

Metodología para el diseño de autómatas finitos con salidas en lenguaje ladder bajo elestándar IEC 61131-3

Methodology for design of finite automatas in ladder language under the Standard IEC 61131-3

Mauricio Holguín Londoño¹, Álvaro Ángel Orozco Gutiérrez², Andrés Escobar Mejía³

Universidad Tecnológica de Pereira, Pereira, Colombia

mau.hol@utp.edu.co

aaog@utp.edu.co

andreses1@utp.edu.co

Resumen— Los autómatas de estados finitos (AF) constituyen una metodología de análisis de autómatas comúnmente empleada en el diseño de sistemas lógicos secuenciales. Ellas se pueden interpretar como una subclase de las redes de Petri. En los sistemas lógicos secuenciales se distingue entre su funcionamiento síncrono y asíncrono, seleccionando normalmente la operación síncrona con flancos de un reloj, sin embargo en la implementación de AF en lenguaje ladder no es práctico hacer sensible las entradas de un automatismo a los flancos de un reloj. En el presente artículo se opta por un enfoque asíncrono, se discute una metodología general para la implementación de AF con salida en lenguaje ladder y se afronta el reto de eliminar posibles transiciones no deseadas.

Palabras clave— Autómatas finitos, Diagrama escalera, IEC 61131-3, Ladder, Redes de Petri, Sistemas asíncronos.

Abstract— Finite state automatas provide a methodology for automata analysis commonly used in the design of sequential logic systems. They can be interpreted as a subclass of Petri nets. In sequential logic systems distinguishes between synchronous and asynchronous operation, usually choosing synchronous operation with a clock, but in the implementation of finite automatas in ladder language it's not practical to make sensible automata inputs to a clock edges. In this article we present an asynchronous approach, we discuss a general methodology for the implementation of finite automatas with outputs in ladder language and facing the challenge to eliminate any unwanted transitions.

Key Word — Finite automata, IEC 61131-3, Ladder diagram, Petri nets, Synchronous systems

I. INTRODUCCIÓN

Los autómatas finitos con salidas constituyen una de las metodologías de mayor uso para el diseño de sistemas secuenciales de estados finitos con o sin salidas. Desde la perspectiva de la automatización, es de interés los sistemas secuenciales con salida, entre los cuales se encuentran las denominadas Máquina de Mealy y Máquina de Moore [1, 2].

Los autómatas finitos ganan su reputación dentro del diseño de sistemas de eventos discretos gracias a la gran variedad de metodologías disponibles para su diseño, entre las cuales se distinguen el diseño por conjunto de estados y el diseño por complemento [3]. Además se destaca la posibilidad de aplicar análisis sistemático con el fin de aplicar simplificación de los autómatas obtenidos, ya sea con la finalidad de simplificar su análisis, donde se obtiene un autómata simplificado con fines específicos de análisis, o con la finalidad de simplificar la implementación física, donde lo que interesa es conservar la funcionalidad pero con una implementación de menos estados. Entre los métodos empleados para la simplificación de autómatas finitos están la simplificación por estados distinguibles y la simplificación por clases de equivalencia [3]-[5].

Los autómatas cada día alcanzan un tamaño y complejidad considerables, buscando continuamente rutinas de producción con mayor eficiencia y bajo condiciones óptimas de seguridad para las personas y las máquinas. Para cierto rango de aplicaciones pequeñas en términos del número de entradas y salidas, pero con niveles de complejidad propios de los autómatas de mayor tamaño se requiere del diseño inicial bajo la

¹ M.Sc. en Ingeniería Eléctrica

² Ph.D. en Bioingeniería

³ M.Sc. en Ingeniería Eléctrica

metodología de autómatas finitos, pero con despliegue final en lenguaje ladder [6]-[8].

La traducción de autómatas finitos a diagrama ladder se encuentra en varios textos de la literatura, entre ellos en [9, 10]. Sin embargo al realizar implementaciones físicas se debe afrontar varios retos que no son tradicionalmente tratados por los autores. Estos retos incluyen en general la necesidad de eliminar los sistemas secuenciales de entradas síncronas para los autómatas a ser implementados en PLC, ya que en la automatización industrial no es práctico tener sistemas físicos a nivel de potencia que implementen accionamientos tipo flip-flop, igualmente las entradas bajo un esquema asíncrono implican cambios de estado con el simple cambio de una entrada lo cual tampoco es deseable.

