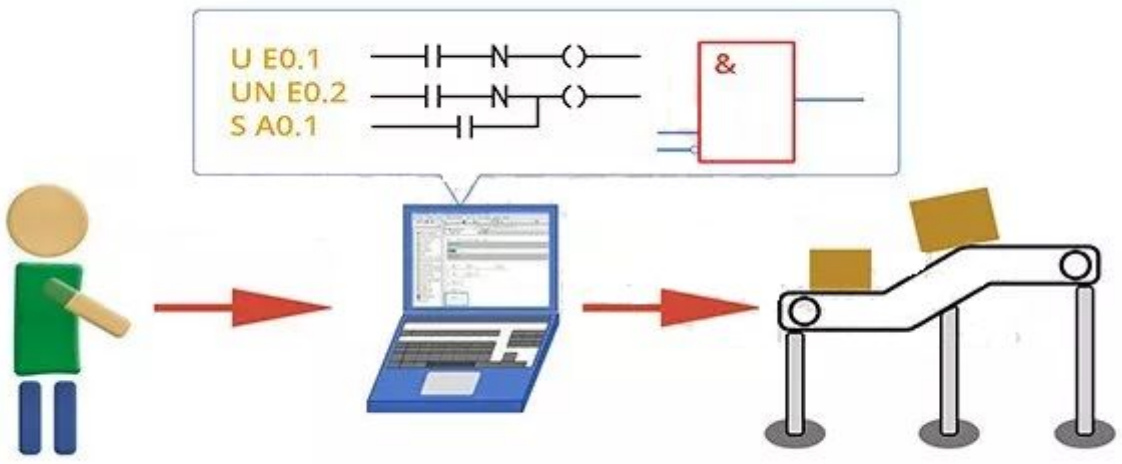
- I/O. La CPU del PLC almacena y procesa los datos del programa, pero los módulos de entrada y salida conectan el PLC con el resto de la máquina; estos módulos de I/O son los que proporcionan información a la CPU y activan resultados específicos. Las E/S pueden ser analógicas o digitales; los dispositivos de entrada pueden incluir sensores, interruptores y medidores, mientras que las salidas pueden incluir relés, luces, válvulas y variadores. Los usuarios pueden mezclar y combinar las E/S de un PLC para obtener la configuración adecuada para su aplicación.

- Comunicaciones. Además de los dispositivos de entrada y salida, un PLC también puede necesitar conectarse con otros tipos de sistemas; por ejemplo, los usuarios pueden querer exportar los datos de la aplicación registrados por el PLC a un sistema de control de supervisión y adquisición de datos (SCADA), que monitorea múltiples dispositivos conectados. Los PLCs ofrecen una gama de puertos y protocolos de comunicación para asegurar que el PLC pueda comunicarse con estos otros sistemas.

- HMI. Para poder interactuar con el PLC en tiempo real, los usuarios necesitan una HMI o Interfaz Hombre-Máquina. Estas interfaces de operador pueden ser pantallas simples, con lectura de texto y teclado, o grandes pantallas táctiles más parecidas a las de la electrónica de consumo, pero de cualquier manera, permiten a los usuarios revisar e introducir información en el PLC en tiempo real.

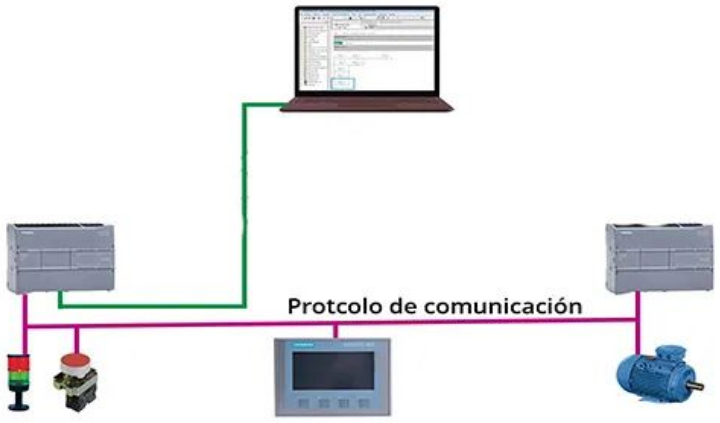
3) Lenguajes de programación de un PLC

3.1) ¿Qué son?



Los lenguajes de programación de PLC son símbolos, caracteres y reglas de uso que fueron diseñados para poder tener una comunicación de los usuarios con las máquinas. Gracias a este vínculo, podemos ser capaces de crear un programa con instrucciones para controlar el funcionamiento de cualquier proceso o máquina.

3.2) ¿Cómo se comunican 2 o más PLC's?



La comunicación entre dos o más plc's sucede a través de una conexión especial a base de reglas, que permiten la transferencia de datos o información entre cada uno de estos. A este tipo de reglas se les conoce como "protocolo de comunicación", algunos de estos son: Profibus, Fieldbus, Modbus, Devicenet, Interbus, entre algunos otros.

