

IMPLEMENTACIÓN DE UN MICROCONTROLADOR PIC CON EL MÉTODO GRAFCET

Implementing of a PIC microcontroller with GRAFCET method

RESUMEN

Se describe una metodología para programar un microcontrolador PIC 16F84 a partir del Grafcet y se realiza una aplicación práctica de esta metodología a un problema en particular.

PALABRAS CLAVES: PIC, Grafcet, microcontrolador.

ABSTRACT

A methodology is described to program a microcontroller PIC 16F84, starting from the Grafcet of the solution of the problem and is carried out a practical application from this methodology to a particular problem.

KEYWORDS: PIC, Grafcet, microcontroller.

OSIEL ARBELÁEZ SALAZAR

Ingeniero en Control Electrónico e Instrumentación
Profesor Asistente
Universidad Tecnológica de Pereira
osiel@utp.edu.co

JAIRO ALBERTO MENDOZA VARGAS

Ingeniero Electricista, M.Sc
Profesor Asistente
Universidad Tecnológica de Pereira
jam@utp.edu.co

JOSÉ A. MURIEL ESCOBAR

Ingeniero Mecánico
Instructor Sena Industria.
Dosquebradas
jamuriel@sena.edu.co

1. INTRODUCCIÓN

La programación de microcontroladores PIC en lenguaje ensamblador, se hace un poco confusa para las personas no expertas en la programación en este ambiente, ya que se carece por lo general, de un método sistemático en la solución del problema y mucho menos en la codificación del mismo en el lenguaje ensamblador del microcontrolador.

El Grafcet (Gráfico Funcional de Etapas y Transiciones) es un método de descripción funcional de sistemas (procesos) en el curso de su operación, al establecer una correspondencia secuencial (sucesión de etapas) entre las entradas (información) y las salidas (acciones).

Una de las ventajas del Grafcet es que, es un gráfico por medio del cual se puede plasmar la solución de un problema de lógica secuencial de una manera ordenada y muy sencilla; razón por la cual es de mucha utilidad poder programar directamente el microcontrolador a partir de la solución dada en el Grafcet, sin necesidad de acudir a un tipo de conversión o traducción al lenguaje del microcontrolador. Para realizar lo anterior, se necesita conocer solo algunas instrucciones básicas del listado de instrucciones del microcontrolador.

En este documento no se pretende profundizar en el Grafcet ni en la programación de los microcontroladores, sino, mostrar una manera de programarlos apoyándose en una herramienta tan poderosa y versátil como lo es el Grafcet.

Fecha de Recepción: 5 de junio de 2008

Fecha de Aceptación: 4 de Agosto de 2008

2. GRAFCET

El gráfico secuencial de funciones (Grafcet) es un lenguaje gráfico que proporciona una representación en forma de diagrama las secuencias del programa. En este se pueden representar selecciones alternativas de secuencia y secuencias paralelas. Los elementos básicos son etapas y transiciones[2].

Las Etapas que representan cada uno de los estados del sistema. El símbolo para una etapa es un cuadrado con un número o una letra en su interior que la identifica. (ver figura 1)



Figura 1. Etapa

Etapa Inicial aquella en que se posiciona el sistema al iniciarse por primera vez y se representa por un cuadrado con doble línea. (ver figura 2)

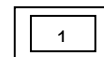
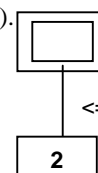


Figura 2. Etapa inicial

Las Líneas de Evolución que unen entre si las etapas que representan actividades consecutivas. Las líneas se entenderán siempre orientadas de arriba hacia abajo, a menos que se represente con una flecha en sentido contrario. (ver figura 3).