Otro reto a direccionar tiene que ver con el enfoque tradicional donde todo un sistema se modela con un único autómata finito, pero en aras de la modularidad, mantenibilidad y escalabilidad es preferible definir un diseño jerárquico donde cada acción individual del sistema es controlada por un autómata específico. Bajo este último enfoque un estado en un autómata puede ser, a su vez, modelado como otro autómata finito.

Los autores presentan en este artículo una metodología generalizada que partiendo de la implementación tradicional de sistemas diseñados bajo un enfoque de autómatas finitos con salida, permite su implementación por traducción en sistemas restringidos a lenguaje ladder y afrontando los retos derivados de tener entradas que se ajusten a sistemas tanto síncronos como no síncronos. Se hace énfasis en un diseño modular y bajo la norma IEC 61131-6.

II. CONTENIDO

A. AUTÓMATAS FINITOS FORMALES

Un autómata finito se define formalmente como el quintuplo:

$$AF = \{Q, \Sigma, \delta, s, F\} \tag{2.1}$$

donde Q es el conjunto de símbolos de estado, Σ es el conjunto de símbolos de entrada, δ es la función de transición $\delta: Q \times \Sigma \rightarrow Q$ que desde un estado y dado un símbolo de entrada genera un nuevo estado siguiente, s es el estado inicial y F es el conjunto de estados finales. Ya que δ es una función, y no solo una relación, para cada estado actual y símbolo de entrada debe existir un único estado siguiente, además δ debe estar definida en cada

estado para cada uno de los símbolos de entrada en Σ . El estado inicial define el estado desde donde inicia el sistema una vez es energizado y los estados finales definen los posibles estados donde la máquina puede terminar para considerar que realizó la tarea propuesta, desde un enfoque informal, o el conjunto de cadenas de símbolos de entrada aceptados, para un enfoque formal [11].

Una de las representaciones más empleadas para un autómata finito consiste de círculos para representar los estados, un círculo doble para los estados finales, el signo $>$ para el estado inicial y arcos para las transiciones entre estados. La gráfica así obtenida recibe el nombre de diagrama de transiciones o diagrama de estados, figura 1.

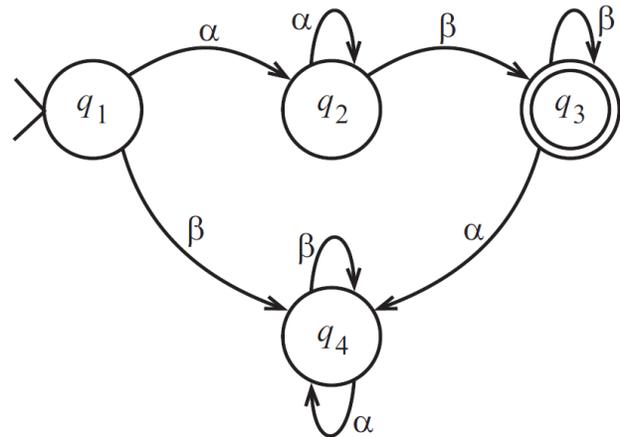


Figura 1. Diagrama de estados o transiciones.

Todo diagrama de estados tiene una tabla de transiciones asociada, la cual se lee por renglones y define para cada estado presente y todo símbolo de entrada el estado siguiente. La tabla 1 es la tabla de transiciones asociada al diagrama de estados de la figura 1.

q	símbolo	$\delta(q_i, \alpha_j)$
q_1	α	q_2
q_1	β	q_4
q_2	α	q_2
q_2	β	q_3
q_3	α	q_4
q_3	β	q_3
q_4	α	q_4
q_4	β	q_4

Tabla 1. Tabla de transiciones.

