






Redes de Petri en la Automatización de una Máquina Tampográfica de Plásticos

Petri net in the automation of a pad printing machine for plastics

Y. García-Blanco , H. Ripoll-Sierra , H. Ripoll-Goenaga , J. Roldán-Mckinley , E. Yime-Rodríguez 

Resumen— Se documenta la implementación de la rutina de programación de una máquina para tampografía Comec-1998 usada para la impresión de piezas de plástico. El ciclo de trabajo comprende tres rutinas: batido de tinta, operación de ciclo sencillo y operación de ciclo continuo; programadas con tres cilindros, cuatro finales de carrera y dos temporizadores. Después de identificar los elementos para el ciclo de trabajo, se establecen las condiciones para generar los estados. Las transiciones en la red de Petri sirven para ir de un lugar a otro, usando la Automatización de las Redes de Petri y los conceptos de la Lógica de Pase de Tokens. Un PLC Siemens S7-200 se programó en programa KOP Ladder, dada la simplicidad y facilidad al pasar de la red de Petri al diagrama de lógica cableada.

Palabras claves— Comec LPE-60, KOP Ladder, Red de Petri, Siemens S7-200, Tampografía

Abstract— It is presented the implementation of the programming routine for a pad printing machine Comec-1998 used for printing plastic pieces. Three cylinders, four limit switch sensors and two timer routines were integrated in the work-cycle comprised of three sub-routines: ink shaking, single cycle operation and continuous cycle operation. After identifying all elements required for the work-cycle, the conditions for generating the states are set. The transitions in the Petri net were used for going from place to place, using the Automation Petri Nets and the Token Passing Logic concepts. A Siemens S7-200 PLC was programmed in KOP Ladder program, based on the simplicity and easiness when passing from Petri net to wired logic diagram.

Index terms— Comec LPE-60, KOP Ladder, Plastic pad printing, Petri net, Siemens S7-200

I. INTRODUCCIÓN

LA aparición de las redes de Petri como una manera organizada de establecer comunicación con los autómatas data del año 1962 [1], y su fundamentación teórica fue poco a poco siendo establecida [2,3] hasta que finalmente a partir de la

década de los 1980s empiezan a documentarse sus aplicaciones en la industria [4-8], llegando eventualmente a fusionarse en la aplicación con el lenguaje Ladder [9] y programación de PLC [10], con aplicaciones que pasaron de automatización de sistemas de corte de acero forjado [11] a, más recientemente, sistemas automatizados de manufactura [12, 13] y CNC [14]. Inclusive, se encuentran hoy día aplicaciones de las redes de Petri a la programación de robots [15], monitoreo de desempeño de satélites [16] y supervisión de sistemas interactivos humano-computador [17].

Un proceso de producción que no reporta aplicación de redes de Petri en la planeación de la automatización es el de la tampografía, un proceso de impresión relativamente nuevo [18] en donde se utiliza una placa metálica o plástica con un huecograbado recubierta de tinta, junto con un tampón de silicona que recoge la tinta de la placa para impregnarla por contacto sobre una pieza [19]. Como proceso de impresión, la tampografía abrió nuevas posibilidades frente a la impresión tradicional [20], nacida de la necesidad en las industrias relojeras suizas de realizar impresiones sobre piezas que requerían gran precisión, hasta llegar hoy en día a la impresión de piezas de gran complejidad [21] con máquinas de alta velocidad de accionamiento neumático y de motores lineales.

En este trabajo se presenta la implementación de las redes de Petri para la programación del ciclo de trabajo de una máquina tampográfica que era utilizada para marcar cepillos de plástico y madera. Las especificaciones de la máquina son: origen Italia, marca: Comec, modelo 1998, referencia Inkprint LPE-602C, funcionalidad de tampografado en superficies de plástico y de madera, con opción de doble tinta. Dada la antigüedad del equipo, era imposible conseguir el reemplazo de la tarjeta de control averiada, por lo cual se optó por implementar un PLC para la recuperación. La metodología escogida para el

Este manuscrito fue enviado el 03 de mayo de 2018 y aceptado el 15 de febrero de 2019.

Este trabajo fue soportado por la Vicerrectoría de Investigaciones de la Universidad del Atlántico (Puerto Colombia, Colombia).

Y. García-Blanco es Ingeniero Mecánico egresado de la Universidad del Atlántico (Puerto Colombia, Colombia). (e-mail: yjgarcia.blanco@gmail.com).

H. Ripoll-Sierra es Ingeniero Mecánico egresado de la Universidad del Atlántico (Puerto Colombia, Colombia). (e-mail: hripoll_14@hotmail.com).

H. Ripoll-Goenaga es Ingeniero Electrónico egresado de la Universidad del Norte (Puerto Colombia, Colombia). (e-mail: heripoll@yahoo.com).

J. Roldán-Mckinley trabaja como professor del Programa de Ingeniería Mecánica de la Universidad del Atlántico (Puerto Colombia, Colombia), y es investigador del grupo DIMER. (e-mail: javierroldan@mail.uniatlantico.edu.co). Autor de correspondencia

E. Yime-Rodríguez trabaja como professor del Programa de Ingeniería Mecánica de la Universidad del Atlántico (Puerto Colombia, Colombia), y es investigador del grupo DIMER. (e-mail: eugenioyime@mail.uniatlantico.edu.co).

desarrollo de la rutina de programación permite el desarrollo de las secuencias de proceso asociadas a las etapas de trabajo y la representación gráfica de eventos discretos, con un modelado de sistemas distribuidos, paralelos o concurrentes [9, 22, 23].

El propósito de este trabajo es mostrar una metodología general para la conversión de las redes de Petri o Peri Nets-PN en lógica de escalera o Ladder Logic Diagram-LLD. El método es conceptualmente simple, y consiste de dos etapas. Primero, la red de Petri se convierte en Token Passing Logic-TPL, y luego el TPL se convierte en LLD. Esto también proporciona una correlación directa entre la información básica de secuenciación y los pasos de programación. Además, debido a la estructura el método es muy fácil de modificar o de ampliar el programa si cambian los requisitos de control. El artículo inicia con la familiarización de la máquina y el proceso de tampografía para establecer la respectiva rutina de trabajo. Se describen la operación y partes de la impresora, el proceso y la rutina de trabajo. Se continúa con las propiedades básicas de las redes de Petri y se crea una red de Petri para la máquina de tampografía. Posteriormente se desarrolla el programa en lógica escalera a partir de la red de Petri. Se cierra con el hardware y conexiones de entrada salida al PLC siemens S7200.

II. MATERIALES Y MÉTODOS

A. Descripción de la Máquina Tampográfica

La Fig. 1 presenta un esquema de la máquina de tampografía Comec LPE-60 en el cual se ilustran las principales partes. La placa de la imagen o cliché, elemento (2), es una placa plana que contiene la imagen grabada a ser imprimida. Los clichés utilizados para la impresión de tampografía son clichés de acero o clichés de fotopolímero [24]. La bandeja de tinta (3) encierra la tinta en un recipiente a prueba de evaporación. Durante la impresión, la bandeja de tinta se mueve hacia delante y hacia atrás a través de la superficie de la placa de imagen, depositando tinta en el grabado. La velocidad a la que la bandeja de tinta de tinta se mueve a través del cliché está determinada por la velocidad del ciclo de la máquina [25].

La almohadilla de impresión (1) transfiere la tinta de la imagen grabada en el cliché (2) a la superficie de impresión de una pieza de producción. El material básico de la almohadilla es sobre todo caucho de silicona. Las almohadillas están disponibles en varias formas, tamaños, durezas y estructuras superficiales [25, 26]. La mesa de trabajo ajustable (4) puede moverse cartesianamente con tres grados de libertad: hacia arriba y abajo, avance y retroceso, y lateralmente. La función del cilindro impresor (5) es que la almohadilla haga contacto con la pieza a marcar. Por su parte el cilindro (6) tiene como función hacer retroceder el cilindro impresor (5) para que este recoja tinta del cliché (2), y lo regresa a la posición de entintado, mientras que el cilindro entintador (7) tiene como tarea el entintado del cliché (2) por el contacto directo con la copa de tinta (3).

