

# Desarrollo de control de motores a pasos por hardware utilizando microcontroladores PSoC

*E. Machado-Díaz<sup>1</sup>, H. Coto-Fuentes<sup>2</sup>*

**Resumen**—Los motores a pasos son uno de los actuadores más utilizados en los diversos sistemas automáticos, principalmente aquellos en que la precisión es el objetivo principal, como lo pueden ser los sistemas de control numérico, los brazos manipuladores industriales, entre otros. El control electrónico de estos actuadores generalmente se realiza a través de dispositivos especializados para este propósito, que requieren la generación de un tren de pulsos ocupando recursos como: interrupciones, contadores, registros de desplazamiento, etc.

Con la mejora en la elaboración de sistemas en un solo chip, como lo son la plataforma PSoC de Infineon, es posible realizar la generación de las señales cuadradas utilizando los elementos de hardware digital dentro del circuito integrado, optimizando el diseño de control eléctrico de motores, pues permite dividir la parte del manejo de los actuadores de la lógica general del programa. En el marco antes descrito, se desarrollaron y evaluaron varias lógicas para la generación de las señales necesarias (pulsos y dirección) utilizando elementos de hardware de electrónica digital integrados dentro de un microcontrolador PSoC 5, siendo probados en el control de motores a pasos a través de drivers industriales.

**Palabras claves**—controlador, electrónica, microcontrolador, motor a pasos, PSoC.

**Abstract**— Stepper motors are one of the most used actuators in various automatic systems, mainly those in which precision is the main objective, such as numerical control systems, industrial manipulator arms, among others. The electronic control of these actuators is usually done through specialized devices for this purpose, which requires the generation of a pulse train using resources such as: interrupts, counters, shift registers, etc.

With the improvement in the development of systems on a single chip, such as Infineon's PSoC platform, it is possible to generate square signals using digital hardware elements within the integrated circuit, optimizing the design of electrical motor control, since it allows dividing the part of the handling of the actuators from the general logic of the program. Within the framework described above, several logics for the generation of the necessary signals (pulses and direction) were developed and evaluated using digital electronics hardware elements

integrated within a PSoC 5 microcontroller, being tested in the control of stepper motors through of industrial conductors.

**Keywords**—Control, Electronics, Microcontroller, Step Motor, PSoC.

## I. INTRODUCCIÓN

Uno de los principales problemas que se han tenido en ingeniería a lo largo de la historia es la transformación de movimientos angulares en movimientos lineales o realizar acciones de forma precisa a través de sistemas automáticos. Actualmente se utilizan diferentes tipos de actuadores electromecánicos para este fin, entre ellos se encuentran los motores a pasos que permiten realizar movimientos con alta precisión y de una forma relativamente sencilla [1].

Debido a su popularidad y amplia área de aplicación, existen diferentes dispositivos y bibliografía para el uso y control de los motores a pasos. Sin embargo, debido a la naturaleza de su control, la mayoría de los métodos que se utilizan para este fin conllevan también un alto consumo de recursos de los dispositivos de control encareciendo y complicando su implementación en sistemas que pretenden tener a fin un bajo costo. Por ello, se plantea un método alternativo a través de lógica combinatorial en Sistemas de un solo Chip (PSoC) de la marca Infineon que permita controlar la señal de pulsos necesaria para el movimiento de motores a pasos de tipo bipolar pues son éstos los que más complejidad se requiere para su control de movimiento, utilizando también un driver de potencia DRV8825.

### A. Motores a pasos

Los motores a pasos son una subcategoría de los motores de imanes permanentes. En lo que se refiere a la conversión de energía eléctrica a movimientos angulares, los motores a pasos trabajan bajo el mismo principio de los motores de corriente continua y alterna. Sin embargo, la conmutación de este tipo de motores se logra a través de sistemas programables, los cuales deben de generar una secuencia de pulsos para alimentar los devanados internos del motor [2].