<==Línea de evolución

Figura 3. Líneas de evolución

Las Transiciones que representan las condiciones lógicas necesarias para que finalice la actividad de una etapa y se inicie la etapa o etapas inmediatamente consecutivas. Las transiciones se representan gráficamente por una línea cruzada sobre las líneas de evolución.(ver figura 4)

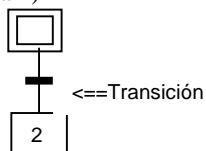


Figura 4. Transición

Los Reenvíos son símbolos en forma de flecha que indican la procedencia o destino de las líneas de evolución. Las flechas de reenvío permiten fraccionar un gráfico o evitan dibujar líneas de evolución con excesivos cruces.(ver figura 5)

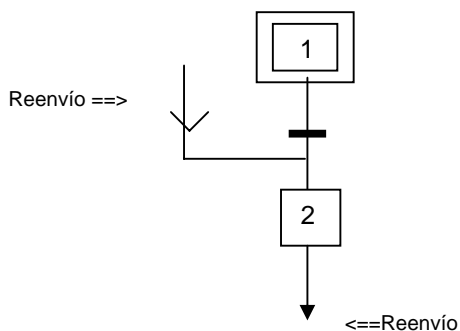


Figura 5. Reenvío

Las acciones asociadas a cada etapa indican cual es la actividad a desarrollar cuando ésta se active(ver figura 6).

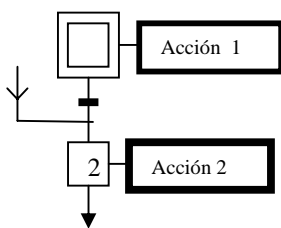


Figura 6. Acciones asociadas

Reglas de evolución:

- **Regla 1: etapa inicial.**
- **Regla 2: Franqueo de una transición.**

El franqueo de una transición no se produce hasta que dicha transición sea validada y además su

receptividad asociada sea verdadera (se verifique o cumpla).

Se dice que una transición está validada cuando la etapa (o las etapas) precedente está activa.

- **Regla 3: Evolución de las etapas activas.**

El franqueo de una etapa provoca simultáneamente

- La desactivación de la etapa (o de las etapas, en caso anterior).
- La activación de la etapa (o de las etapas) siguiente.

3. LOS MICROCONTROLADORES PIC

Un microcontrolador es un circuito integrado que contiene todos los componentes de un computador. Se emplea para controlar el funcionamiento de una determinada tarea y, debido a su reducido tamaño, suele ir incorporado en el propio dispositivo que el gobierna. Esta última característica es la que le confiere la denominación de “controlador incrustado” [1].

En su memoria solo reside un programa destinado a gobernar una aplicación determinada; sus líneas de entrada / salida soportan el conexionado de sensores y actuadores del dispositivo a controlar. Una vez programado el microcontrolador solamente sirve para gobernar la tarea asignada.

La programación de los microcontroladores se realiza en lo que llaman lenguaje ensamblador, el cual es un lenguaje de bajo nivel (más cercano a la máquina) y que es propio del microcontrolador (fabricante).

Los programas bien realizados en lenguaje ensamblador optimizan el tamaño de la memoria que ocupan y su ejecución es muy rápida. Los lenguajes de alto nivel más empleados con microcontroladores son el C y el Basic.

Los microcontroladores PIC, son una familia de dispositivos fabricados por la firma Microchip, los cuales son muy populares debido a sus prestaciones, bajo costo y reducido número de instrucciones, el cual consta de 35 instrucciones, con las cuales se realiza toda la programación de las tareas del microcontrolador, el manejo de éstas instrucciones y el conocimiento de la arquitectura del microcontrolador aseguran una programación efectiva.

Éstas instrucciones por ser de bajo nivel, no son muy claras para las personas que inician la programación de los microcontroladores, creando con ello un doble problema en la programación, como lo es el resolver el

problema de control en si mismo y la programación en el microcontrolador.

Para mostrar en que consiste la metodología presentada en este artículo vamos a ver unos ejemplos de aplicación.

4. EJEMPLO DE APLICACIÓN I

Sea el proceso de taladro mostrado en la figura 7.