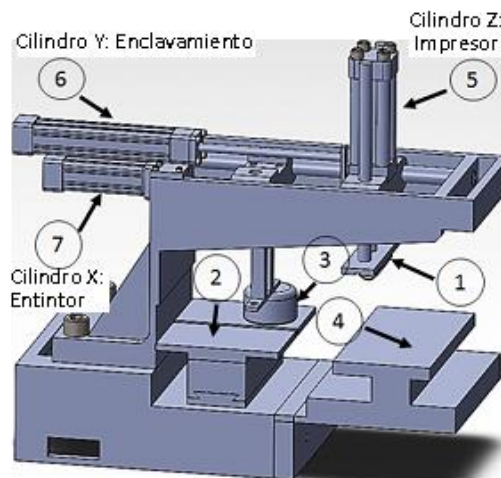


Fig. 1. Esquema de la máquina tampográfica Comec LPE-60. (1) Pad o almohadilla – (2) Cliché – (3) Copa de tinta – (4) Mesa de trabajo ajustable – (5) Cilindro impresor – (6) Cilindro de enclavamiento – (7) Cilindro entintador.

B. Descripción del Proceso de Tampografía

La Fig. 2 presenta la descripción del proceso de tampografía típico de la máquina Comec LPE-60, en secuencia A-B-C-D. En el inicio del proceso, cuadro A, la almohadilla (1) está situada en la posición hacia adelante y la bandeja de tinta cubre el grabado (2) con tinta. Al principio, la almohadilla y la bandeja de tinta se mueven hacia atrás y la bandeja de tinta libera el grabado en la placa (cliché) (B). La almohadilla baja y recoge la tinta del grabado, parte de la película de tinta pegajosa dentro del grabado es recogida por la almohadilla, el resto de la tinta permanece dentro del grabado (B), luego sube y se mueve hacia adelante junto con la bandeja de tinta (C). Durante el movimiento hacia delante, la bandeja de tinta llena el grabado de tinta, mientras que la tinta en la almohadilla se vuelve más adhesiva, como consecuencia de la evaporación de los disolventes, facilitando así la liberación completa de la almohadilla. Ahora la almohadilla baja sobre el objeto, cuadro D, y deja la tinta en la forma tomada por el cliché (2). A medida que la almohadilla regresa a su posición original sobre la placa y está lista para la próxima impresión.

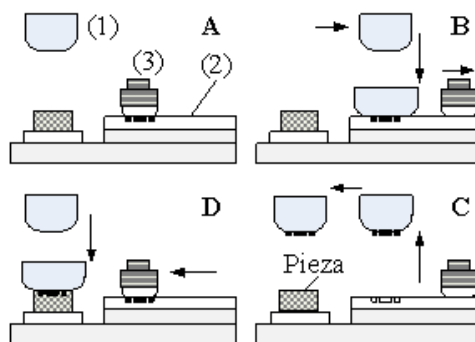


Fig. 2. Transferencia del contorno de tinta en el proceso de tampografía, secuencia A-B-C-D.

C. Elementos y Actuadores del Proceso de Tampografía

Se presentan los elementos primarios que intervienen en el ciclo de trabajo. La Fig. 3 presenta la disposición de los cilindros X: entintador, unido al depósito de tinta o tintero; Y: de enclavamiento y unido al cilindro impresor; y Z: impresor, que

contiene el tapón de silicona que recoge la tinta para el entintado del tampón y posteriormente imprime sobre el producto. En la misma figura se muestran los sensores de posición, representados por SX0 y SX1 para detectar las posiciones adentro y afuera, respectivamente, del cilindro entintador; SY0 y SY1 para detectar las posiciones adentro y afuera, respectivamente, del cilindro de enclavamiento; y SZ0 para detectar la posición adentro del cilindro impresor. En la posición afuera del cilindro impresor se tienen dos temporizadores, un temporizador T1 para establecer el tiempo de entintado o recolección de tinta en el tampón de silicona, y un temporizador T2 para el tiempo de impresión o tiempo en contacto con el producto, dependiendo del material a tampografiar, sea madera o plástico. También en la Fig. 3 el producto a marcar se identifica como pieza, y como tintero el elemento en el cual el tampón de silicona recoge la tinta.

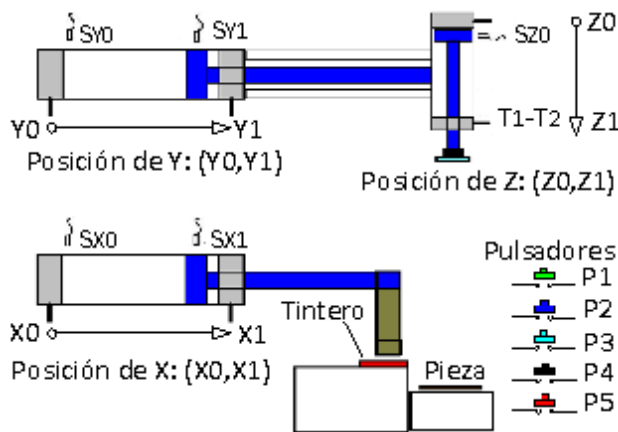


Fig. 3. Elementos de la máquina en el proceso de tampografía.

D. Diagramas de Fase

El diagrama de fase para las rutinas de ciclo único y de tinta se presenta en la Fig. 4, el ciclo único se distingue entre las líneas punteadas y el ciclo continuo es la repetición de un ciclo único varias veces, donde la secuencia se repite a partir de la primera línea punteada hasta la siguiente, de manera indefinida hasta que se genere o se interrumpa con una señal de STOP al ciclo por el operario. Para la rutina de batido de tinta, Fig. 5, el cilindro estampador impresor Z mantiene su posición adentro durante todo el desarrollo de la secuencia, mientras que los demás cilindros se mueven siguiendo la secuencia establecida.

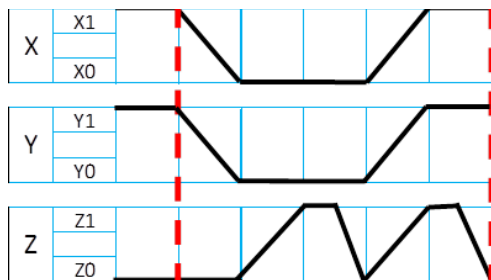


Fig. 4. Diagrama de fase para rutina de ciclo único y ciclo continuo.

E. Desarrollo de Estados y Condiciones

Se divide la secuencia de trabajo en pasos o estados. La transición de estado a estado genera las rutinas para el funcionamiento de la máquina, siendo los estados de partida las

posiciones de los cilindros definidos en la posición HOME y serán tomados como condiciones de inicio para la rutina de trabajo. En la Tabla I se agrupa cada uno de los estados y las condiciones para cada uno de estos estados, es decir que estas condiciones son funciones tipo “y”, requiriendo que todas estas condiciones se cumplan para dar paso al estado siguiente dentro de la secuencia de trabajo.

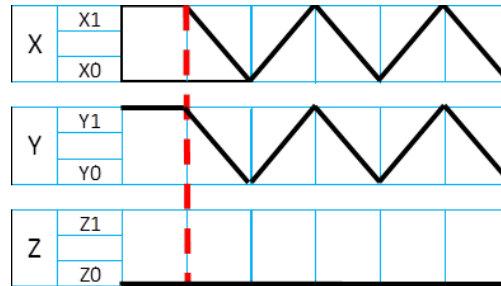


Fig. 5. Diagrama de fase para la rutina de batido de tinta.

Como se puede interpretar de las condiciones de la Tabla I, se hace referencia a un temporizador (T1 o T2) cada vez que el cilindro impresor sale a la posición Z1. Este temporizador se utiliza para programar el regreso del cilindro a la posición Z0. Se emplea el temporizador debido a que el cilindro de entintado físicamente no dispone de un sensor, y no es posible hacer uso de espacio para instalarlo, por lo que no se puede captar alguna señal cuando el vástago de este cilindro haya salido completamente. La solución escogida fue temporizar el retroceso del cilindro a partir del momento en que abandona la posición Z0. Teniendo en cuenta estas condiciones y cada una de las acciones a realizar, junto con los diagramas de fases con cada una de las rutinas, es posible realizar el análisis mediante la metodología de Redes de Petri y comenzar la programación mediante el lenguaje Ladder del PLC.

F. Condiciones, Acciones de Estados y Rutinas de Trabajo

Para comenzar el desarrollo de la programación primero se realiza un diagrama tabular para establecer las condiciones necesarias para pasar de un estado a otro, así como las acciones que se realizarán en cada estado [9], ver Tabla I.