Dependiendo de las entradas de control, el eje motriz es capaz de avanzar un número concreto de grados según la secuencia de activación de devanados que se indique en el

<sup>1</sup>Instituto Tecnológico Superior de Lerdo, División de Ingeniería en Sistemas Automotrices, .Av. Tecnológico No. 1555 Sur Periférico Gómez - Lerdo Km. 14.5, Ciudad Lerdo, Dgo. C.P. 35150. \* [eduardo.md@itslerdo.edu.mx](mailto:eduardo.md@itslerdo.edu.mx)

<sup>2</sup>Instituto Tecnológico Superior de Lerdo, División de Ingeniería en Electrónica, Av. Tecnológico No. 1555 Sur Periférico Gómez-Lerdo Km. 14.5, Ciudad Lerdo, Dgo. C.P. 35150.

controlador. La cantidad de rotación es directamente proporcional al número de pulsos y la velocidad de rotación es relativa a la frecuencia de éstos. Entre las ventajas que se tienen al utilizar este tipo de actuadores se encuentran:

- Diseño sencillo y de bajo costo.
- Son altamente confiables
- Relativamente poco mantenimiento.
- Se utilizan en lazo abierto.

Los motores a pasos, en comparación con los de corriente continua de escobillas encontrados de forma típica, no son de una compleja construcción, pero requieren de mejor precisión. Según criterios básicos, los tipos de motores a pasos se pueden dividir de acuerdo a su constitución y número de fases necesarias para alimentar las bobinas internas.

1) *Motores a pasos unipolares.*

Este tipo de motores se componen de dos bobinas, cada una con una derivación en el centro. Las derivaciones del centro son llevadas fuera del motor mediante cables separados o conectados internamente y exponiéndolos como uno de los cables por lo que pueden variar en el número de cables que tienen (generalmente 5 o 6). Sin embargo, sin importar el número de conductores todos los motores de este tipo tienen el mismo principio de funcionamiento [3].

El cable central está conectado a la fuente de alimentación, mientras que los conductores de las bobinas se conmutan a tierra de forma alternada para genera el movimiento. El ejemplo de conexión de un motor a pasos unipolar se muestra en la Figura 1.

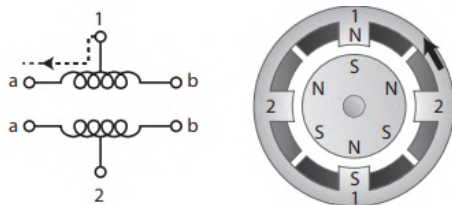


Figura 1. Distribución de bobinas de motor a pasos unipolar.

Al tener varios devanados, para producir el avance deben de ser alimentados en una secuencia adecuada. Al invertir el orden, el giro del motor cambia de dirección [4]. Se pueden realizar diferentes tipos de secuencias de acuerdo a lo que se busque, ya sea velocidad o torque. En la Tabla I se muestra la secuencia simple para el movimiento de un motor a pasos unipolar en donde solamente se alimenta una bobina a la vez.

TABLA I. SECUENCIA SIMPLE PARA MOTOR A PASOS UNIPOLAR

Paso	Bobinas			
	A	C	B	D
1	+	-	-	-
2	-	+	-	-
3	-	-	+	-
4	-	-	-	+

2) *Motor a pasos bipolar.*

En los motores a pasos bipolares, el cambio de dirección del flujo magnético para hacer girar el motor en una dirección u otra se hace mediante el cambio de polaridad en la corriente aplicada a las bobinas. Generalmente, para cambiar esta polaridad en la corriente se utilizan cuatro interruptores homologando a un puente H. En la Figura 2 se muestra la estructura de conexión interna de un motor a pasos bipolar.

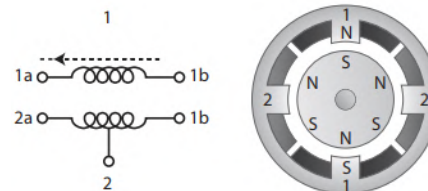


Figura 2. Distribución de bobinas de un motor a pasos bipolar.

Los motores a pasos bipolares tienen aproximadamente un 30% más de toque en comparación con un motor de tipo unipolar del mismo tamaño. En desventaja, los motores bipolares requieren de circuitos de control más complejos que los motores unipolares, lo que equivale a un mayor costo de aplicación.