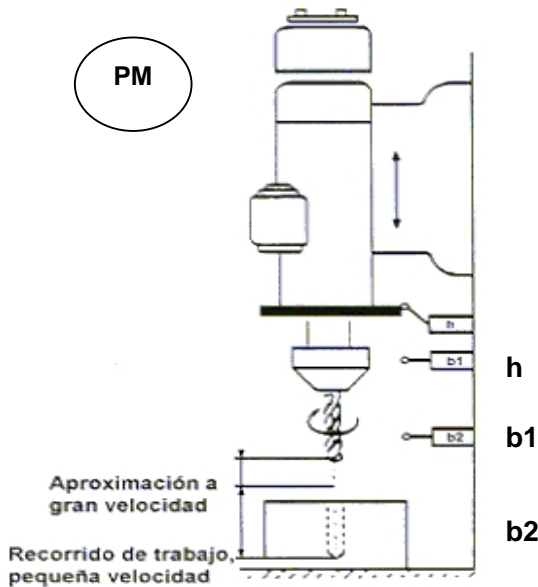


Figura 7 Sistema de perforación

El proceso deberá comenzar con el taladro en posición alta h. Activando el pulsador de marcha (PM), el taladro descenderá a dos velocidades diferentes, a gran velocidad hasta llegar a la posición de b1 y a baja velocidad hasta b2. El proceso concluirá con un retroceso a gran velocidad hasta la posición de inicio h.

Se desea programar el microcontrolador (µC) 16F84 de tal manera que las entradas correspondan al puerto A y las salidas al puerto B, de acuerdo a la tabla 1:

	Entrada puerto A	Salida Puerto B
Final de carrera h	RA0	
Pulsador de inicio PM	RA1	
Final de carrera b1	RA2	
Final de carrera b2	RA3	
Subir taladro Rápidamente		RB0
Bajar taladro Rápidamente		RB1
Bajar taladro Lentamente		RB2

Tabla 1. Acciones, entradas, salidas del microcontrolador

El Grafcet correspondiente a la solución se representa en la figura 8, así:

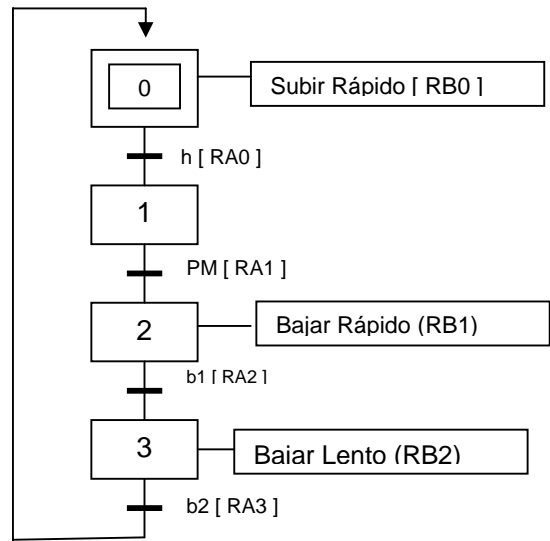


Figura 8. Grafcet ejemplo aplicación 1

Una vez se tiene la solución en el Grafcet, se procede a la codificación en el lenguaje ensamblador propio del microcontrolador.

El repertorio de instrucciones que se utilizan es casi el mismo para cada etapa, lo que permite que su codificación y revisión sea muy fácil de interpretar por personas no expertas en este ambiente de programación, pero que con el Grafcet como punto en común, fácilmente se puede generar un puente de comunicación estructurado en relación a la programación, entre experto y los no expertos.

Atendiendo a las reglas de evolución del Grafcet, hay que tener presente la etapa inicial, las etapas que se activan y las que se desactivan con el disparo de las transiciones, al igual que las acciones asociadas a cada etapa y la posterior verificación de la transición, para poder tomar la decisión de saltar a la etapa siguiente o permanecer en la etapa actual. Este procedimiento se repite indefinidamente, generando un loop en la etapa que se encuentre activa, el cual se rompe solo cuando se cumple la condición de transición.