TABLA I
CONDICIONES PARA GENERAR ESTADOS MEDIANTE RED DE PETRI

Condiciones Previas	Estado	Condiciones Generadas
	Posición de HOME	X en X1; Y en Y1; Z en Z0
X en X1; Y en Y1; Z en Z0	Entrada cil. entintor X; entrada cil. enclavamiento Y	X en X0; Y en Y0; Z en Z0
X en X0; Y en Y0; Z en Z0	Salida cil. impresor Z	X en X0; Y en Y0; Z en Z1; T1 activo
X en X0; Y en Y0; Z en Z1; T1 activo	Entrada cil. impresor Z	X en X0; Y en Y0; Z en Z0
X en X0; Y en Y0; Z en Z0	Salida cil. entintor X; salida cil. enclavamiento Y	X en X1; Y en Y1; Z en Z0
X en X1; Y en Y1; Z en Z0	Entrada cil. entintor X; entrada cil. enclavamiento Y; salida cil. impresor	X en X0; Y en Y0; Z en Z1; T2 activo
X en X0; Y en Y0; Z en Z1; T2 activo	Entrada cil. impresor Z	X en X0; Y en Y0; Z en Z0

Este diagrama tabular debe tener el estado, las acciones y las condiciones para cada estado. Con base en la Tabla I, los diagramas de fase de las Figs. 4 y 5, y las condiciones de trabajo para cada rutina, se desarrollarán los diagramas tabulares que relacionarán las condiciones que se generan en cada estado dentro de las secuencias de trabajo, con un conjunto de acciones que son ejecutadas una vez alcanzadas tales condiciones. El contenido de cada uno de los campos del diagrama tabular está dado por: E: estado presente; Estado: descripción del estado presente; Acción: acciones que se realizan en el estado presente; Condiciones: son los permisivos y/o las condiciones lógicas que permiten pasar de un estado a otro; ED: estado siguiente a ejecutar cuando las condiciones de salto se cumplan [24]. Se establecen todos los elementos del sistema: Cilindros {X, Y, Z}, Fig. 3; Sensores {SX0, SX1, SX0, SX1, SZ0}, Fig. 1; Pulsadores {P1, P2, P3, P4, P5}, Fig. 1; Temporizadores {T1 (entintado), T2 (impresión)}, Fig. 3. A continuación se definen: Estados {E1, E2, E3, E4, E5, E6, E7, E8, E9}; Condiciones {C1, C2, C3, C4, C5, C6, C7}; Acciones {A1, A2, A3}, y Acciones Negadas { $\overline{A1}$, $\overline{A2}$, $\overline{A3}$ }. La Tabla II presenta el diagrama tabular de estados y condiciones, cuya narrativa de operación o secuencia de trabajo de la máquina se presenta a continuación. La máquina inicialmente se encuentra en la posición de HOME. Se tienen dos pulsadores, P1 para activar el ciclo único o P2 para ciclo continuo. Después de activar cualquiera de los dos pulsadores, el cilindro entintor X y el cilindro de enclavamiento Y entran, por lo que los sensores SX0 y SY0 detectan la posición de los cilindros y envían la señal de salida del cilindro impresor Z para que recoja la tinta, al tiempo que se envía señal al temporizador T1.

Transcurrido el tiempo del temporizador T1, cesa la alimentación de la válvula solenoide del cilindro Z y este regresa la posición Z0, en donde el sensor SZ0 detecta el cilindro adentro y envía una señal que es tomada junto con la de los sensores SX0 y SY0. Luego el cilindro entintor X y el cilindro de enclavamiento Y salen hasta llegar a las posiciones X1 y Y1 (Fig. 2), respectivamente, posición detectada por los sensores SX1 y SY1, que envían una señal para que el cilindro de entintado salga nuevamente y realice el entintado sobre la pieza de trabajo, y a su vez se vuelve a activar el temporizador T2 para accionar la entrada del cilindro.

Nuevamente el sensor SZ0 detecta la posición del cilindro impresor y finaliza el ciclo, si la opción que se ha escogido ha sido la de ciclo único. Si por el contrario, la opción escogida es la de ciclo continuo, entonces se activa una señal que permanece activa para continuar el ciclo de trabajo, cuando esta conmuta con la última señal recibida por el sistema. Esta señal permanece activa hasta que se haga un stop general de la máquina o una parada del ciclo continuo mediante otro pulsador que se ha denominado P3. Si la opción escogida para el funcionamiento de la máquina ha sido la de batido de la tinta, se activa un pulsador que se ha marcado como P4, el cual genera un reset tanto en la rutina principal como en la secuencia de batido de la tinta. Esta comienza cuando los sensores SX1 y SY1 generan señal debido a que están activas las posiciones X1 y Y1. Por lo tanto, los cilindros de enclavamiento y el entintor

entran y llegan a las posiciones X0 y Y0. Cuando las posiciones son sensadas por SX0 y SY0, los cilindros nuevamente salen a las posiciones X1 y Y1, mientras el cilindro impresor mantiene la posición Z0. Este ciclo se repite de manera indefinida hasta que se pulse el pulsador P5 para dar fin a la etapa de batido de la tinta.

G. Diagrama de Red de Petri

La elaboración de las redes de Petri se fundamenta en definiciones matemáticas y relaciones lógicas para los parámetros necesarios en el diagrama. Las secuencias dentro de los modelos de Petri son una serie de operaciones que ocurren de manera consecutiva. Estas operaciones se denominan estados o lugares. Para ir de un lugar a otro se deben cumplir unas condiciones llamadas transiciones, que determinan el fin de una operación y el comienzo de otra. La transición correspondiente a la secuencia i se representa con t_i [6]. Los lugares son los elementos pasivos de la red y junto con los tokens, se utilizan para modelar los estados del sistema.

TABLA II
DIAGRAMA TABULAR DE ESTADOS Y CONDICIONES





E	Estado	Acciones	Condiciones	ED
E1	Posición HOME	A1: X afuera A2: Y afuera $\overline{A3}$: Z adentro	C1: SX1 activo C2: SY1 activo C3: SZ0 activo P1 o P2 activo C4: SX0 activo C5: SY0 activo C3: SZ0 activo	E2
E2	X entrando; Y entrando	$\overline{A1}$: X adentro $\overline{A2}$: Y adentro		E3
E3	Z saliendo	$\overline{A1}$: X adentro $\overline{A2}$: Y adentro A3: Z afuera T1 Activo	C6: T1 activo	E4
E4	Z entrando	$\overline{A1}$: X adentro $\overline{A2}$: Y adentro $\overline{A3}$: Z adentro	C4: SX0 activo C5: SY0 activo C3: SZ0 activo	E5
E5	X saliendo; Y saliendo	A1: X afuera A2: Y afuera $\overline{A3}$: Z adentro	C1: SX1 activo C2: SY1 activo C3: SZ0 activo	E6
E6	Z saliendo	A1: X afuera A2: Y afuera A3: Z afuera T2 activo	C7: T2 activo	E7
E7	Z entrando	A1: X afuera A2: Y afuera $\overline{A3}$: Z adentro	C1: SX1 activo C2: SY1 activo C3: SZ0 activo	E1
E1	Posición HOME	A1: X afuera A2: Y afuera $\overline{A3}$: Z adentro	C1: SX1 activo C2: SY1 activo C3: SZ0 activo P4 activo	E8
E8	X entrando; Y entrando	$\overline{A1}$: X adentro $\overline{A2}$: Y adentro $\overline{A3}$: Z adentro	C4: SX0 activo C5: SY0 activo	E9
E9	X saliendo; Y saliendo	A1: X afuera A2: Y afuera $\overline{A3}$: Z adentro	C1: SX1 activo C2: SY1 activo	E1

Las transiciones son los elementos activos de la red de Petri, y representan las acciones de un sistema, las cuales originan cambios en los estados. El conjunto de lugares, transiciones y arcos son finitos y estáticos, por lo cual el sistema no tendrá más causas y eventos que los originalmente representados en el modelo. Sin embargo, el conjunto de tokens y marcas puede

cambiar durante la ejecución de la red, describiendo las características variantes (dinámicas) del sistema modelado que requieren para la evolución que las transiciones estén sensibilizadas, es decir, que los lugares de entrada de cada transición se encuentren marcados y que se verifiquen los disparos correspondientes a cada transición [9, 26]. Las redes de Petri representan las condiciones como lugares, y los eventos que generan estas condiciones los representa como transiciones. Una transición o evento tiene un número de lugares de entrada y salida llamadas pre-condiciones y las post-condiciones del evento, respectivamente [9].

La red de Petri es un tipo particular de grafo dirigido que consiste en dos tipos de nodos: lugares y transiciones. También se puede ver como una estructura matemática compuesta por lugares representados por circunferencias; y transiciones que se representan por segmentos rectilíneos, arcos y tokens [9]. Gráficamente, la red de Petri es un gráfico orientado, compuesta por etapas, transiciones y arcos, como se describe en la Tabla III, y se explica a continuación.