Formalmente un autómata finito es la representación abstracta que se muestra en la figura 2 y recibe el nombre de modelo en cinta, el cual permite estudiar jerarquías, utilidades y

limitaciones de diferentes tipos de autómatas. En este modelo existen tres componentes: el programa o instrucciones de que dispone el autómata, una cinta que es el medio de entrada y una cabeza de lectura que se desplaza exclusivamente de izquierda a derecha.

La cinta consiste de una serie de cuadros en cada uno de los cuales existe un y sólo un símbolo de entrada. La cabeza lectora se posiciona en el inicio y lee el símbolo del primer cuadro, transfiere este dato al programa quien decide si la cabeza se mueve un lugar a la derecha o permanece quieta. El programa se comporta como un sistema de control que evalúa el estado, si luego de recorrer la cinta de entrada el estado actual es un estado final, entonces se dice que la secuencia de símbolos de entrada es aceptada.

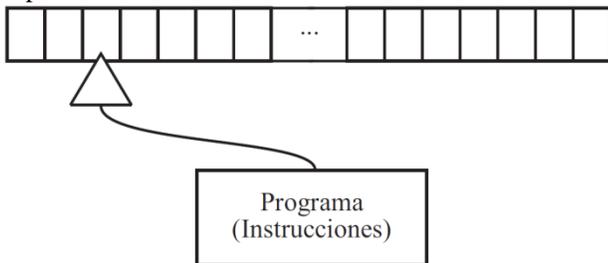


Figura 2. Modelo en cinta de un autómata finito.

B. AUTÓMATAS FINITOS CON SALIDA

Representan el modelo físico de sistemas donde el beneficio está en responder en cada estado adecuadamente de acuerdo al valor del conjunto de entradas. Su desarrollo puede incluir determinar la aceptación de secuencias de caracteres de entrada, pero normalmente se enfocan en una operación cíclica sin interés en la aceptación [3].

1. Autómata de Moore

Autómata finito donde en cada instante de tiempo el valor de sus salidas es función exclusiva del estado presente [2]. Su diagrama de estados, figura 3, adiciona un indicador del valor de salida en cada estado.

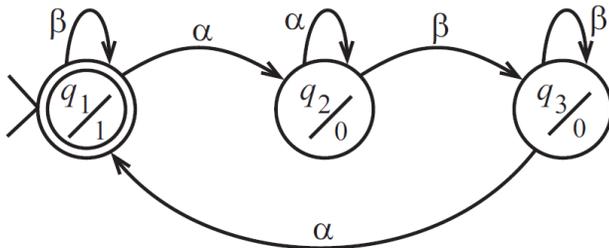


Figura 3. Diagrama de estados de una máquina de Moore.

Formalmente, un autómata de Moore es el séxtuplo:

$$AF_{moore} = \{Q, \Sigma, \Gamma, \delta, \lambda, s\} \tag{2.2}$$

donde Q, Σ, δ y s se definen igual que en la ecuación 2.1. Γ es el conjunto de símbolos de salida y λ es la función de salida que depende sólo del estado actual, es decir, $\lambda(q_i) = v \in \Gamma^*$, donde Γ^* es la cerradura sobre Γ .

2. Autómata de Mealy

Autómata finito donde en cada instante de tiempo el valor de sus salidas depende del estado presente y de la función de transición [1]. Su diagrama de estados, figura 4, adiciona un indicador del valor de salida en cada transición.

Formalmente, un autómata de Mealy es el séxtuplo:

$$AF_{mealy} = \{Q, \Sigma, \Gamma, \delta, \lambda, s\} \tag{2.3}$$

donde $Q, \Sigma, \Gamma, \delta$ y s se definen igual que en la ecuación 2.2, y λ es la función de salida que depende del estado actual y del valor de los caracteres de entrada, es decir, $\lambda(q_i, \alpha_j) = v \in \Gamma^*$.