La configuración preliminar del microcontrolador (entradas, salidas, acumulador, estado, tmr_opt, etc.), es particular dependiendo de la aplicación que se vaya a dar al dispositivo; seguidamente solo se explicará la codificación del programa y su relación directa con el Grafcet (ver figura 9).

```

===== CONFIGURACIÓN DE PIC =====

list p=16f84      ;tipo de µC
radix hex        ;base numérica para trabajar

ptob equ 0x06    ;se le asigna al puerto b la dirección 06h
ptoa equ 0x05    ;se le asigna al puerto a la dirección 05h
estado equ 0x03  ;el registro estado es la dirección 03h
w equ 0x00       ;el registro de trabajo es la dirección 00h

reset org 0      ;salto del vector de interrupciones
goto inicio
org 7

inicio bsf estado,5 ;configuración de las
movlw b'00000000' ;entradas y salidas
movwf ptob ;se configura el puerto b como salida
movlw 0xff ;se configura el puerto a como entrada
movwf ptoa
bcf estado,5

;
; Este programa controla un sistema de perforación de piezas
;
; ===== DESDE AQUI EL PROGRAMA GRAFCET =====
;

etapa0 movlw b'00000001' ;la etapa 0 sube rápidamente el taladro
movwf ptob ; se explora la transición h (RA0) si es
btfss ptoa,0 ; 1 salta a la etapa 1 y si es 0 vuelve a la
goto etapa0 ; etapa 0 haciendo un loop.

etapa1 movlw b'00000000' ;la etapa 1 desactiva todas las salidas
movwf ptob ;espera la orden del pulsador PM, se
btfss ptoa,1 ;explora la transición RA0 y si es 1 salta
goto etapa1 ;a la etapa 2 y si no se queda en la 1

```

Figura 9. Configuración del microcontrolador

Como se puede observar, la codificación muestra una cierta estructura y que fácilmente se asocia al Grafcet del problema, con lo cual se lleva de alguna manera todas las ventajas sistemáticas del Grafcet a la programación del microcontrolador.

Para dar mayor claridad analicemos la etapa número 2 del Grafcet de la figura 8 y el código generado para éste. Se observa en el Grafcet que cuando se activa la etapa 2 se realiza la acción de poner un 1 en el bit RB1 del microcontrolador (bajar rápido), lo que se logra con las instrucciones MOVW y MOVWF; seguidamente se explora la transición RA2 (que se hace con la instrucción BTFSS) y si es 0 (falsa), permanece en la etapa 2 (que se hace con GOTO a etapa 2 nuevamente) de lo contrario salta a la etapa 3. se puede observar que los enlaces entre etapas del Grafcet se hacen en el microcontrolador con GOTO, la validación de la transición en el Grafcet se hace en el microcontrolador con la instrucción BTFSS.

Lo que para el Grafcet es permanecer en una etapa hasta que se cumpla la condición de transición, en el

microcontrolador es un loop en la misma etapa, hasta que la condición permita el salto a otra etiqueta.

5. EJEMPLO DE APLICACIÓN II

Se quiere mejorar el sistema automático de perforación de piezas planteado en el Ejemplo de Aplicación I, agregando un botón de paro prioritario mientras se realiza la perforación de la pieza, de tal forma que este paro (Stop) suba rápidamente el taladro y deje el sistema listo para una próxima operación. El dato de entrada del botón de paro (Stop) se detectará por el puerto A del microcontrolador y en el bit RA4 (ver figura 10).