B. *Drivers de potencia para motores a pasos*

Un controlador o driver de potencia para motores es un amplificador de corriente que permite tomar las señales de control (de baja corriente) para convertirlas en señales de corrientes más altas que permitan la activación de los devanados de potencia. Este tipo de dispositivos son sistemas por lo que contienen diferentes elementos que permiten realizar las tareas que se han determinado. La estructura básica de un driver de potencia para motores a pasos se puede ver en la Figura 3 [5].

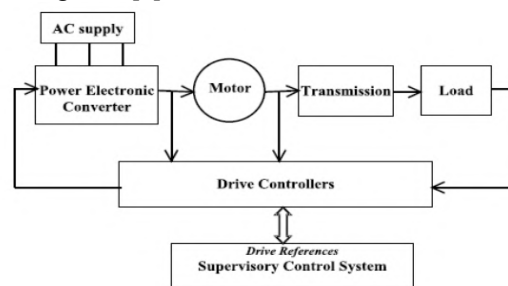


Figura 3. Estructura general de un driver de potencia para motores a pasos

Si bien existen diferentes tipos de controladores, debido al incremento en los sistemas de control numérico e impresoras 3D se han popularizado el modelo DRV8825 que integra dos puentes H de tecnología CMOS, detección de corriente, circuitos de regulación de voltaje y un indexado a micro pasos que permiten realizar movimientos precisos sin requerir que el procesador controle los niveles de corriente. La estructura y distribución de conexiones de este dispositivo de control se muestran en la Figura 4.

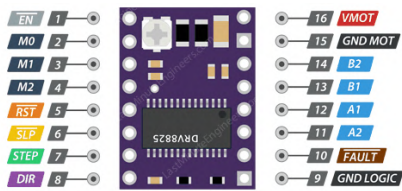


Figura 4. Driver DRV8825.

El controlador cuenta con tres entradas de selección para definir la resolución de los pasos. Al configurar de manera apropiada los niveles lógicos en estas entradas se pueden configurar a las resoluciones que se muestran en la Tabla II.

TABLA II. CONFIGURACIÓN DE RESOLUCIÓN PARA DRIVER DRV8825

M0	M1	M2	Resolución
Bajo	Bajo	Bajo	Paso completo
Alto	Bajo	Bajo	Medio paso
Bajo	Alto	Bajo	1/4 paso
Alto	Alto	Bajo	1/8 paso
Bajo	Bajo	Alto	1/16 paso
Alto	Bajo	Alto	1/32 paso
Bajo	Alto	Alto	1/32 paso
Alto	Alto	Alto	1/32 paso

Una vez que se ha configurado la resolución a la que el motor va a trabajar se deben de enviar trenes de pulsos a la entrada STEP que controla los pasos del motor, cuando más rápido sean estos pulsos, más rápido girará el motor. La entrada DIR controla la dirección del giro de acuerdo al estado bajo o alto [6].

### C. Dispositivos PSOC.

La palabra PSOC es el acrónimo de (Programmable System on Chip), estos dispositivos fabricados por la empresa Infineon son microcontroladores cuya principal característica y atractivo es el contar con módulos tanto analógicos como digitales en un solo chip, así mismo poder reconfigurar dinámicamente las entradas y salidas de estos módulos [7]. De esta manera se obtiene un componente electrónico con una gran flexibilidad en su estructura y que permite una gran facilidad para el desarrollo de sistemas electrónicos embebidos.

Este tipo de dispositivos contienen diferentes áreas principales que hacen que tenga versatilidad al momento de implementarlo para diferentes aplicaciones. Cuenta con bloques digitales y analógicos que pueden configurarse y adaptarse de acuerdo a las tareas que se quieran realizar. Las cuatro áreas principales que componen a un microcontrolador PSOC son: Núcleo, Sistema digital, sistema analógico y recursos del sistema.

De acuerdo a la versión del microcontrolador que se tenga, se contarán con diferentes prestaciones. Este documento está enfocado en la familia PSOC 5, especialmente en la tarjeta de

desarrollo CY8CKIT-059 PSOC 5 LP que se muestra en la Figura 5.



Figura 5. Tarjeta CY8CKIT-059.