TABLA III
CONDICIONES PARA GENERAR ESTADOS MEDIANTE RED DE PETRI

Elemento	Etapa		Transición	Arco
	Lugar	Marcaje		
Simbología				

Etapas. Son estados estables en el proceso que asocian las órdenes que se envían al proceso. La etapa ubicada antes de una transición se denomina etapa de entrada y la que está después de una transición se conoce como etapa de salida. La etapa o estado presente (etapa activa) se diferencia con un elemento en su interior llamado marca o “token”. El marcaje será la disposición dada a las marcas durante la operación del sistema. Las transiciones se denotan por el conjunto finito $P = \{p1, p2, \dots, pn\}$, y no vacío de n lugares.

Transición. Las transiciones en una red de Petri permiten el paso (transición) de un estado del proceso a otro. Cada transición tiene asociada una ecuación lógica denominada “receptividad”, que indica las condiciones necesarias para la evolución del proceso. El conjunto no vacío y finito de m transiciones se denota por $T = \{t1, t2, \dots, tm\}$.

Arco. Son los conectores, o flechas, que relacionan las entradas y transiciones.

La función de entrada o incidencia previa I , se denota por $I: P \times T \rightarrow N$, y especifica los lugares de entrada de la transición, $N = \{1, 2, 3, \dots\}$. La función de salida o incidencia posterior es $O: P \times T \rightarrow N$ especifica los lugares de salida de la transición. Se tiene entonces una estructura algebraica $R = (P, T, I, O)$ asociada a la red de Petri.

Para que una red de Petri permita modelar correctamente un proceso secuencial, debe cumplir con las propiedades que se definen en la Tabla IV, las cuales se basan en los modelos matemáticos desarrollados para la validez de la funcionalidad de esta metodología de programación, pero que no son de interés de estudio en este artículo. Únicamente se enuncian las propiedades desde un punto cualitativo para un desarrollo práctico de las redes de Petri. Tales redes presentan procesos

secuenciales o una serie de estados o etapas, las cuales se ejecutan siguiendo una secuencia sencilla, desde un estado inicial hasta un estado final, como se presenta esquemáticamente en la Fig. 6.a [28, 29]. La aplicación de las redes de Petri para la programación permite implementarlas en procesos con acciones paralelas, hasta llegar a un estado en común para las líneas de trabajo, lo cual es posible gracias a los procesos de bifurcación como se ilustra en la Fig. 6.b para el caso más sencillo de dos secuencias de acciones, que se puede implementar para más de dos acciones en paralelo.

Cuando se habla de evolución de la marca (token) dentro de la secuencia de la red de Petri, se alude al movimiento o desplazamiento de las marcas por los diferentes estados o etapas del diagrama, según la evaluación de las transiciones asociadas a dicho estados [30]. En la Tabla V se presentan algunas reglas de juego básicas para trabajar las secuencias y evolución del “token”. Una metodología típicamente utilizada en Sistemas Dinámicos Discreto-DEDs está basada en la denominada APN o Automation Petri Nets, redes que son robustas, no son ordinarias ni binarias, sus arcos pueden tener pesos mayores a uno, y los lugares pueden acumular más de una marca, por lo cual las APN son acotadas. El modelo APN es apropiado para el manejo de señales, sensores y actuadores. Bajo este concepto se siguen los métodos para el desarrollo de programas de automatización por medio del concepto de Token Passing Logic-TPL que permite obtener una sencilla implementación en el lenguaje Ladder [9].

TABLA IV
PROPIEDADES DE LAS REDES DE PETRI

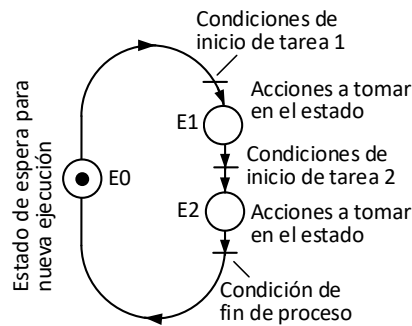
Propiedad	Descripción
Sana	Esta propiedad garantiza que nunca puede haber más de una marca (token) en las etapas cuando se realiza la evolución. Si esto se cumple entonces es posible describir el sistema utilizando funciones lógicas para activar o desactivar un estado o etapa.
Viva	Esta propiedad garantiza que después de evolucionar, todas las transiciones que conforman la red sean activadas. Así se asegura que la red no se bloquee en un determinado momento.
Propia	Esta propiedad garantiza que después de evolucionar, la red puede regresar a su estado inicial, con lo que se asegura un comportamiento repetitivo del sistema.

Teniendo en cuenta las definiciones anteriores para el desarrollo de los diagramas de Petri, se establecen los lugares, transiciones y sensores a tener en cuenta para generar la APN. Haciendo uso de la información en el diagrama tabular de la Tabla II, se recuerdan los estados o lugares: $\{E0, E1, E2, E3, E4, E5, E6, E7, E8, E9\}$; los sensores: $\{SX0, SX1, SZ0\}$ y los pulsadores: $\{P1: \text{ciclo único}, P2: \text{ciclo continuo}, P3: \text{fin de ciclo continuo}, P4: \text{batido de tinta}, P5: \text{parada de batido de tinta}\}$. Los sensores y los pulsadores hacen parte de las condiciones, y serán enunciados en las transiciones dentro del diagrama. Teniendo todas estas definiciones y las condiciones necesarias para el desarrollo del proceso se desarrolla la gráfica de Red de Petri de la Fig. 7. E0 indica un estado de posición inicial que es al mismo tiempo el estado de HOME, es decir en E0 y E1 en la máquina se encuentran en la misma posición, sin

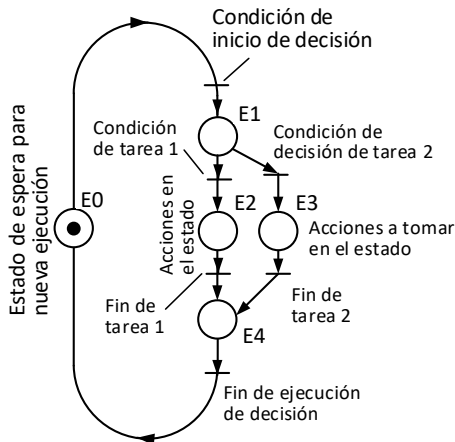
embargo E0 ha sido creado con el fin de permitir un estado de inicio que permita hacer conexiones con los demás lugares pertenecientes a las etapas de batido de la tinta y de parada de ciclo continuo.

REGLAS DE EVOLUCIÓN DE LOS TOKENS EN LAS REDES DE PETRI [30]

Regla	Representación
La marca evolucionar si la transición es sensible, es decir, todas las etapas de entrada deben estar activadas.	
La marca evoluciona si la transición es activada, es decir, la ecuación de receptividad asociada a la transición sea igual a 1.	
En la evolución se retira una marca en cada una de las transiciones de entrada de la transición activa y se coloca una marca en cada una de las etapas de salida.	



a. Proceso secuencial en redes de Petri



b. Proceso de bifurcación en redes de Petri

Fig. 6. Diagrama de proceso secuencial y de decisión en redes de Petri [31].

Cuando se energiza la máquina, se cuenta con tres opciones iniciales de inicio a definir con los pulsadores: ciclo único con el pulsador P1, ciclo continuo con el pulsador P2 y batido de tinta con el pulsador P4, respectivamente. Si se escoge el pulsador P1, el token realiza su trayectoria a través de la transición t8. El token pasa al lugar intermedio U, que permite la recepción del token cuando se ha elegido la etapa de ciclo único, mientras espera que en t9 se cumplan las condiciones de señales en SX1, SY1 y SZ0. Luego de ser disparada la

transición t9, el token llega al lugar o estado E2 que se enuncia en el diagrama tabular donde el cilindro enclavador y entintor entran al mismo tiempo, hasta llegar a las posiciones SX0 y SY0. Después de cumplida la función asignada para el lugar E2, la transición t2 espera por las condiciones SX0, SY0 y SZ0. Cuando se dispara la transición, el token es marcado en el lugar E3, y la acción realizada aquí es la salida del cilindro impresor a SZ1, así como la activación del temporizador T1.