3.3) Tipos de lenguaje de programación de PLC

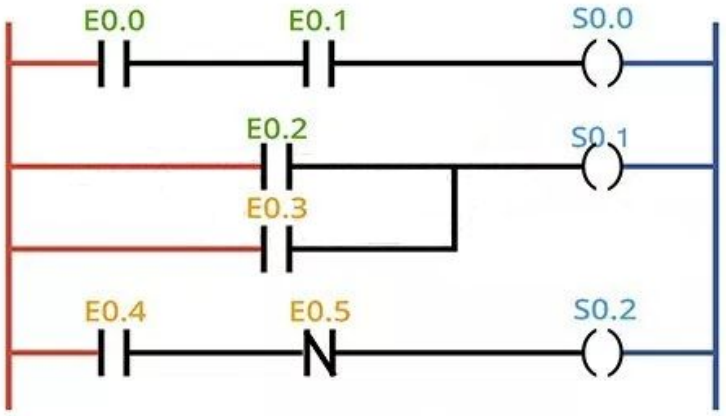
Existe diversidad dentro de los lenguajes de programación debido a que los usuarios tienen diferente formación en diferentes ramas de la ingeniería, por ejemplo los ingenieros o técnicos eléctricos y electrónicos están acostumbrados a utilizar símbolos en los diagramas eléctricos, mientras que los ingenieros en sistemas siempre utilizan lenguajes escritos, por lo cual unos prefieren programar un lenguaje más visual y otros prefieren un lenguaje escrito. Todo esto fue estandarizado por la norma IEC 61131-3 , al igual que el método SFC que tiene estrecha relación con ellos, del cual hablaré en el próximo capítulo.

Los lenguajes de programación de PLC se pueden clasificarse en dos clases, lenguajes de alto y bajo nivel cada uno con diferentes tipos.

3.3.1) Lenguajes de alto nivel

En esta categoría se encuentran los lenguajes que son gráficos, ya que estos utilizan una interfaz de símbolos para declarar las instrucciones de control, una de las desventajas de estos lenguajes visuales es que la programación está limitada a los símbolos que se proporcionan.

3.3.1.1) Diagrama de escalera o Ladder

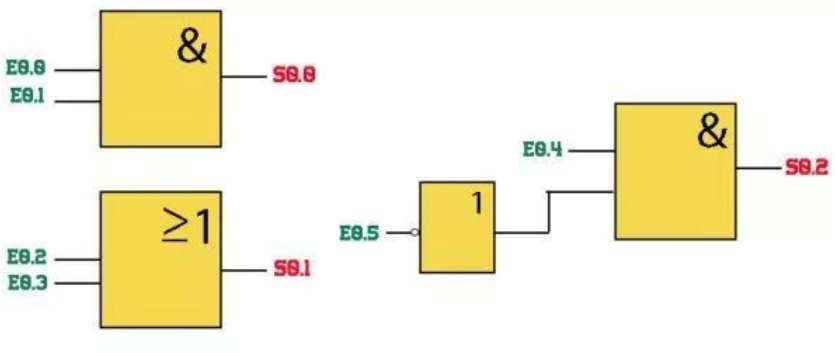


Este lenguaje fue uno de los pioneros ya que fue uno de los primeros en ser utilizados, ya que se asemeja mucho a los diagramas con relevadores. Se le llama de escalera por que es muy similar a la estructura de una escalera, ya que contiene dos rieles verticales, y varios rieles horizontales (en este caso serían los escalones).

Características principales:

- Los 2 rieles verticales son de alimentación (en el caso de VCD uno es voltaje y otro tierra y en VCA son L1 y L2)
- Las instrucciones se colocan del lado izquierdo
- Las salidas siempre se colocan del lado derecho.
- Se pueden colocar varias instrucciones o varias salida en paralelo.
- El procesador del PLC interpreta los datos de arriba hacia a abajo y de izquierda a derecha

3.3.1.2) Diagrama de bloques



En este tipo de programación se utilizan bloques de símbolo lógicos. Las salidas no se requieren incorporar a una bobina de salida, por que la salida esta asignada en las salidas de los bloques lógicos. Estos diagramas en su mayoría son preferidos por personas acostumbrados a trabajar con circuitos de compuertas lógicas, ya que la simbología utilizada es equivalente.

Características principales:

- Las salidas de los bloques no se conectarán entre sí.
- La evaluación de una red se termina antes de iniciar la siguiente

3.3.2) Lenguajes de bajo nivel

En este tipo se encuentran los lenguajes de programación a través de texto, utilizando cadenas de caracteres para indicar las instrucciones de control.

3.3.2.1) Lista de instrucciones

Este tipo de lenguaje es el más antiguo y es la base para todos los lenguajes de programación que existen, este lenguaje es el precursor del diagrama escalera ya que este se utilizaba cuando las computadoras aún no tenían capacidad gráfica. Todos los lenguajes son traducidos a lista de instrucciones.

Características principales:

- Todos los lenguajes pueden ser traducidos a lista de instrucciones, pero no al revés.
- La programación es más compacta.
- Este lenguaje es el más completo de todos.