Las prestaciones generales en forma gráfica de este dispositivo se presentan en la Figura, internamente cuenta con un gran número de entradas y salidas digitales de propósito general, así como sistemas de comunicación UART, relojes internos, procesadores analógicos y bloques digitales [8].

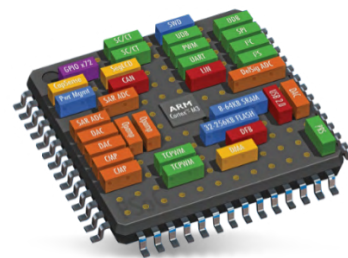


Figura 6. Configuración interna del microcontrolador PSOC 5

## II. PARTE TÉCNICA DEL ARTÍCULO

En la realización de este proyecto se evaluaron diferentes configuraciones de bloques digitales con la finalidad de evaluar los recursos del sistema para usarlos de una forma más eficiente.

### A. Pruebas iniciales utilizando programación estructurada

Como se ha mencionado en el apartado de introducción, se utilizará un Driver DRV8825 como dispositivo de control de potencia. De acuerdo al fabricante es necesario utilizar un microcontrolador o generador de pulsos externo que en este caso será el PSOC5.

El primer paso para el control de pulsos es la configuración de los pines de la tarjeta como salidas, debido a la configuración del controlador se requieren dos: un pin para establecer la dirección de giro de motor y otro para la generación de los pulsos. En la Figura 7, se presenta la configuración realizada para el uso de salidas digitales.

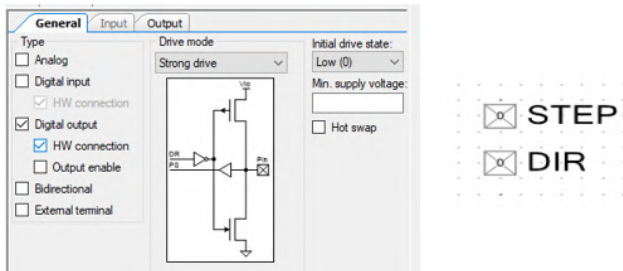


Figura 7. Configuración de salidas digitales

Es importante tomar en cuenta que lo que se busca es realizar el control utilizando únicamente elementos de hardware digital, por lo que es necesario activar la opción de HW conexión al momento de realizar la configuración. Al hacer esto, será posible hacer la conexión del pin físico de la tarjeta con los elementos de hardware interno de ésta.

Una vez seleccionado los pines, se procede a hacer el “mapeo” es decir, elegir el pin físico a utilizar en la tarjeta. Esto se logra mediante el cuadro de diálogo de la Figura 8.

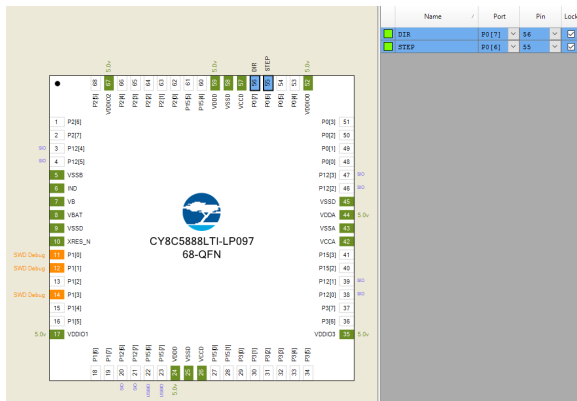


Figura 8. Mapeo de salidas digitales.

Posteriormente se genera el código estructurado para el control de la secuencia del sistema mediante software. Un código de prueba utilizado se muestra en la Figura 9, donde se crea un bucle infinito que permite mover el motor de un sentido a otro en intervalos de 3 segundos.

```
#include "project.h"

int steps = 200;

int i;
int main(void)
{
    CyGlobalIntEnable;
    for(;;)
    {
        DIR_Write(1);
        for(i=0; i<steps;i++)
        {
            STEP_Write(1);
            CyDelay(5);
            STEP_Write(0);
            CyDelay(5);
        }

        CyDelay(3000);

        DIR_Write(1);
        for(i=0; i<steps;i++)
        {
            STEP_Write(1);
            CyDelay(5);
            STEP_Write(0);
            CyDelay(5);
        }

        CyDelay(3000);
    }
}
```

Figura 9. Código de prueba para secuencias de control por software

En este caso, el control se realiza aún por software debido a que solo se requería conocer el comportamiento de los motores a pasos al utilizar esta plataforma.