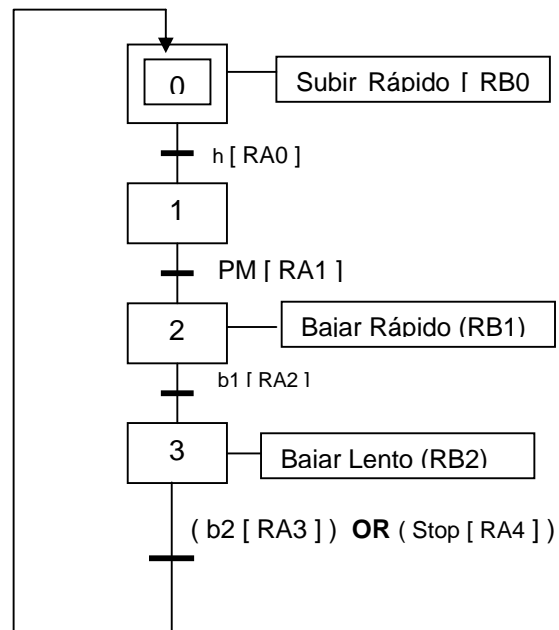


Figura 10. Ejemplo de aplicación 2

Se puede observar que el Grafcet del Ejemplo de Aplicación II (figura 10), no cambió sustancialmente, ya que solamente se agregó la opción de paro prioritario como una transición desde la etapa 3 a la etapa inicial (etapa 0).

Si se desea hacer el código en ensamblador para el microcontrolador, no topamos con que hay que hacer una lectura simultánea de el bit RA3 y RA4 del puerto A, ya que las condiciones de transición así lo exigen, pero el microcontrolador no trabaja lecturas paralelas, salvo en el caso que se lea todo el puerto directamente y se enmascaren los datos que se quieren evaluar; pero esta tarea que para un experto es fácil; para los que nos son expertos se vuelve un problema mas complejo.

Por otra parte, si lo que se pretende es que la codificación en el lenguaje ensamblador del microcontrolador, conserve la misma estructura que el Grafcet del problema hay que explorar primero una transición (la prioritaria) y si es falsa (no se ha disparado), pase a otra etapa y

ejecute la acción correspondiente y posteriormente explore la otra condición de transición del proceso (b2).

Para este problema en particular se rediseña el Grafcet teniendo en cuenta lo anteriormente citado respecto a la lectura secuencial de las transiciones (ver figura 11).

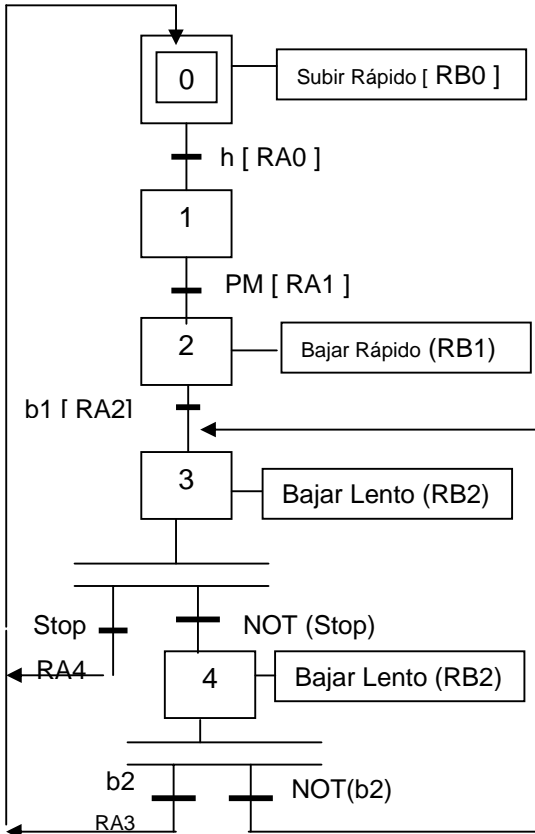


Figura 11. Grafcet adaptado para el Ejemplo de Aplicación II

Como se puede observar en el Grafcet se adaptó (sin ser estrictos con la norma IEC 848), la etapa 3 se extendió a otra etapa (etapa 4), pero pasa de la etapa 3 a la 4 solo después de validar la transición de paro prioritario. Ya en la etapa 4 se ejecuta nuevamente la acción de bajar lento el taladro, pero solo explora la condición de transición (finalización de la perforación b2) y si NO se ha disparado, regresa a la etapa 3, volviendo a ejecutar la acción de bajar lento, pero ahora explorando nuevamente la validación o No de la transición Stop ([RA4]).