3.3.2.2) Texto estructurado

```
IF ((E0.0 == TRUE) && (E0.1 == TRUE))
{
S0.0 = TRUE;
}
ELSE S0.0= FALSE;

IF ((E0.2 == TRUE) || (E0.3 ==TRUE))
{
S0.1 = TRUE;
}
ELSE S0.1 = FALSE;

IF ((E0.4 == TRUE) && (E0.5 == FALSE))
{
S0.2 = TRUE;
}
ELSE S0.2 = FALSE;
```

El texto estructurado se compone de una serie de instrucciones que se pueden ejecutar, como sucede con los lenguajes superiores, de forma condicionada. Este lenguaje es muy similar al lenguaje C y sobre todo a PASCAL

Características principales:

- Trata indistintamente las mayúsculas y las minúsculas
- Soporta instrucciones aritméticas complejas.
- Soporta ciclos de iteración (repeat – until, while – do)

4) Sequential Function Chart (SFC)

4.1) Introducción

SFC es un formalismo gráfico muy expresivo de la norma IEC 61131-3 que deriva de GRAFCET y de las Redes de Petri. No se considera un lenguaje de programación ya que necesita otros lenguajes para expresar las condiciones de transición y las acciones.

- SFC describe el comportamiento secuencial del programa de control.
- SFC estructura la organización interna de un programa y ayuda a descomponer un control

problema en partes manejables, manteniendo la visión general.
- SFC permite hacer mas sencillo un problema, diviendolo en partes o modulos.
- SFC facilita el rápido diagnóstico de problemas y las tareas de mantenimiento.

Software modular y el rol del SFC

El prerequisito principal para mejorar la práctica de programación actual y crear una mayor productividad y una mejor calidad del producto resultante es la modularidad del software.

La modularidad del software significa que un programa de software debe organizarse en partes acopladas y cada una de ellas debe desarrollarse y testearse independientemente de las demás. Para esto, el proceso de desarrollo de software del PLC debe basarse en una metodología adecuada que, en mi opinión, debe basarse en dos supuestos fundamentales:

- Centrarse en el diseño. El peso de la fase de diseño debe aumentar, de modo que la fase de desarrollo comienza sólo después de una definición clara de la estructura del programa de control y las interacciones entre sus partes.
- Centrarse en el estándar. Las potencialidades de los lenguajes estándar IEC 61131 deben ser plenamente explotado en las fases de diseño y desarrollo, ya que pueden cubrir la mayor parte de las necesidades del programador.

SFC es un lenguaje adecuado (en IEC 61131-3) para soportar las fases iniciales y más cruciales del ciclo de vida de desarrollo de software para PLC. Algunas razones son:

-Alto poder expresivo: El lenguaje SFC tiene el mismo potencial expresivo que los diagramas de estado y es comparable a las redes de Petri para enfrentar problemas concurrentes. Diagramas de estado y las redes de Petri se consideran las herramientas más apropiadas para modelos dinámicos y son ampliamente utilizados en muchos campos. Por lo tanto, podemos decir que el lenguaje SFC es intrínsecamente capaz de modelar el comportamiento de un sistema.

-Formalismo gráfico: SFC no es el único lenguaje con gráficos primitivos disponibles por el estándar, pero con respecto a los otros dos lenguajes gráficos (LD y FBD) este ofrece características de nivel superior para describir las dinámicas del sistema. Gracias a su sintaxis gráfica, es muy fácil de aprender y usar. Además, resulta especialmente adecuado para representar el proceso en diferentes niveles de detalle.

-Soporte de diseño preliminar: SFC se puede utilizar directamente desde el principio, para dar una primera representación formal del comportamiento del sistema. Por lo tanto, es una herramienta importante en el análisis preliminar y las primeras fases de diseño, cuando muchos aspectos todavía no está bien definido o incluso desconocido para el diseñador. Usando SFC evitamos agregar la ambigüedad de la descripción del lenguaje natural a las especificaciones aproximadas disponibles para el diseñador. De esta manera, el número de malentendidos entre cliente, diseñador y los programadores se reducen sustancialmente.

-Soporta un diseño detallado: El esquema SFC producido en la fase inicial de diseño puede ser

progresivamente especificado y refinado a medida que se dispone de nueva información. Por lo tanto, el deseado nivel de detalle no es algo de lo que hay que preocuparse ya que se alcanza paso a paso.

-Conexión natural con los otros lenguajes: Es bastante evidente que el lenguaje SFC puede ser usado en combinación con los otros lenguajes del estándar, particularmente adecuado para describir detalles de control tales como condiciones de transición y acciones elementales. La posibilidad usar el lenguaje correcto en el momento correcto aumenta la eficiencia global del software de desarrollo y mejora el rendimiento del código ejecutable resultante.

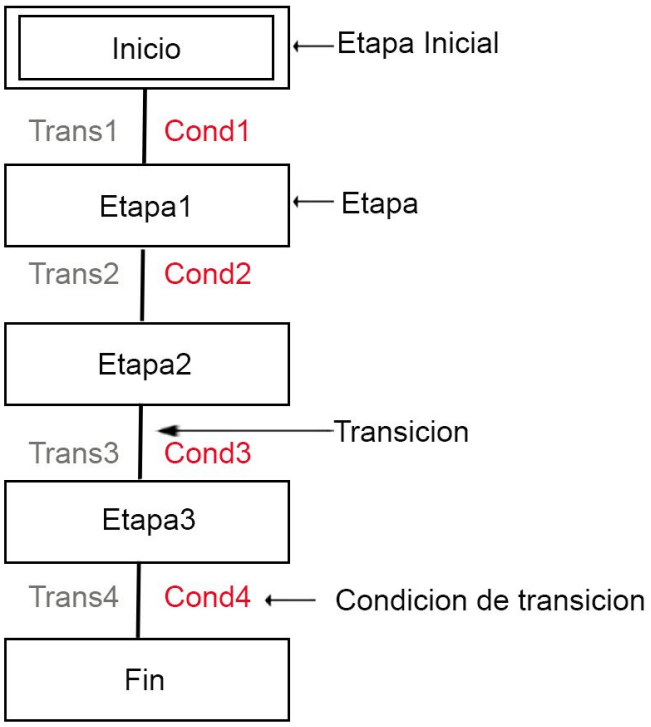
Por lo tanto , SFC es una buena elección porque:

- Cada estado del proceso se puede asignar claramente a un paso.
- La transición de un estado/paso a un estado/paso diferente/ siguiente puede expresarse por su condición correspondiente.
- La relación de estados/pasos y el flujo de actividad de un estado a otro es visualizado/programado (generalmente) de manera gráfica y fácil de entender debido a reglas subyacentes para la evolución de los estados activos de los pasos. En consecuencia, el gráfico SFC refleja el diseño del proceso. Este beneficio se aplica a todas las fases del ciclo de vida del software del PLC: planificación, diseño (incluida la codificación, depuración y prueba), puesta en marcha y mantenimiento.
- Las acciones a realizar se crean independientemente de los estados / pasos y se combinan utilizando bloques de acción (y los calificadores de acción correspondientes).

4.2) Estructura

Una secuencia en SFC se compone de una serie de etapas representadas por cajas rectangulares conectadas entre sí por líneas verticales.

Cada etapa representa un estado particular del sistema.
Cada línea vertical representa a su vez una transición.
Una transición está asociada a una condición de “true/false”, lo cual da paso a la desactivación de la etapa que la precede y activación de la posterior.
Por ejemplo:



4.3) Elementos

Elementos soportados del estándar SFC:
– Etapa
– Transición
– Acción
– Divergencia
– Convergencia
– Divergencia simultánea
– Convergencia simultánea

4.3.1) Etapa

Etapa:
– Estado del sistema
– Una etapa puede estar sólo en dos estados:
• Activa
• No activa (inactiva)

Tipos de etapa:
– Etapa normal
– Etapa inicial: Aquella que queda activada al comienzo del algoritmo de control. Se representa con doble recuadro.

4.3.2) Transición

Transición:
– Representa la condición que da paso del control de una o más etapas que la preceden a una o más etapas que figuren a continuación
– Está representada por una línea horizontal que cruza la unión entre etapas
– El resultado de la condición da como resultado una expresión booleana.

4.3.3) Acción

Cero, una o más acciones pueden estar asociadas con cada etapa o paso. Un paso sin ninguna acción funciona como una función de "Wait" que solo espera que la la siguiente condición de transición se vuelva verdadera.
La declaración de una acción consiste en el nombre de la acción (de tipo string) y el cuerpo de la acción. El cuerpo de la acción puede ser una variable booleana, una colección de instrucciones en IL (Lista de Instrucciones), una colección de declaraciones en ST (texto estructurado), una colección de peldaños en LD(en ladder), una colección de redes en FBD (diagrama de bloques) o un SFC en su momento.

El control de acciones está definido por calificadores de acción. Los posibles calificadores de acción son los enumerados en la siguiente tabla. Los calificadores especifican la ejecución de acciones en cada ciclo de ejecución en relación a los estados de sus pasos asociados. Las acciones que utilizan el calificador N o ninguno se ejecutan siempre que sus pasos asociados están activos. Los calificadores L, D, SD, DS y SL requieren un adicional parámetro asociado de tipo TIME para el retraso o la limitación.

Condición	Descripción
Ninguna	No almacenado
N	No almacenado
R	Reset
S	Set
L	Tiempo limitado
D	Tiempo de retardo
P	Pulso
SD	Almacenado y tiempo de retardo
DS	Tiempo de retardo y almacenado
SL	Almacenado y tiempo limitado

P1	Pulso (flanco ascendente)
P0	Pulso (flanco descendente)

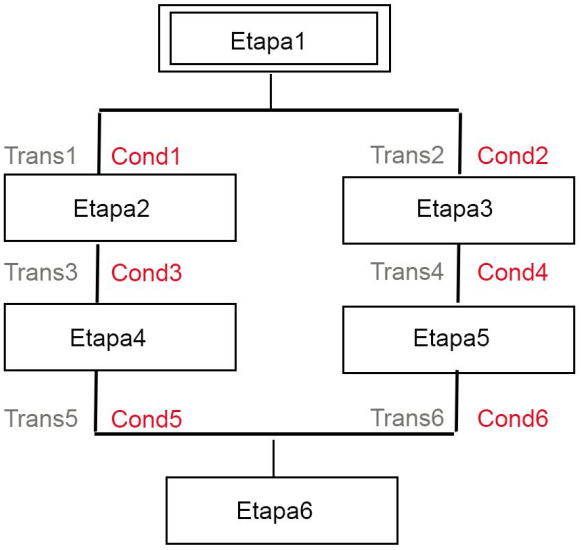
4.3.4) Reglas de evolución

En SFC existirá siempre una etapa inicial, que tiene doble recuadro. Esta se activa con la puesta en marcha del sistema. Luego el gráfico evolucionará con las siguientes reglas:
-Una transición se franqueará cuando:

1) La etapa anterior a la misma esté activa.
2) La condición asociada a la misma se cumpla.