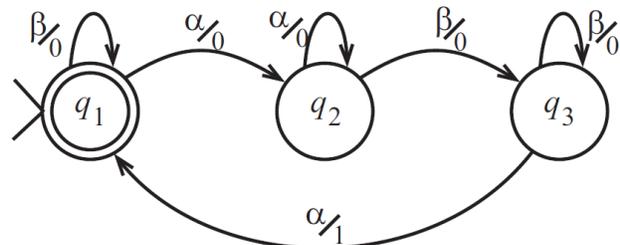


Figura 4. Diagrama de estados de una máquina de Mealy.

III. ETAPAS DEL DISEÑO

Se describe a continuación la forma de implementar cada una de las partes de un autómata finito con y sin salidas y las consideraciones especiales de diseño según su tipo.

A. LOS ESTADOS

Los estados son la única memoria con la que cuentan los autómatas finitos [12, 13]. Por tanto su implementación corresponde con la ecuación de un biestable:

$$q_i = (Set + q_i) \overline{Reset} \tag{3.1}$$

El estado inicial s representa el estado con activación incondicional al instante de la energización, por tanto como al inicio todos los estados están inactivos:

$$q_s|_{t=0} = \prod \bar{q}_i, \quad \forall i \in s \quad (3.2)$$

La acción de activación de estado siguiente (*set*) ocurre simultáneamente con la acción de desactivación de estado previo (*reset*) cuando desde el estado presente se verifica la transición hacia el estado siguiente, es decir el *Set* para q_k ocurre junto con el *Reset* de q_i cuando:

$$\delta(q_i, \alpha_j) = q_k \quad (3.3)$$

B. TRANSICIONES TEMPORIZADAS

La evolución en un autómata finito ocurre cuando se verifica una transición determinada, la cual a su vez es función del estado presente y del símbolo de entrada. Bajo este enfoque, un símbolo representa una entrada en un autómata físico, lo cual frecuentemente se relaciona como simples valores booleanos. Sin embargo es frecuente que la señal de transición sea la respuesta de un estado ante una acción de temporización. La temporización a la energización (tipo ON) se representa:

$$Temp = Set^{te} \quad (3.4)$$

y da como respuesta un uno lógico que se retarda en el tiempo un valor de selección determinado en referencia a la energización [14]. Igualmente, la temporización a la desenergización (tipo OFF) se representa:

$$Temp = Set^{td} \quad (3.5)$$

y da como respuesta un uno lógico que se prolonga en el tiempo un valor de selección determinado en referencia a la desenergización [14].

C. SALIDAS

Las salidas son fundamentales en los autómatas tipo Mealy o Moore. Para los primeros, las salidas son función del estado y del símbolo de entrada, con lo cual su evaluación corresponde con:

$$\lambda(q_i, \alpha_j) = v \in \Gamma^* \quad (3.6)$$

En los segundos, la salida es función exclusiva del estado, por lo que:

$$\lambda(q_i) = v \in \Gamma^* \quad (3.7)$$

Sin embargo, en ambos casos una salida puede presentarse repetidamente en varios estados con lo cual las ecuaciones 3.6 y 3.7 se convierten en:

$$v = \sum_{\lambda=v} \lambda(q_i, \alpha_j) \quad (3.8)$$

$$v = \sum_{\lambda=v} \lambda(q_i) \quad (3.9)$$

donde con $\sum_{\lambda=v} \lambda(\dots)$ se quiere significar que se deben sumar todas las funciones de salida que entregan como resultado el mismo símbolo de salida v .

D. ENTRADAS

Las entradas presentan un gran desafío a la hora de implementaciones físicas, ya que en los modelos de Mealy o Moore un cambio en un valor de entrada, símbolo, produce la evolución inmediata del autómata hacia un posible nuevo estado; sin embargo es deseable que los cambios ocurran a intervalos mínimos que garanticen el cumplimiento de las acciones propuestas de salida.

En los sistemas secuenciales digitales este problema se afronta con los denominados sistemas síncronos, donde las entradas se escanean solo ante los flancos de subida o bajada de una señal de reloj de referencia gracias al empleo de los dispositivos denominados flip-flops.

Para el caso presente, el objetivo es implementar en PLC usando lenguaje ladder, para ello las entradas deben igualmente permitir una evolución gradual del autómata, pero sin empleo de reloj. Para solventar este problema se ha optado por adicionar detectores de flancos de las señales de entrada con lo cual el avance del autómata se puede controlar pausadamente dando el tiempo necesario para la actualización de salidas.