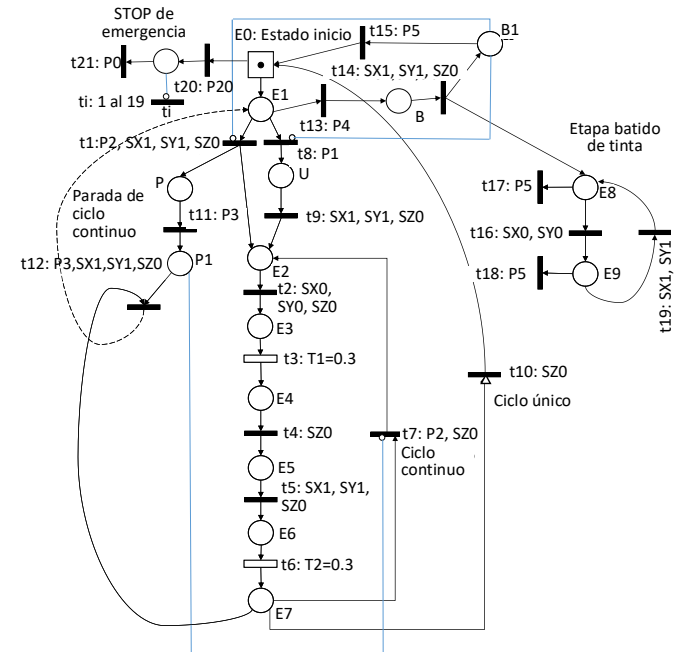


Fig. 7. Simulación para máquina tampográfica COMEC mediante diagrama de redes de Petri.

Se continúa con la transición t3. Esta tiene como condición que el temporizador T1 marque el tiempo programado, se ha dado un valor de 0.3 s para recoger la tinta de la placa con el huecograbado. Cuando se cumple la condición, el token es marcado en el lugar E4, donde el cilindro impresor regresa a la posición de SZ0. La transición t4 tiene como condición la activación de la señal SZ0, al ser disparada se marca el token en el lugar E5, para este lugar los cilindros entintor y enclavador salen de nuevo pasando a SX1 y SY1. La siguiente transición, t5, depende de la condición de activación de las señales SZ0, SX1 y SY1. Luego el token es marcado en el lugar E6, donde el cilindro impresor baja, SZ1, esta vez para realizar el tampografeado sobre el producto, y a su vez se activa el temporizador T2, el cual después de 0.3s habilita la transición t6 para que el token sea marcado en el lugar E7 donde el cilindro impresor regresa a la posición de entrada y da la señal SZ0 en la transición t10, finalizando el ciclo de trabajo. Cuando esto ocurre el token es marcado de nuevo en el estado inicio E0. Si al inicio del ciclo se escoge P2, es decir la etapa ciclo continuo, el token será condicionado por la transición t1 y luego de que se cumplan las condiciones de activación para SX1, SY1 y SZ0, el token será marcado directamente en el lugar E2 y seguirá el mismo flujo explicado anteriormente para el ciclo único a partir de E2 hasta el lugar E7.

En el lugar E7 se asignaron 3 arcos de salida, el primero va a

la transición t7, donde se encuentran las condiciones de SX0 y la marca activa de SZ0. Es decir, cuando se escoge ciclo continuo y el token se encuentra en el lugar E7 y el cilindro impresor llega a la posición Z0, mediante la activación de la señal de SZ0, el token es marcado en el lugar E2, repitiendo así el ciclo de funcionamiento. Esta condición de ciclo continuo se repetirá hasta que se active el pulsador P3. Cuando se escoge el ciclo continuo y se activa el pulsador P2 junto con las condiciones SX1, SY1 y SZ0, un token es marcado en el lugar P hasta que se activa la transición t11; al ser pulsado P3, el token es marcado en el lugar P1 que tiene dos salidas: una a la transición t12 y otra a la transición t7, pero el arco que llega a la última es un arco inhibidor, representado por un pequeño círculo vacío. Los arcos inhibidores permiten que la transición sea activada con las condiciones que la conforman, sin importar que esta tenga dos entradas. Cuando las dos entradas a la transición se encuentran activas, la transición queda inhabilitada, por lo tanto, el token no puede ser marcado en el lugar siguiente. Este es el caso para dar stop al ciclo continuo de la máquina, cuando se activa el pulsador P3 y el pulsador P1 se encuentra marcado, se inhabilita la transición t7, por lo tanto, el token es marcado en el lugar E1 después de que se cumplan las condiciones en la transición t12 cuando todos los cilindros se encuentren en la posición de home. Es decir, el token se lleva a la posición inicial para comenzar una nueva etapa, a la espera de señal.

Para distinguir entre un ciclo único y un ciclo continuo, cuando se ha realizado un ciclo completo, se dispone de un arco habilitador el cual es representado por una flecha vacía. Este arco tiene una función de testeo, este solo habilita la transición con el arco habilitador cuando el token permanece en el lugar anterior a la transición, y no es marcado a través del arco normal en otro lugar, por lo cual este arco especial habilita la transición a la cual se encuentra ligado. Este es el caso de la transición t10, cuando se encuentra la marca en el lugar E7, y tanto la transición t7 como la transición t12 no se encuentran habilitadas, la transición t10 se habilita y se marca de nuevo al estado de inicio. Para la secuencia de batido de la tinta se tiene una transición t13 que depende de la condición de pulsar P4. Cuando la transición se habilita la marca o token pasa al siguiente estado, el lugar B. Al llegar la marca a B, la transición t14 que depende de las condiciones de SX1, SY1 y SZ0, es habilitada y el token es marcado en el lugar B1 y en el lugar E8. El token marcado en el lugar B1 inhabilita las transiciones t1 y t8, esto permite que al activar las funciones de batido de tinta, las otras etapas no puedan ser activadas, evitando conflictos dentro de la programación y el funcionamiento de la máquina. Cuando el token es marcado en el lugar E8, los cilindros entintor y de enclavamiento entran a las posiciones X0 y Y0, respectivamente. Luego la transición t16 es habilitada cuando se cumplen las condiciones SX0 y SY0, y el token es marcado en el lugar E9 donde los cilindros de enclavamiento y entintor salen de nuevo. Este ciclo se repite pasando por la transición t19 que depende de las condiciones SX1 y SY1, hasta que se pulsa P5 y se desactiva la etapa de batido de tinta. Al pulsar P5 las transiciones t17 y t18 eliminan los tokens de los lugares E8

y E9, y al mismo momento la transición t15 permite que el token que se encontraba en el lugar B1 sea marcado en E0, el estado de inicio del sistema, dejando la máquina habilitada para realizar una nueva función o etapa de trabajo.

Para el stop general del sistema, se designó un lugar denominado S, en donde se marcará un token después de se cumpla la condición de S0 y se habilite la transición t20. Cuando el lugar se encuentra con un token, el arco especial inhabilita todas las transiciones, generando la parada donde se encuentra la marca principal, deteniendo el proceso y anulando todo tipo de función. Esta marca del lugar S es anulada cuando se pulsa de nuevo P0 y queda abierto el contacto, permitiendo que la máquina continúe el proceso. Los arcos que se han representado mediante líneas de trazo corto y puntos en la Fig. 7 son arcos inhabilitadores, representados mediante este color para que puedan ser apreciados y diferenciados del resto de los arcos que conforman el sistema, y tener en cuenta la importancia de la función de estos arcos dentro del desarrollo de las funciones por medio de las redes de Petri. Finalmente se ha obtenido toda la secuencia requerida para el funcionamiento de la máquina y en cada una de las funciones específicas de trabajo. La metodología de Petri ha permitido hallar las condiciones necesarias para el cumplimiento de cada una de las actividades de la secuencia de trabajo, al tiempo que evita los conflictos y cruces de señales, algo que se debe evitar en las programaciones, para evitar contratiempos.