Como ya dije, entre dos etapas hay una transición. A cada transición le corresponde una condición que se ha de cumplir para poder pasar la transición. Una transición es válida cuando la etapa inmediatamente anterior a ella está activa. Cuando una transición es válida y su condición se cumple se dice que la la transición es franqueable.
Al franquear una transición se desactivan sus etapas anteriores y se activan las posteriores.

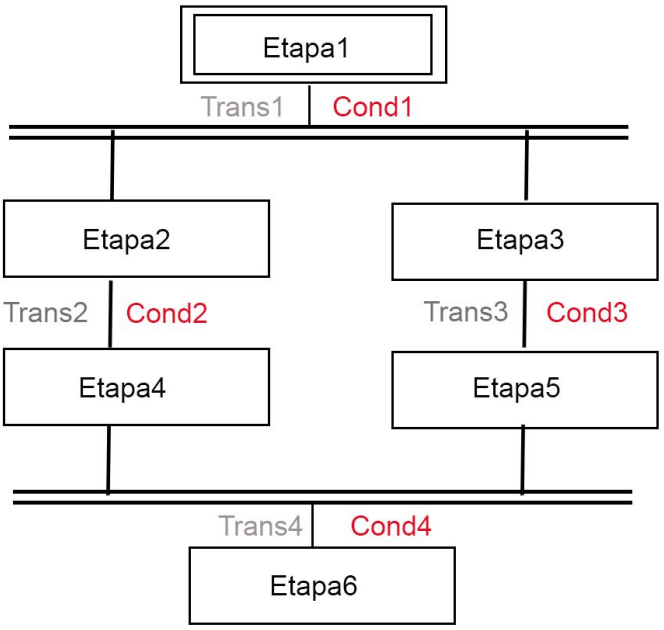
4.3.5) Divergencia y convergencia



Divergencia:
– Estando activa la etapa1 se pasa a la etapa2 o a la etapa3 según este activa Trans1 o Trans2

Convergencia
– Para pasar a la etapa Etapa6 debe estar activa la etapa Etapa4 y cumplirse la condición5 o estar activa la etapa Etapa5 y cumplirse la condición6

4.3.6) Divergencia y convergencia simultánea



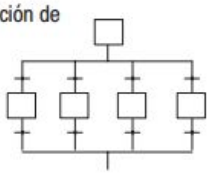
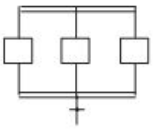
Divergencia simultánea
– Estando activa la etapa1 al verificarse la condición1 se pasa simultáneamente a las etapas 2 y 3

Convergencia simultánea
– Si las etapas 4 y 5 están activas simultáneamente y se cumple la condición 4se pasa a la etapa 6.

4.4) Decidiendo cómo usar SFC

Después de haber identificado las áreas principales de operación de la máquina, se convierten los pasos y caminos lógicos que se identificaron en la especificación de diseño a bloques de constitución de SFC. La Tabla siguiente ayuda a explicar cuándo usar diferentes bloques de constitución de SFC.

En este momento no hay que preocuparse sobre la lógica real para cada paso y transición. Después de terminar el SFC, se puede desarrollar la lógica.

Si tiene:	Entonces dibuje:	Usando estas reglas:
Un estado de máquina independiente	Un paso con su transición 	Un paso siempre debe ser seguido por una transición.
Una cadena de sucesos definida claramente que ocurre secuencialmente Por ejemplo, en una área de tratamiento al calor, la temperatura debe subir a una velocidad particular, manteniendo la temperatura durante un cierto tiempo y luego enfriar a una velocidad particular.	Un camino simple de pasos y transiciones 	Para propósitos de diseño, numere los pasos y transiciones consecutivamente a partir del 2. Empiece el camino con un paso, termine el camino con una transición.
Dos o más caminos alternativos donde sólo uno es seleccionado Por ejemplo, dependiendo de un código de constitución, una estación debe perforar o pulir.	Una bifurcación de selección 	Las transiciones que empiezan cada camino son escaneadas de izquierda a derecha. La primera transición verdadera determina el camino tomado.
Dos o más caminos paralelos que deben ser escaneados simultáneamente por lo menos una vez Por ejemplo, las comunicaciones y transferencias en bloques deben ocurrir mientras la lógica de control está ejecutando.	Una bifurcación simultánea 	Todos los caminos están activos en la estructura. Usted puede definir hasta 7 caminos paralelos.