Las pruebas se realizaron utilizando un motor a pasos NEMA17 y tomando en cuenta que el giro completo de este motor toma 200 pesos.

*B. Pruebas de movimiento por hardware mediante timers.*

Esta prueba fue realizada ya utilizando los elementos de electrónica digital de la tarjeta, tratando de evitar el uso de elementos de programación estructurada. Se utilizaron contadores de 16 bits debido a que se tiene una gran cantidad de cuentas disponibles para los divisores de frecuencia de reloj para los diferentes niveles de velocidad.

Para la generación de los pulsos se tiene un reloj general de 100Hz conectado a la entrada del contador. Este elemento no tiene una salida lógica por lo que no puede conectarse directamente al pin físico de la tarjeta por lo que se recurre a un comparador que detecta cuando se han hecho los pulsos de reloj necesarios y los transforma en niveles lógicos a través de un arreglo de compuertas digitales. En la Figura 10 se muestra el diagrama digital utilizado para esta prueba.

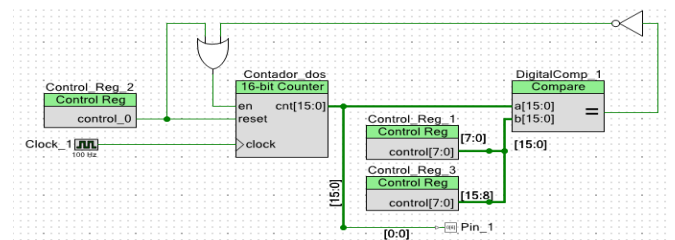


Figura 10. Configuración de control utilizando timers.

El número de pasos que quieren darse se realiza mediante el Registro de control 2 y los registros 1 y 3 se utilizan para identificar si el conteo ha llegado a su fin y desactivar el contador.

Una vez compilado el programa se analizó el uso de recursos teniendo los resultados que se muestran en la Figura 11.

Digital	
DMA Channels	0 / 24
Timer	0 / 4
UDB	29.2 %
Macrocells	48 / 192
Unique P-...	112 / 384
Datapath Cells	3 / 24
Status Cells	3 / 24
Control Cells	4 / 24

Figura 11. Resultados de uso de recursos de prueba con timer.

Se puede observar que se utilizaron cerca del 30% de los recursos de hardware digital para el control de un solo motor a pasos. Es decir, mediante este método, y utilizando la tarjeta PSOC 5, solamente se podrían controlar un máximo de 3 motores a pasos dejando poco espacio en el procesador digital para el uso de otros tipos de bloques en el programa.

Con esto, se optó por la implementación de otro método con la finalidad de reducir el número de recursos digitales requeridos para el control de motores.

*C. Prueba de movimiento por hardware mediante contadores e interrupciones.*

La segunda opción por hardware valorada combina el uso de contadores con interrupciones del mismo, lo que permite tener un control del conteo más preciso al detener el funcionamiento del sistema si es que los contadores se han desbordado. El diagrama de flujo de la prueba se presenta en la Figura 12.

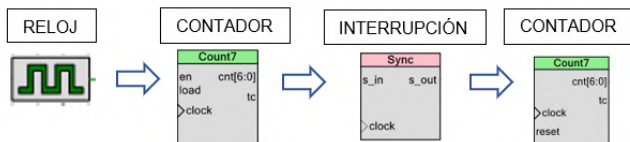


Figura 12. Diagrama de flujo de la prueba con contadores e interrupciones.

Se hizo uso de contadores descendentes de 7 bits en cascada, donde al igual que en la prueba pasada, de inicio se tiene un reloj de 100 Hz el cual marca la velocidad del motor, este llega a la entrada de reloj del contador utilizado; el contador al llegar al final del conteo establecido, genera una señal de tiempo completado el cual va dirigido a un segundo contador en cascada, esto con el fin de tratar de lograr una lógica de reset óptima que en la prueba pasada no se logró; ahora bien, la señal generada de tiempo completada es asíncrona, por lo que, para llevar dicha señal hacia el segundo contador se tenía que sincronizar, para lo cual se hace uso de un bloque sincronizador, el cuál cumple perfectamente con la

función buscada. El diagrama lógico utilizado se muestra en la Figura 13.