III. RESULTADOS

A. Programación en KOP Ladder

Se presenta ahora la implementación del sistema de eventos discretos por medio del lenguaje KOP, teniendo en cuenta la portabilidad entre las Redes de Petri y la lógica cableada. La forma como se desarrollará el lenguaje Ladder será a través de programación lineal, por lo que no se hará uso de lenguaje estructurado a través de subrutinas, debido a la simplicidad del proceso [32]. La red de Petri permite el paso a una estructura tipo lógica cableada para fácil ejecución bajo técnicas de programación en PLC's. Cada lugar de una red de Petri representa un estado, condición o recurso y se asocia en los diagramas de lógica cableada con uno de los escalones, permitiendo el control de marcas en el lugar asociado. Las transiciones corresponden a los eventos o acciones que permiten la evolución del marcado y se asocian con los contactos [33]. Los procedimientos de diseño de sistemas secuenciales mediante redes de Petri deben contar con una metodología para su implementación en equipos digitales. Para la traducción del modelo del sistema a un lenguaje de PLC normalizado se tendrán en cuenta las representaciones de la Fig. 8 para el cambio de red de Petri a lógica cableada: para la función "Y" (Fig. 8.a), la función "O" (Fig. 8.b), para un flanco (Fig. 8.c) y para un retardo (Fig. 8.d). A partir del desarrollo de APN basado en el Token Passing Logic para el proceso de tampografeado, se realiza la migración al lenguaje Ladder teniendo en cuenta las equivalencias establecidas por la norma IEC-1131-3 [34], para cada una de las rutinas que componen el

ciclo [31, 35]. Se tienen en cuenta cada una de las transiciones, lugares y los tipos de arcos tanto de entradas como de salidas.

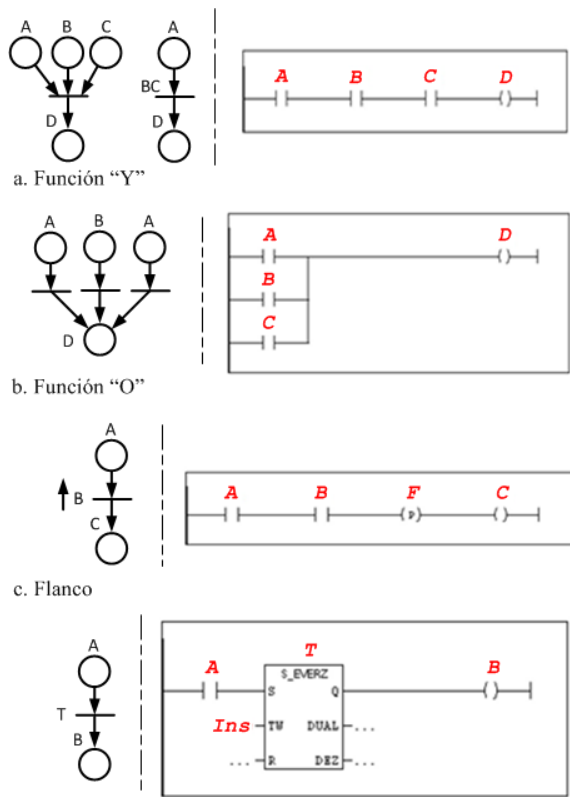


Fig. 8. Equivalencia de red de Petri a lógica cableada, algunas funciones [36].

En las Figs. 9 y 10 se observan los diagramas de Ladder con las secuencias correspondientes a las redes de Petri de las funciones de selección e inicio de rutina, respectivamente. En la Fig. 9 se aprecian los lugares, transiciones y arcos establecidos en la red de Petri. El estado de inicio se llama ahora con la marca SM0.0, que siempre está activa, así como en la red de Petri lo hace el estado de inicio correspondiente, E0, representado por un cuadro con un token que indica el punto de partida del flujo del token a través de las secuencias. El lugar E1 es representado a través de la marca M0.0, y los arcos ordinarios que determinan las salidas para cada una de las rutinas de funcionamiento de las máquinas son incluidos en los diagramas de Ladder en cada una de las líneas que conforman una función "O" a la salida de la marca M0.0. Las transiciones t1, t8 y t13 están dadas por las marcas IO.0, IO.1 e I1.0 respectivamente. Estas son las entradas de señal al PLC de pulsador de la rutina a seguir, donde IO.0 es la entrada de P1 (ciclo único), IO.1 es la entrada de P2 (ciclo continuo) y finalmente I1.0 que es la entrada para P4 (batido de la tinta).

En el diagrama de Petri, Fig. 7, se tienen dos arcos inhibidores de salida desde el lugar B1, que llegan a las transiciones t1 y t8. Los arcos inhibidores son arcos que, al ser estimulados por la presencia de un token en el lugar anterior, inhiben la transición a la cual se encuentra anidada y por ende inhibe las condiciones que disparan la transición siguiente. Por lo tanto, se puede afirmar que los arcos inhibidores son una interrupción en un grupo de condiciones, es decir, una interrupción en una

condición "Y" dentro del lenguaje Ladder. Los arcos inhibidores para este caso son representados como relés normalmente cerrados, aunque es posible representarlos a través de relés amarrados a marcas y que en el momento de actuar como un arco inhibidor se encuentren cerrados.

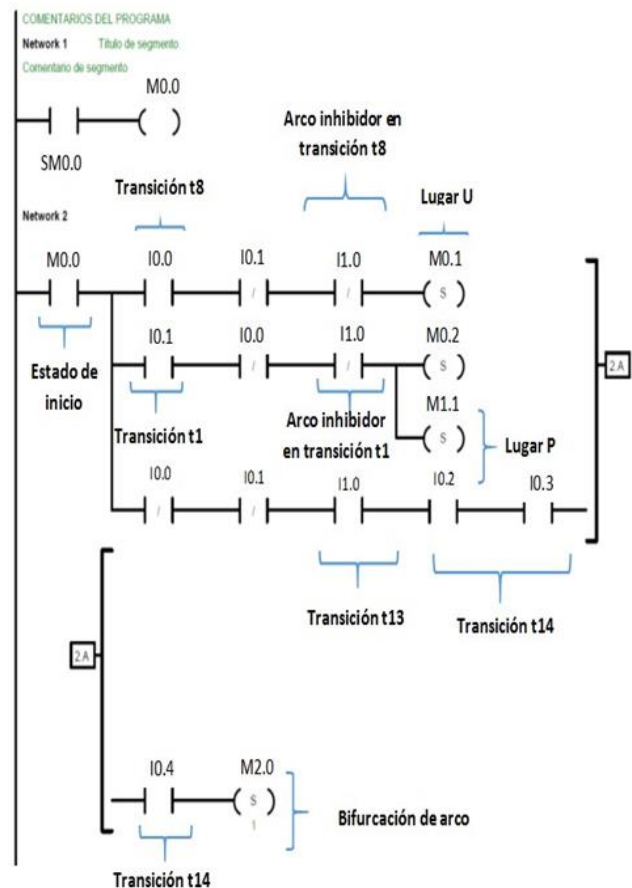


Fig. 9. Lenguaje Ladder con las referencias de la red de Petri de la función selección de rutina.

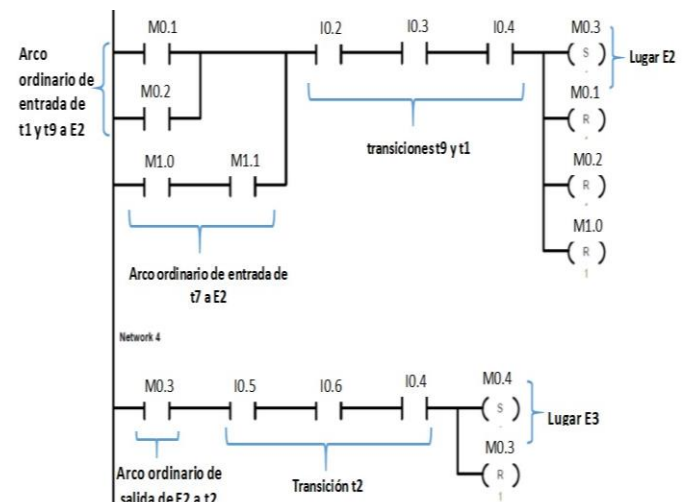


Fig. 10. Ladder con las referencias de la red de Petri de inicio de rutina.

Así como en las transiciones t1 y t8, Fig. 9, los arcos inhibidores se encuentran representados por los contactos normalmente cerrados, marcados por I1.0, que se encuentran en las líneas tanto para ciclo único como para ciclo continuo, como el nombre de los contactos lo indican estas están amarradas a la

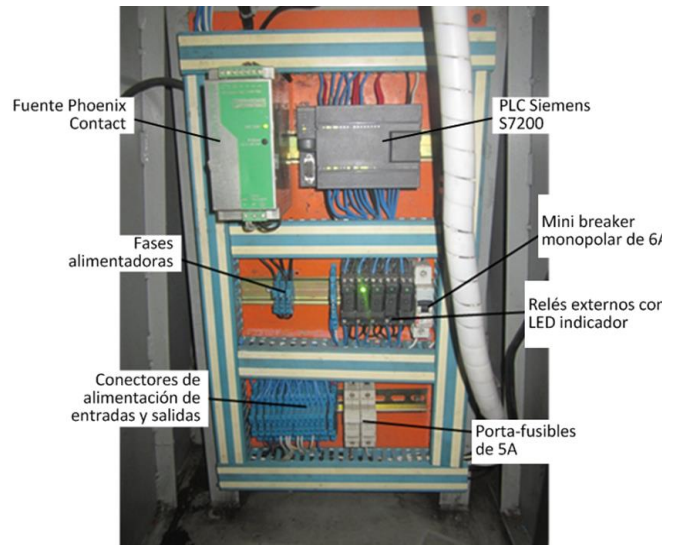
entrada I1.0 que dependen del pulsador P4, con el cual se activa la función de batido de tinta; aquí se cumple lo anteriormente descrito en el diagrama de Petri: al pulsar P4 el arco inhibitor no permite que se active simultáneamente la función de ciclo único o de ciclo continuo.