4.5) SFC y Redes De Petri

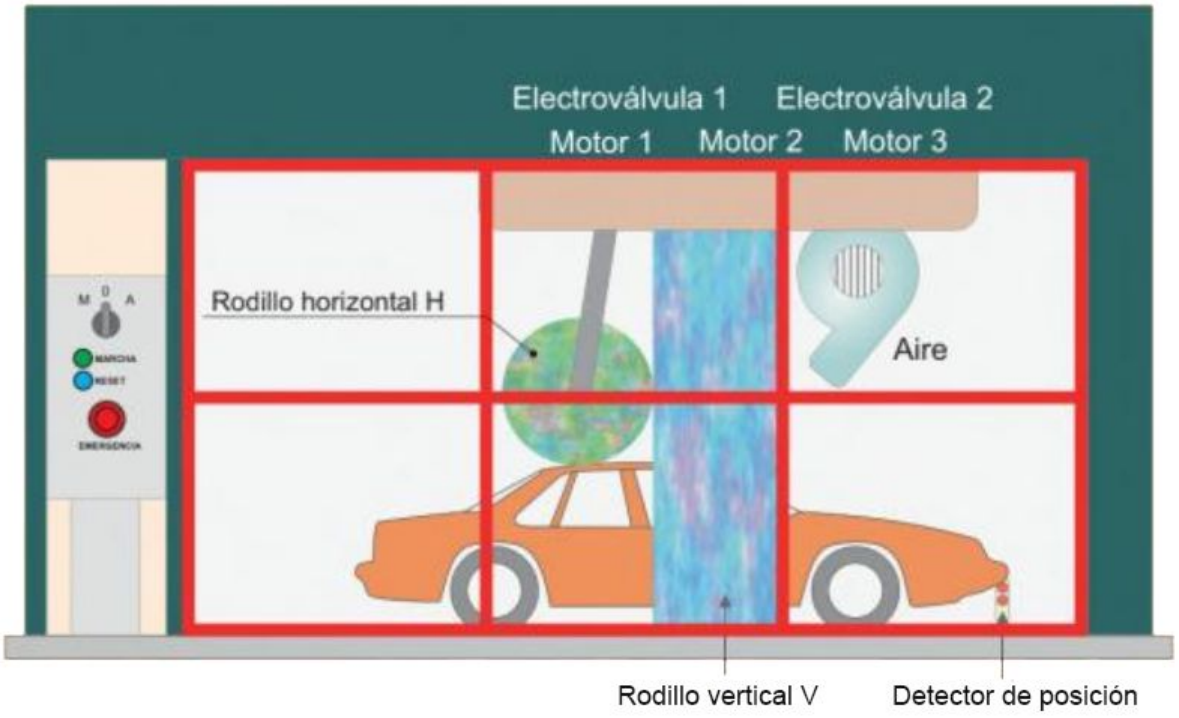
La presencia simultánea, en tiempo de ejecución, de dos o más pasos activos sugiere la posibilidad de establecer una analogía entre esquemas SFC (con secuencias paralelas) y las redes de Petri. Las redes de petri son una herramienta de modelado conocida y apreciada, particularmente adecuada para representar el comportamiento de sistemas concurrentes. Una red de Petri proporciona una descripción formal del sistema que hace riguroso su análisis y el estudio de su comportamiento.

En un autómata, los estados del sistema están predefinidos y el sistema pasa de un estado a otro de tal manera que, en todo momento, solo un estado esté activo. Por el contrario, en las redes de petri el estado es un concepto distribuido y las transiciones es un concepto local: en cada transición que ocurre en la red, el estado en algunos lugares cambia mientras que en otros no se ven afectados. Por esta razón, las redes de Petri son adecuadas para modelar sistemas asincrónicos, donde los eventos impactan en diferentes puntos y no ocurren según una secuencia predefinida. Además las redes de Petri tienen una falta sustancial con respecto al enfoque SFC: comportamiento no determinista.

Por estas razones, no considere a los esquemas SFC como variantes de las redes de Petri. Las redes de Petri dan sugerencias interesantes para modelar sistemas dinámicos asíncronos. Sin embargo, forzar la analogía con el lenguaje SFC podría ser engañoso y, después de todo, negativo. En cambio, nosotros creemos que diseñar la arquitectura de control como un diagrama de estado jerárquico es muy útil para Programador PLC.

4.6) Ejemplo

Máquina para el lavado de autos



En la máquina tenemos seis motores: dos que mueven un rodillo de gamuza horizontal, haciéndolo rotar y desplazándolo; otros dos para los rodillos verticales; otro más para accionar un ventilador de aire y otro para el desplazamiento de este. La acción de dos electroválvulas permite el enjabonado y el aclarado del auto, respectivamente, mediante el flujo de jabón líquido y agua. El automatismo debe estar compuesto por diversos temporizadores que darán paso a las distintas etapas. La descripción del funcionamiento es:

-Inicialmente el sistema está en reposo, si se acciona el pulsador de marcha S1 y está activado el detector de posición S2 y no se ha disparado la protección de ninguno de los motores F1,F2,F3, entonces comienza a la vez el enjabonado y el movimiento de los rodillos verticales V con rotación izquierda. A la vez empieza a bajar el rodillo horizontal H hasta que activa el final de carrera S3, que se pone en marcha girando a izquierda.
-Una vez que se ha acabado el enjabonado y los rodillos han llegado a su fin, empieza el aclarado con el movimiento de todos los rodillos en sentido contrario.
-Una vez que el aclarado ha terminado, y los rodillos han llegado a su posición original (rodillo H activando el final de carrera S4), empieza el secado mediante la marcha del ventilador y su desplazamiento en un sentido, y cuando llegue al tope, en sentido contrario.
-Una vez que el ventilador termine su recorrido y llegue a su posición original, todo se para.