IV. METODOLOGÍA DE DISEÑO

En general la metodología para implementar en ladder un autómata finito con salida se puede resumir de la siguiente forma:

- Identificar el número de entradas, salidas y estados del automatismo
- Definir e implementar el estado inicial
- Construir un diagrama de estados que permita identificar las funciones de transición y salida
- Implementar las equivalencias de las ecuaciones 3.1 a 3.9
- Adicionar detectores de flancos a las entradas

- En el caso de autómatas finitos sin salida, la aceptación ocurre cuando se consumen los símbolos de entrada y el estado actual coincide con uno final

V. EJEMPLO DE APLICACIÓN

Un móvil se desplaza a derecha (Vd) e izquierda (Vi) entre los límites a y b. El pulsador de entrada D activa el movimiento a derecha, el pulsador I activa el movimiento a izquierda y el pulsador A1 es señal de alto total. La figura 5 ilustra lo planteado.

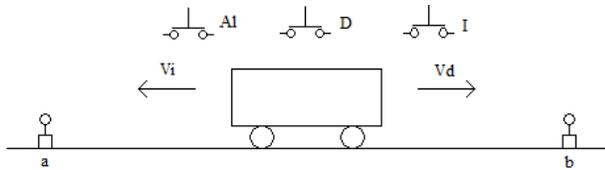


Figura 5. Ejemplo de aplicación

El primer paso de la metodología es identificar el número de entradas, salidas y de estados requeridos, que para este caso son: 5 entradas (los 3 pulsadores y los límites), 2 salidas (el desplazamiento a derecha e izquierda) y 3 estados, así:

- Estado A: define móvil en reposo
- Estado B: define movimiento a derecha
- Estado C: define movimiento a izquierda

En segundo término se define el estado inicial, que para el presente caso es el estado de reposo A.

El tercer paso consiste en obtener un diagrama de estados (figura 6), del cual se puede deducir la tabla de transiciones (tabla 2).

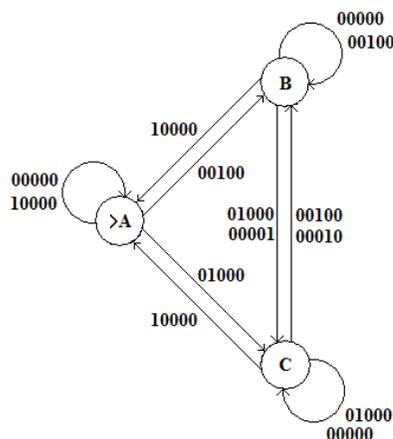


Figura 6. Diagrama de estados del ejemplo

C	A	C	B	B	--	C
---	---	---	---	---	----	---

Tabla 2. Tabla de transiciones del ejemplo

Las ecuación 3.1 y 3.3 definen la activación de estado, que aplicadas a cada uno quedan:

$$q_A = (A\{C1 + C2\} + B.C2 + C.C2 + q_A)\bar{B}. \bar{C}$$

$$q_B = (A.C3 + B\{C1 + C3\} + C\{C3 + C4\} + q_B)\bar{A}. \bar{C}$$

$$q_C = (A.C5 + B\{C5 + C6\} + C\{C1 + C5\} + q_C)\bar{A}. \bar{B}$$

Donde:

$$C1 = \bar{A}. \bar{I}. \bar{D}. \bar{a}. \bar{b}$$

$$C2 = A.I. \bar{D}. \bar{a}. \bar{b}$$

$$C3 = \bar{A}. \bar{I}. D. \bar{a}. \bar{b}$$

$$C4 = \bar{A}. \bar{I}. \bar{D}. a. \bar{b}$$

$$C5 = \bar{A}. I. \bar{D}. \bar{a}. \bar{b}$$

$$C6 = \bar{A}. \bar{I}. \bar{D}. \bar{a}. b$$

La ecuación 3.2 para activación del estado inicial queda:

$$q_A = \bar{B}. \bar{C}$$

El ejemplo es una máquina de Moore, ya que el desplazamiento depende exclusivamente del valor de los estados, así la ecuación 3.9 queda:

$$V_d = B$$

$$V_i = C$$

Para las señales de entrada se debe considerar el tipo de flanco a emplear. Por ejemplo, es común que la orden ingresada mediante un pulsador se tome en el flanco de bajada, es decir cuando se suelta el pulsador y no cuando se presiona. Por el contrario, las señales de los sensores del límite de desplazamiento se deben tomar apenas ocurren, lo cual coincide con el flanco de subida.