Hay que aclarar que este tipo de gráficos Grafcet, como el de la figura 11, son un poco *sui generis*, ya que se está particularizando su aplicación al caso de los microcontroladores, violando la normatividad de la independencia de la tecnología de Grafcet. Sin embargo, lo que se pretende es tener como base un método estructurado de solución de problemas de carácter secuencial para los microcontroladores, que le dé una estructura fácil de comprender por los no expertos en la programación de los microcontroladores.

El código en lenguaje ensamblador para esta solución en el microcontrolador PIC 16F84 se muestra a continuación ver figura 12).

```

===== CONFIGURACIÓN DE PIC =====

list p=16f84           ;tipo de uC
radix hex              ;base numérica para trabajar

ptob equ 0x06          ;se le asigna al puerto b la dirección 06h
ptoa equ 0x05          ;se le asigna al puerto a la dirección 05h
estado equ 0x03        ;el registro estado es la dirección 03h
w equ 0x00             ;el registro de trabajo es la dirección 00h

reset org 0            ;salto del vector de interrupciones
goto inicio
org 7

inicio bsf estado,5    ;configuración de las
        movlw b'00000000' ;entradas y salidas
        movwf ptob        ;se configura el puerto b como salida
        movlw 0xff        ;se configura el puerto a como entrada
        movwf ptoa
        bcf estado,5

;-----
; Este programa controla un sistema de perforación de piezas
;-----
; ===== DESDE AQUI EL PROGRAMA GRAFCET =====
;-----

etapa0 movlw b'00000001' ;la etapa 0 sube rápidamente el taladro
        movwf ptob        ; se explora la transición h (RA0) si es
        btfss ptoa,0      ; 1 salta a la etapa 1 y si es 0 vuelve a la
        goto etapa0      ; etapa 0 haciendo un loop.

etapa1 movlw b'00000000' ;la etapa 1 desactiva todas las salidas
        movwf ptob        ;espera la orden del pulsador PM, se
        btfss ptoa,1     ;explora la transición RA0 y si es 1 salta
        goto etapa1      ;a la etapa 2 y si no se queda en la 1

etapa2 movlw b'00000010' ;activa el descenso rápido hasta que se
        movwf ptob        ;accione el final de carrera b1luego se
        btfss ptoa,2     ;explora el final de carrera b1 (RA2), si
        goto etapa2      ;es 1 salta a la etapa 3 y si NO se queda

etapa3 movlw b'00000100' ;se activa el descenso lento y se explora
        movwf ptob        ;el paro prioritario (stop [RA4]) si es 1
        btfss ptoa,4     ;salta la instrucción y va a la etapa0,
        goto etapa4      ;de lo contrario va a la etapa 4
        goto etapa0

etapa4 movlw b'00000100' ;se repite la acción de la etapa3, se
        movwf ptob        ;explora el final de carrera b2 [RA3], si
        btfss ptoa,3     ;es 1 salta la instrucción y va a la etapa0
        goto etapa3      ;de lo contrario ejecuta la instrucción
        goto etapa0      ;siguiente que lo regresa a la etapa 3
    
```

Para el caso donde la transición sea la simultaneidad de dos variables de entrada (AND), se puede plantear el siguiente Grafcet (ver figura 13).

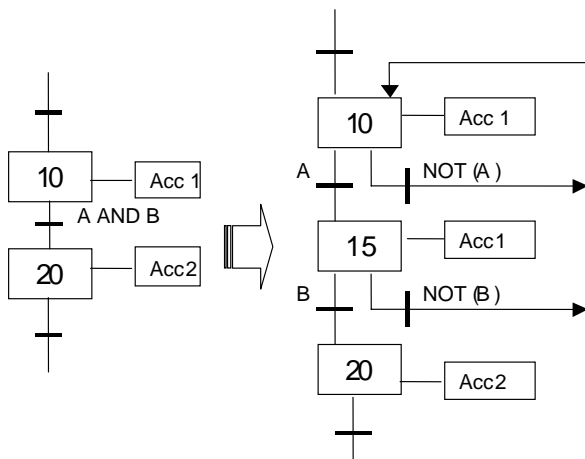


Figura 13. Grafcet para estructura AND

Se observa que la estrategia es similar a la utilizada en el Ejemplo de Aplicación II. analicemos el Grafcet de la figura 13, si el proceso está en la etapa 10, primero se explora la condición de transición A y si es verdadera pasa a la etapa 15, ya en ésta etapa se explora la condición B y si es verdadera pasa a la etapa 20, indicando que tanto la condición A como la B son verdaderas simultáneamente (A and B).