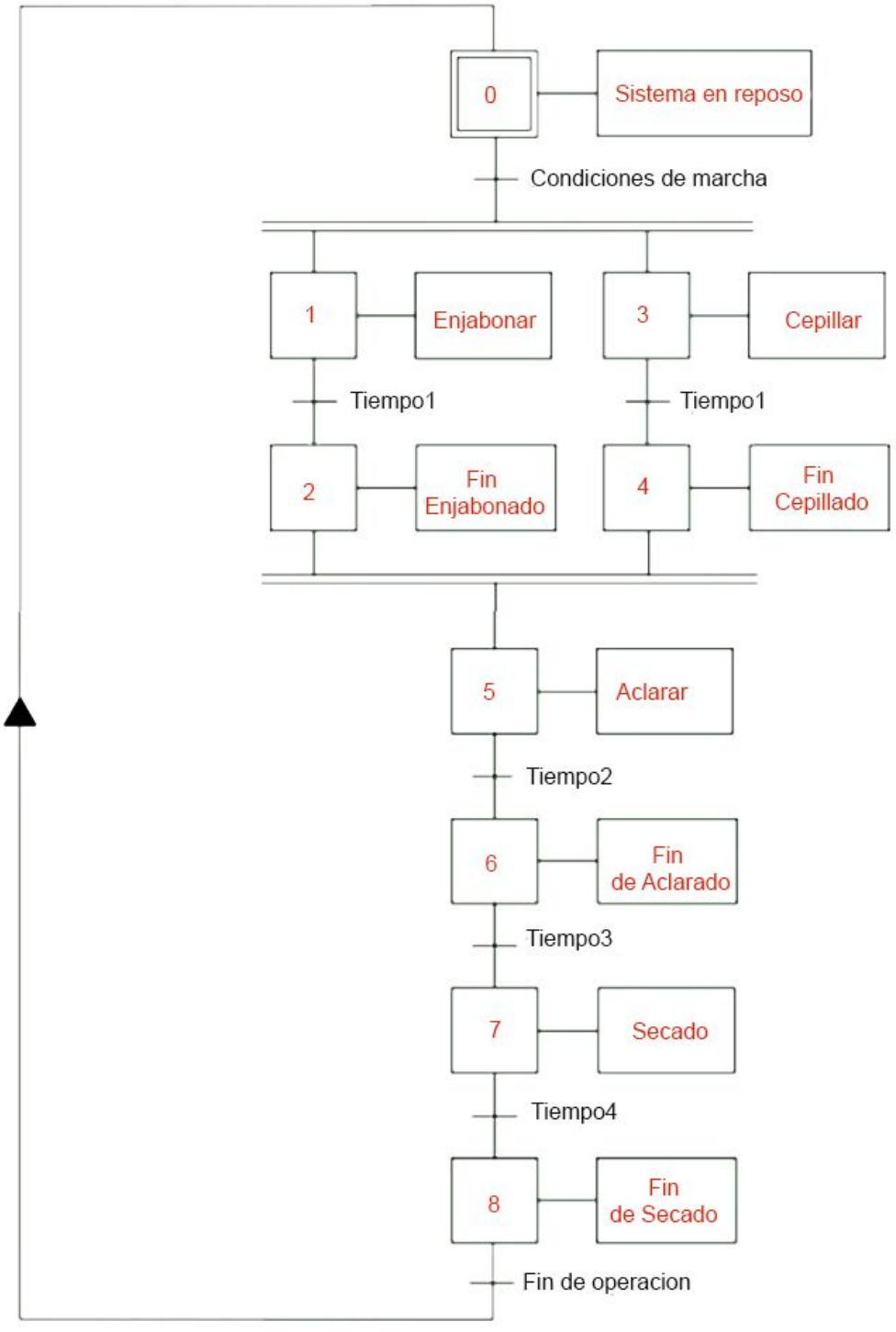
Podemos resolverlo con 3 niveles de complejidad y detalle distintos:

SFC de nivel 1, descripción funcional: En el primer nivel interesa una descripción global (normalmente poco detallada) del automatismo que permita comprender rápidamente su función. Es el tipo de descripción que haríamos para explicar lo que queremos que haga la máquina a la persona que la ha de diseñar o el que utilizaríamos para justificar, a las personas con poder de decisión en la empresa, la necesidad de esta máquina. Este nivel no tiene que hacer referencia a la tecnología utilizada (eléctrica, neumática, hidráulica, etc), ni los elementos utilizados para las distintas operaciones.