La implementación se muestra en la figura 7 usando el software ZelioLogic, disponible libremente en la red [7], y desplegado sobre una unidad Telemecanique SR2B201FU.

	ENTRADAS A1, I, D, a, b					
Estados	10000	01000	00100	00010	00001	00000
A	A	C	B	--	--	A
B	A	C	B	--	C	B

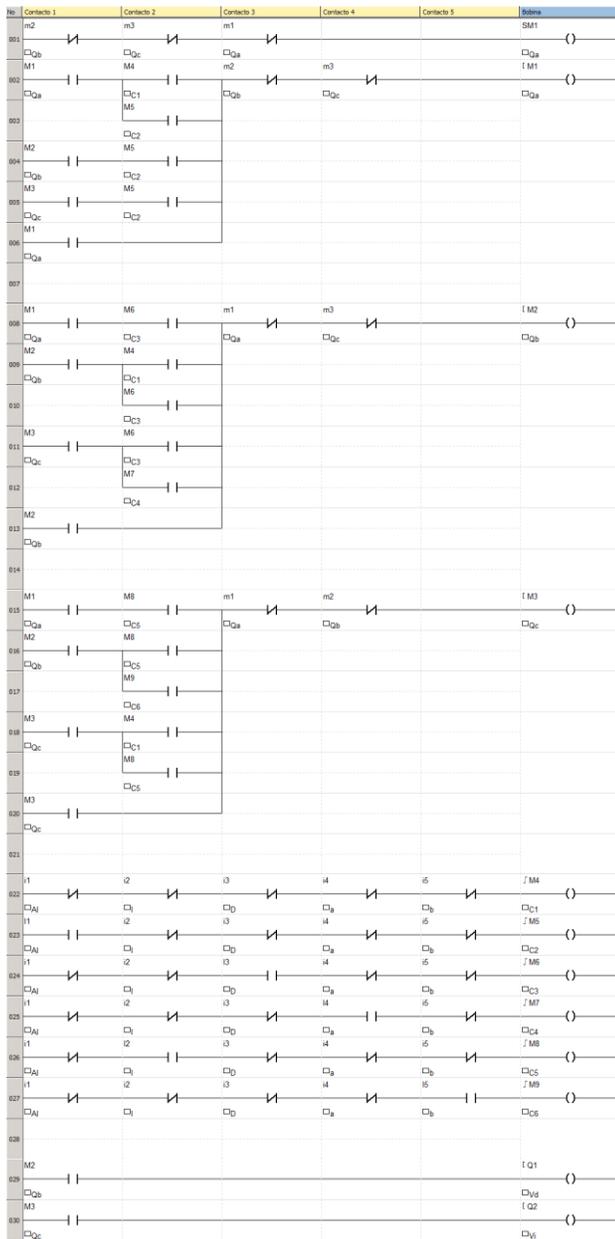


Figura 7. Implementación programada en Ladder

VI. CONCLUSIONES Y TRABAJOS FUTUROS

- Los autómatas finitos constituyen una metodología de diseño que permite el planteamiento de soluciones a problemas de diversa naturaleza y de gran complejidad. Los autómatas con salida adicionalmente facilitan la mantenibilidad y escalabilidad de aplicaciones industriales.
- La implementación de los autómatas finitos con salida en PLCs con baja cuenta de canales de entrada/salida, representa un reto ya que normalmente este tipo de

dispositivos son ajustados para ser programados mediante el lenguaje ladder que define la norma IEC 61131-3. Lo anterior obliga a plantear metodologías sistemáticas de traducción de sistemas complejos que frecuentemente se diseñan con la ayuda de otros enfoques no ladder.