B. Circuitos Asociados

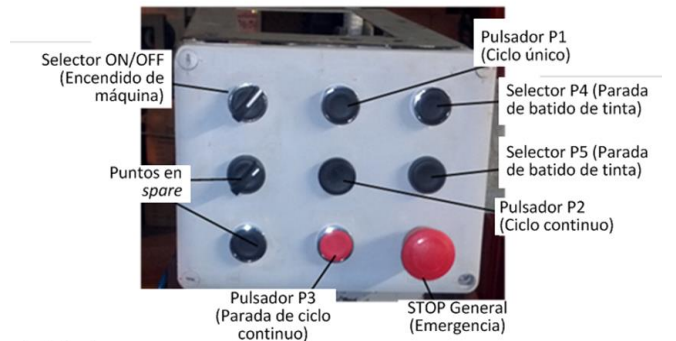
La Fig. 11 muestra la máquina recuperada. El cableado con el hardware integrado se muestra en la Fig. 12.a, mientras que los pulsadores fueron integrados en la botonera de la Fig. 12.b. El esquema de entradas y salidas al PLC presenta en la Fig. 13 concuerda con los rótulos en la Tabla VI. Para la implementación de la rutina de programación fue necesario rediseñar todos los sistemas de potencia y control. Se utilizó una fuente reguladora de voltaje Phoenix Contact, con una disminución de potencial de 110 VAC a 24 VDC con una corriente de alimentación de 5A. El circuito de potencia también cuenta con un minibrakeer de 6A con el fin de proteger la alimentación al PLC. Para el circuito de control se cuenta con todas las conexiones de los sensores y de alimentación de las válvulas.



Fig. 11. Máquina tampográfica COMEC recuperada.



a. Circuito de alimentación



b. Pulsadores

Fig. 12. Hardware y botones implementados.

TABLA VI
EQUIVALENCIA DE ENTRADAS Y SALIDAS AL PLC

Rótulo	Equivalencia
I0.0	Entrada señal de pulsador de inicio de ciclo único P1
I0.1	Entrada señal de pulsador de inicio de ciclo continuo P2
I0.2	Entrada de señal de sensor S _{X0}
I0.3	Entrada de señal de sensor S _{Y0}
I0.4	Entrada de señal de sensor S _{Z0}
I0.5	Entrada de señal de sensor S _{X1}
I0.6	Entrada de señal de sensor S _{Y1}
I0.7	Entrada de señal de pulsador de parada de ciclo continuo P3
I1.0	Entrada señal pulsador de inicio de batido de tinta P4
I1.1	Entrada señal de pulsador de fin de batido de tinta P5
Q0.0	Salida de señal a bobina para cilindro entintor adentro
Q0.1	Salida de señal a bobina para cilindro entintor afuera
Q0.2	Salida señal a bobina cil. de enclavamiento adentro
Q0.3	Salida señal a bobina cil. de enclavamiento afuera
Q0.4	Salida señal a bobina para cilindro de impresor afuera

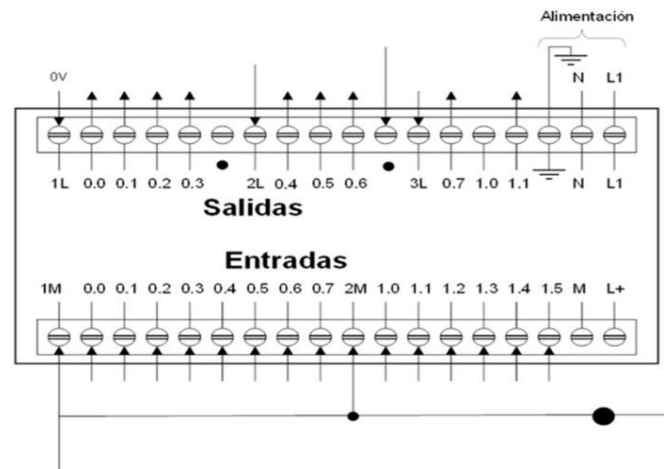


Fig. 13. Esquema de entradas y salidas al PLC.

IV. CONCLUSIONES

Se documentó en este artículo la programación del ciclo de automatización de una máquina usada en la tampografía de cepillos plásticos y de madera, dentro de un proyecto de recuperación funcional del equipo. Seguido de la identificación del ciclo de trabajo, se identificaron las rutinas requeridas: batido de tinta, operación de ciclo sencillo y operación en ciclo continuo. El hardware del proceso constituido por cilindros, temporizadores y finales de carrera, permitió determinar los diagramas de fase y desarrollo de estados y condiciones. Un

total de nueve condiciones y siete estados de la máquina permitieron la generación del diagrama tabular del proceso, para la generación de la red de Petri, en la cual los estados y transiciones se evalúan con la señal depositada en un token para facilitar comunicación durante el proceso. Finalmente, la red permitió la programación y simulación en lenguaje KOP Ladder, proporcionando de esta manera una aplicación específica de tampografía que puede ser utilizada como base para replicar en otros procesos.