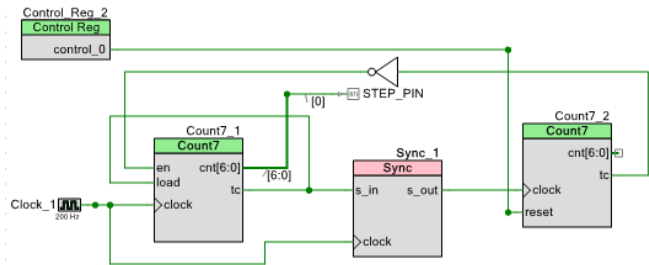


Figura 13. Diagrama lógico para control por contador e interrupción.

Una vez terminado el conteo del segundo contador, este logra realizar el reinicio del primero, mientras que el reset del segundo contador se realiza con un registro de control, sin embargo, al realizar pruebas prolongadas, se encontró que al final de los ciclos establecidos, se mostraba un conteo extra al deseado en el primer contador.

Los resultados en cuanto a consumo de recursos se muestran en la Figura 14. Se debe tomar en cuenta que el bloque anterior se replicó 6 veces, por lo que es posible controlar 6 motores de forma simultánea y aún tener espacio en memoria.

Digital	
DMA Channels	0 / 24
Timer	0 / 4
UDB	58.3 %
Macrocells	30 / 192
Unique P-terms	50 / 384
Datapath Cells	3 / 24
Status Cells	11 / 24
Control Cells	14 / 24

Figura 14. Recursos utilizados con contadores e interrupción.

Si bien se disminuyó considerablemente el número de bloques utilizados en comparación con la opción anterior, se tienen cuentas que pueden ser inestables por lo que no es recomendable en aplicaciones en donde se la precisión es un requisito principal. Sin embargo, es funcional en aplicaciones que solamente requieren movimiento del motor.

*D. Prueba de movimiento por hardware mediante timer y contador*

Derivado de los resultados de las pruebas anteriores se diseñó una opción combinando el uso de temporizadores con los contadores. Es decir, se combinaron los métodos pasados, el diagrama de flujo de esta prueba se muestra en la Figura 15.

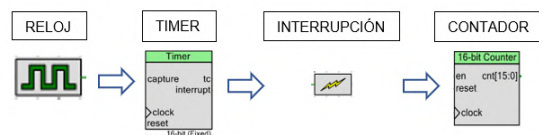


Figura 15. Diagrama de flujo de la prueba combinada.

Para esta prueba se hizo uso de un temporizador al que a su entrada de reloj tiene 100Hz al igual que las pruebas anteriores, en esta prueba no se recurre al uso del tiempo completado del timer, sin embargo, ahora se hace uso de una interrupción que se estará llamando desde programación en el Main.c (programa estructurado), subsecuente a esto se encuentra un contador de 16 bits ascendente, donde la interrupción proveniente del timer, seguido de un inversor, llegan al pin de habilitación del contador.

A la salida del contador se encuentran dos registros de estatus con la finalidad de guardar los valores de las cuentas de los pulsos, sin embargo, una limitante de los registros de estatus, es que estos son de 8 bits, por lo que es necesario utilizar dos, uno para el registro alto y otro para el registro bajo.

El funcionamiento del diagrama de la Figura 16 se resume en que, el timer comienza su funcionamiento, posteriormente al llegar al fin del conteo establecido, activa una interrupción con el fin de hacer el reinicio del sistema.

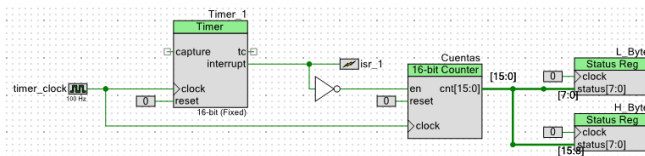


Figura 16. Diagrama lógico para control combinado.