- El diseño ladder es altamente demandante de la experiencia y experticia de los ingenieros de desarrollo. El tiempo para entrega, la facilidad de adicionar restricciones y de escalar un diseño son restrictivas ladder.
- La metodología presentada permite traducir satisfactoriamente todo autómata finito, con y sin salidas, para su implementación en sistemas de desarrollo ladder. Ya que el proceso respeta la misma estructura de los autómatas finitos, la adición de nuevas restricciones y la escalabilidad del diseño original no afectan la topología ladder obtenida.
- Se debe observar como en la figura 7 existen bloques bien definidos para la inicialización del estado inicial, para la activación y desactivación de estados, para las condiciones de las transiciones, para las salidas y para las entradas. Estos bloques son los que permiten una estructura uniforme que no se ve mayormente afectada por cambios en el diseño por diagrama de estados.

REFERENCIAS

[1] Mealy, George H. “A Method to Synthesizing Sequential Circuits” Bell Systems Technical Journal.pp. 1045–1079, 1955.

[2] Moore, E. F. “Gedanken-experiments on Sequential Machines” Automata Studies, Annals of Mathematical Studies, vol. 34, pag. 129–153. Princeton University Press, Princeton, N.J., 1956.

[3] Brena, Ramón. *Autómatas y lenguajes, un enfoque de diseño*. Tecnológico de Monterrey, México, Verano de 2003.

[4] Nelson, Víctor P. *Análisis y diseño de circuitos lógicos digitales: Circuitos de conmutación. Edición Ilustrada*. Texas. Pearson Prentice Hall. Universidad De Arburn. ISBN 968-880-706-0, 1996.

[5] Wakerly, John F. *Diseño digital principios y prácticas: Principios y diseño de lógica secuencial*. Tercera Edición. México. Pearson Educación. Stanford University. ISBN 970-17-0404-5, 2001

[6] Click Koyo, AutomationDirect. Disponible en la red en el URL: http://www.aboutpls.com/click/About_CLICK/Features_and_Benefits/madebykoyo.html

- [7] Relé ZelioLogic. Telemecanique. Disponible en la red en el URL: <http://www.electricidadlynch.com.ar/novedadesreletelemecaniquers.htm>
- [8] 1761 MicroLogix 1000. Programmable controllers Allen-Bradley. Disponible en la red en el URL: <http://www.ab.com/programmablecontrol/plc/micrologix1000/>
- [9] Ramirez-Serrano, Alejandro. “Discrete Event Systems” Implementing Finite State Machines (e.g., supervisory controllers) into PLC Ladder Logic code. ENMF 619.20
- [10] Barragán Piña, Antonio Javier. *Instrumentación y Control Industrial*. Dpto. Ing. Electrónica, Sistemas Informáticos y Automática Universidad de Huelva, España. Disponible en la red en el URL: <http://www.uhu.es/antonio.barragan>
- [11] Holguín, Mauricio. Holguín Germán A. Orozco, Álvaro. *Fundamentos Teóricos para los Automatas Industriales*. Impreso en Taller de Publicaciones de la Universidad Tecnológica de Pereira. Realizado bajo el auspicio de COLCIENCIAS, Proyecto:1110-405-20247: Identificación en línea de modos tempranos de fallas dinámicas en máquinas rotativas. ISBN: 978-958-722-096-4, 2010.
- [12] Hopcroft, John E. Ullman, Jeffrey D. Introduction to automata theory languages, and computation Addison-Wesley, ISBN: 0-210-02988-X, 1993.
- [13] Davis, Martin. Sigal, Ron. Weyuker, Elaine. *Computability, complexity, and Languages Fundamentals of theoretical computer science*. Morgan Kaufmann Publishers, ISBN: 0-12-206382-1, 2003.
- [14] Delhaye, C. *La concepción lógica de automatismos industriales: Relés electromecánicos – Relés electrónicos*. Única Edición. España. Marcombo S.A. ISBN 26.676-1968, 1971.