REFERENCIAS

- [1] C. Petri, "Kommunikation mit automaten", Ph.D. dissertation, Bonn Institut für Instrumentelle Mathematik, Bonn, NRW, Deutschland, 1962.
- [2] J. Peterson, *Petri-Net Theory and the Modeling of Systems*. New Jersey, USA: Prentice Hall, 1983.
- [3] G. Berthelot, and R. Terrat, "Petri Nets theory for the correctness of protocols", *IEEE Trans. Comm.*, vol. 30, pp. 2497-2505, Dec. 1982, DOI: 10.1109/TCOM.1982.1095452
- [4] T. Murata, "A Petri Net based controller for flexible and maintainable sequence control and its application in factory automation", *IEEE Trans. Ind. Electronics*, vol. IE-33, no. 1, pp. 1-8, Feb. 1986, DOI: 10.1109/TIE.1986.351700
- [5] M. Kayama, "A high speed control method for automated machine system based on token transition of Petri Net theory", *Trans. Inst. of Electrical Eng. of Japan*, vol. 109-D, no. 7, pp. 463-469, July 1989.
- [6] H. Murakoshi, M. Sugiyama, G. Ding, T. Oumi, T. Sekiguchi, and Y. Dohi, "A high speed programmable controller based on Petri Net", in *Proc. of 1991 Int. Conf. Ind. Electronics, Control and Inst.*, Kobe, Japan, pp. 1966-1971, Oct. 1991, DOI: 10.1109/IECON.1991.239040
- [7] Y. Takeda, H. Murakoshi, N. Funakubo, and Y. Dohi, "Proposal of a modeling methodology for factory automation systems and the improved Petri net controller", in *Proc. of 1996 IEEE Conf. Emerging Tech. and Factory Aut.*, Kauai, HI, pp. 540-542, Nov. 1996, DOI: 10.1109/ETFA.1996.573915
- [8] D. Nakamura, Y. Takeda, H. Murakoshi, N. Funakubo, and Y. Dohi, "A modeling language for Petri Net based factory automation systems", in *Proc. of 24th IEEE Annual Conf. (Ind. Electronics Soc.)*, vol. 1, Aachen, Germany, pp. 120-125, Sep. 1998, DOI: 10.1109/IECON.1998.723956
- [9] M. Uzam, and A. Jones, "Discrete event control system design using automation Petri nets and their ladder diagram implementation", *The Int. J. of Adv. Manuf. Tech.*, vol. 14, no. 10, pp. 716-728, Oct. 1988, DOI: 10.1007/BF01438224
- [10] I. Antonzadis, and V. Leopoulos, "A concept for the integrated process description, PLC programming and simulation using Petri nets: application in a production process", in *Proc. of IEEE Int. Conf. on Sys., Man, and Cyber.*, Nashville, TN, pp. 2443-2448, Oct. 2000, DOI: 10.1109/ICSMC.2000.884358
- [11] Y. Nagao, H. Urabe, S. Nakano, M. Inui, and S. Kumagai, "Application of Petri net to automatic cutting system for shaped-steel", in *Proc. of IEEE Int. Workshop Emerging Technologies and Factory Automation*, pp. 50-55, Melbourne, Australia, Aug. 1992, DOI: 10.1109/ETFA.1992.683226
- [12] R. Zurawski, "Petri net models and functional abstractions: applications to the design of automated manufacturing systems", *Trans. Ind. Electronics*, vol. 52, no. 2, pp. 595-609, Apr. 2005, DOI: 10.1109/TIE.2005.844225
- [13] A. Moro, H. Yu, and G. Kelleger, "Hybrid heuristic search for the scheduling of flexible manufacturing systems using Petri nets", *IEEE Trans. Rob. and Automation*, vol. 18, no. 2, pp. 240-245, Aug. 2002, DOI: 10.1109/TRA.2002.999652
- [14] Z. Kasirovalad, M. Motlagh, and M. Shadmani, "An intelligent modular modelling approach for quality control of CNC machines product using adaptive fuzzy Petri nets", in *Proc. of 8th Control, Automation, Robotics and Vision Conf.*, vol. 2, Kunming, China, pp. 1342-1347, Dec. 2004, DOI: 10.1109/ICARCV.2004.1469041
- [15] K. Kobayashi, and T. Ushio, "An application of LLP supervisory control with Petri net models in mobile robots", in *Proc. IEEE Int. Conf. Syst., Man, and Cyber.*, vol. 4, Nashville, TN, pp. 3015-3020, Oct. 2000, DOI: 10.1109/ICSMC.2000.884460
- [16] D. Caihong, "Application of Petri net to fault diagnosis in satellite", *IEEE J. Syst. Eng. and Electronics*, vol. 12, no. 2, pp. 92-96, June 2001.
- [17] J. Lee, M. Zhou, and P. Hsu, "An application of Petri nets to supervisory control for human-computer interactive systems", *IEEE Trans. Ind. Electronics*, vol. 52, no. 5, pp. 1220-1226, Oct. 2005, DOI: 10.1109/TIE.2005.855694
- [18] MicroPrint LC GmbH, *The Pad Printing Book*, MicroPrint LC GmbH, Schaffhausen, Switzerland, 2013.
- [19] Teca-Print, *The Pad Printing Process*, Teca-Print AG, Thaygen, Switzerland, 2006.
- [20] Proell, *Pad Printing-Theory and Practice*, Proell KG, Weissenburg, Germany, 2005.
- [21] A. Gómez, *Materias y Productos en Impresión: Impresión en Serigrafía y Tampografía*. Málaga, España: IC Editorial, 2012.
- [22] T. Eduard, "Relay ladder logic and petri nets for discrete event control design: a comparative study", MSc thesis, New Jersey, USA, 1996.
- [23] R. Zurawski, and M. Zhou, "Petri nets and industrial applications: A tutorial", *IEEE Trans Ind. Electronics*, vol. 41, no. 6, pp. 567-583, Dec. 1994, DOI: 10.1109/41.334574
- [24] C. Ruiz, *Impresión de Productos en Tampografía*. España: IC Editorial, 2013.
- [25] Mascoprint, Screen & Pad Printing Equipment, "Controlling ambient conditions foe better quality," January 2018. [Online]. Available: http://www.mascoprint.co.uk/downloads/resources/pad-printing/Ambient_Conditions.pdf
- [26] Pad Printing Network, "Pad Printing Process: Technical Papers," November 2017. [Online]. Available: http://www.padprinting.net/proc_tec.html
- [27] S. Klein, G. Frey, and M. Minas, "PLC programming with signal interpreted Petri nets", in *Proc. 24th Int. Conf. Applications and Theory of Petri Nets*, Eindhoven, The Netherlands, pp. 440-449, June, 2003, DOI: 10.1007/3-540-44919-1_27
- [28] M. Uzam, and A. Jones, "A new Petri-net-based synthesis technique for supervisory control of discrete event systems", *Turkish J. of Elec. Eng. and Comp. Sc.*, vol. 10, no. 1, pp. 85-109, 2002.
- [29] M. Uzam, A. Jones, and N. Ajlouni, "Conversion of Petri net controllers for manufacturing systems into Ladder logic diagrams", in *Proc. IEEE Conf. Emerging Technologies and Factory Automation*, vol. 2, Kauai, Hawaii, pp. 649-655, Nov. 1996, DOI: 10.1109/ETFA.1996.573978
- [30] M. Heiner, and T. Menzel, "A Petri nets semantics for the PLC language instruction list", in *Proc. IEE Workshop on Discrete Event Systems*, Cagliari, Italy, pp. 161-165, Aug. 1998.
- [31] Siemens, *Control Secuencial: Técnicas de Programación y Manual de Desarrollo, Microsystem SIMATIC S7-200*, Simatic Net, 1999.
- [32] A. Orozco, C. Guarnizo, y M. Holguín, *Automatismos Industriales*. Pereira, Colombia: Taller de Publicaciones-UTP, 2008.
- [33] T. Perme, "Translation of extended Petri net model into Ladder diagram and simulation with PLC", *J. Mech. Eng.*, vol. 55, no. 10, pp. 608-622, 2009.
- [34] K.H. John, and M. Tiegelkamp, *IEC 61131-3: Programming Industrial Automation Systems*. New York: Springer-Verlag, 2001, DOI: 10.1007/978-3-662-07847-1
- [35] J. Chirn, and D. Mcfarlane, "Petri nets based design of Ladder logic diagrams", in *Proc. Control 2000*, Cambridge, UK, Sept. 2000.
- [36] G. Lee, H. Zandong, and J. Lee, "Automatic generation of Ladder diagram with control Petri net", *J. Intelligent Manuf.*, vol. 15, pp. 245-252, Jan. 2004, DOI: 10.1023/B:JIMS.0000018036.84607.37



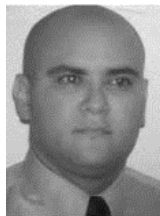
Yamid García Blanco es Ingeniero Mecánico de la Universidad del Atlántico (Puerto Colombia Colombia) graduado en el año 2014 con trabajo de graduación en el área de Automatización de Procesos. Actualmente cursa estudios de posgrado en Ingeniería Mecánica en la Universidad Federal de Tecnología (Paraná, Brasil).



Holmer Ripoll Sierra es Ingeniero Mecánico de la U. del Atlántico (Puerto Colombia Colombia) graduado en el año 2014 con trabajo de graduación en el área de Automatización de Procesos. Holmer es Magister en Ingeniería Mecánica con énfasis en Térmicas y Fluidos en el año 2018. Actualmente trabaja como Analista de Desarrollo de Productos en la empresa Toyama Power Products (Brasil).



Holmes Ripoll Goenaga es Licenciado en Física de la Universidad del Atlántico (Puerto Colombia, Colombia) graduado en el año 1990. Holmes obtuvo su grado de Ingeniero Electrónico en la Universidad del Norte (Puerto Colombia, Colombia), 2012. Es Magister en Instrumentación Científica de la Universidad de Antioquia (Medellín, Colombia), 1996. Actualmente Holmes es Gerente de la Empresa Ripoll Resources S.A. y asesor de la empresa RYMCO (Barranquilla, Colombia).



Javier Roldán Mckinley es Ingeniero Mecánico de la Universidad del Atlántico (Puerto Colombia, Colombia) en el año 2001. Obtuvo grado de Doctor en Ingeniería Mecánica en la Universidad de la Florida (Gainesville, FL-USA) en el año 2007. Actualmente trabaja como profesor asociado del Programa de Ingeniería Mecánica de la Universidad del Atlántico (Puerto Colombia, Colombia). Es integrante activo del grupo de investigación DIMER.



Eugenio Yime Rodríguez es Ingeniero Mecánico de la Universidad del Norte (Puerto Colombia Colombia), 1996. Hizo su Maestría en Ingeniería Mecánica en la U. de los Andes (Bogotá, Colombia), 1999. Recibió grado de Doctor en Automática y Robótica de la Universidad Politécnica de Madrid (Madrid, España) en el año 2008. Actualmente es profesor asociado del Programa de Ingeniería Mecánica de la Universidad del Atlántico (Puerto Colombia, Colombia). Pertenence al grupo de investigación DIMER.