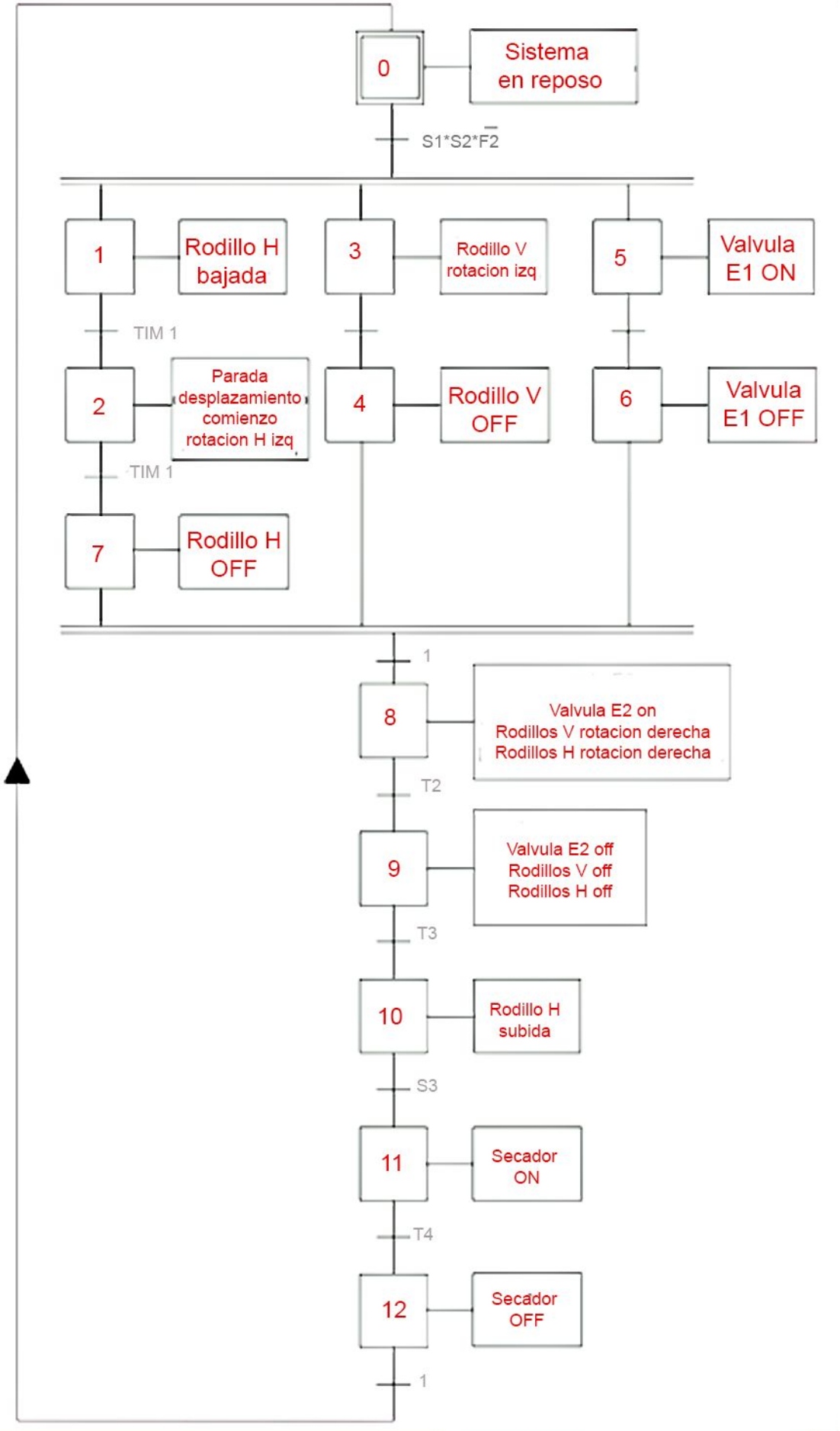
SFC de nivel 2, descripción tecnológica: en este nivel se hace una descripción a nivel tecnológico y operativo del automatismo. Quedan perfectamente definidas las diferentes tareas que han de realizar los elementos escogidos. En este nivel completamos la estructura de la maquina y nos falta el automatismo que controla.

SFC de nivel 3, descripción operativa: en este nivel se implementa el automatismo. El SFC definirá la secuencia de actuaciones que realizará este automatismo. En el caso de que se trate, por ejemplo, de un autómata programable, definirá la evolución del automatismo y la activación de las salidas en función de la evolución de las entradas. Para programar un SFC en un PLC, el nivel de descripción debe ser el adecuado, esto implica que las transiciones y las etapas incluyan código de los lenguajes de la norma IEC 6113-3.

Describiendo el funcionamiento secuencial de la máquina lavadora obtenemos el SFC de nivel 1, que queda de la siguiente manera:



De una manera más detallada, usando los nombres de las variables de entrada y las acciones de salida, se describe el SFC de nivel 2 en el siguiente gráfico:


0
Sistema en reposo
S1*S2*F2
1
Rodillo H bajada
3
Rodillo V rotacion izq
5
Valvula E1 ON
TIM 1
2
Parada desplazamiento comienzo rotacion H izq
4
Rodillo V OFF
6
Valvula E1 OFF
TIM 1
7
Rodillo H OFF
1
8
Valvula E2 on
Rodillos V rotacion derecha
Rodillos H rotacion derecha
T2
9
Valvula E2 off
Rodillos V off
Rodillos H off
T3
10
Rodillo H subida
S3
11
Secador ON
T4
12
Secador OFF
1

5) Fuentes

Las fuentes utilizadas fueron:

-Documentación de Rockwell Automation
-PLCOpen.org
-Sistemas secuenciales programables , por Antonio Garcia.
-InfoPLC.net
-IngMecaFenix.com