En caso que el proceso esté en la etapa 10 y al explorar la condición A sea falsa, retorna a la misma etapa 10 sin siquiera explorar la otra condición (condición B), ya que se necesita que ambas condiciones (A y B) sean verdaderas simultáneamente, entonces no es necesario explorar la condición B.

Mientras se realiza la exploración de la simultaneidad de las condiciones A y B, en cada etapa se debe ejecutar la acción correspondiente, como lo muestra el Grafcet de la figura 4.

6. PRUEBAS Y RESULTADOS

La metodología se ha probado con estudiantes que no tienen ningún conocimiento en la programación de microcontroladores pero si conocen bien el Grafcet como herramienta de solución de problemas referentes al diseño de sistemas control secuencial.

Los resultados han sido satisfactorios, en la medida que se parte de la solución del problema independiente de la tecnología, pero que su aplicación a la implementación en los microcontroladores es fácil para los estudiantes que no poseen ningún conocimiento sobre la programación de los microcontroladores.

Los estudiantes se preocupan más por la solución del problema, que por la misma codificación del programa, ya que son pocas las instrucciones que se utilizan en el programa y conservan a demás la misma estructura.

También pueden abordar problemas más complejos sin necesidad de utilizar esta metodología, pero ésta les enseña a pensar en un orden estructurado en la programación del microcontrolador.

7. CONCLUSIONES

- Se encontró gran aplicación de la metodología en la enseñanza de los microcontroladores y su programación de una forma más sencilla para personas no expertas en dichos dispositivos.
- La metodología al estar apoyada en la realización del Grafcet, hace que se piense inicialmente solo en la solución del problema, independiente de la tecnología en la que va a ser implementada posteriormente; lo cual se traduce en una ventaja a la hora de querer abordar otras tecnologías para la implementación de la solución
- El código de programación del microcontrolador presenta una estructura organizada y con mucha relación al Grafcet del problema, con lo cual se gana tiempo a la hora de la depuración de los errores y el análisis del programa.
- Una desventaja es que el código resulta ser mucho mas largo que si se realiza de otra manera (capacidad de memoria de programa), pero el punto a favor, es que para aplicaciones de control secuencial (control de procesos industriales) por lo general los tiempos de ejecución de las tareas son relativamente largos frente al tiempo de escaneo del microcontrolador, con lo cual el ahorro de instrucciones, frente a la facilidad de programación con esta metodología no es significativo.
- Como toda herramienta, no es 100% falible, pero como estrategia inicial en la enseñanza de la programación de los microcontroladores es muy poderosa, ya que le da sentido funcional a la solución de problemas de control secuencial.

8. REFERENCIAS

- [1] J.M. Angulo, I Angulo, "Microcontroladores PIC diseño y aplicaciones", McGraw Hill, 2^{da} Edición 1999
- [2] J. Balcells, J.L. Romeral, "Autómatas Programables", Alfaomega-Marcombo, 1998
- [3] R. W. Lewis. "Programming industrial control systems using IEC 1131-3". IEE Control Engineering Series 50. 1998.
- [4] Peña Juan Domingo."Autómatas programables". Introducción a los autómatas programables". Biblioteca Multimedia Industria. Editorial UOC, 2003.
- [5] Alfonso Manuel."Teoría de autómatas y lenguajes formales". Editorial Mc Graw Hill. 1999.
- [6] Bocksnick, Bernd "Fundamentos de la técnica de mando". Festo. 1990
- [7] Bollinguer, J. G. y Duffie, N. A. "Computer control of Machines and Process". Addison- Wesley. 1988
- [8] Mayol, A. "Autómatas programables". Marcombo. 1990.