Con este método se tiene un control más preciso de la señal de pulsos del controlador. Sin embargo, se ocupan mayor cantidad de recursos como lo muestran las estadísticas de la Figura 17. En esta configuración, solo es posible un máximo de 4 motores, pero con un movimiento más preciso y controlado.

Digital	
DMA Channels	0 / 24
Timer	2 / 4
UDB	21.9 %
Macrocells	42 / 192
Unique P-terms	61 / 384
Datapath Cells	3 / 24
Status Cells	5 / 24
Control Cells	1 / 24

Figura 17. Consumo de recursos en prueba combinada.

### III. RESULTADOS

Después de las pruebas realizadas con las diferentes configuraciones lógicas se llegó a una versión actualizada donde se retoma la idea de los contadores de 12 bits para realizar un número mayor de cuentas, pero con un arreglo de compuertas diferente.

Se mantiene un contador seguido de un comparador, el cual funge entre el contador y valores proporcionados por medio de registros de control, pero a diferencia de las propuestas anteriores, el pin de STEP en lugar de estar conectado al bit 0 del contador ahora este se conecta al reloj. Sin embargo, se hizo el diseño de una lógica por medio de una compuerta NOR y un inversor, con el fin de desactivar el reloj,

que el contador, ahora es desactivado manualmente desde un registro de control.

Por lo que, con dicho registro de control, se logra el reset del contador, y este al entrar en esa condición, incumple la condición realizada con las compuertas y el reloj vuelve a activarse obteniendo una señal más estable. El diagrama propuesto se muestra en la Figura.

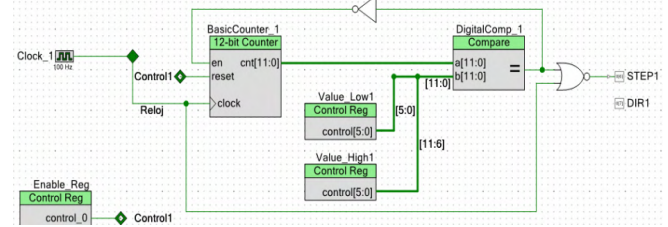


Figura 18. Diagrama de control resultante de las pruebas.

En la parte final de la programación se añade un registro de estatus, el cual permite conocer cuando se termina de contar, al igual se añaden las interrupciones con la finalidad de tener una opción en caso de que se requiera el uso de interruptores de final de carrera para generar un sistema a lazo abierto que pueda ampliar la aplicación del diagrama propuesto en diferentes aplicaciones.

De igual manera, se realizó una prueba con esta configuración para el control de 3 motores a pasos de forma simultánea con la finalidad de obtener una retroalimentación del funcionamiento en más de un motor a la vez, el diagrama para el control de 3 motores se ve en la Figura 19. Sin embargo, se utilizan pocos recursos por lo que es posible ampliarse para el control de hasta 6 motores bipolares, incrementando el campo de aplicación.

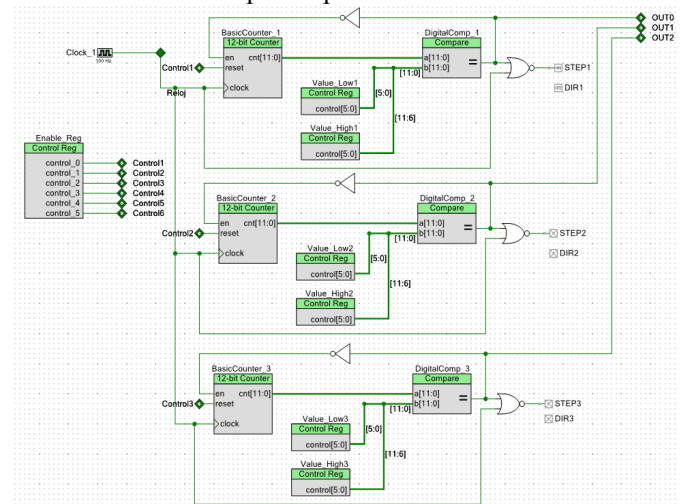


Figura 19. Diagrama de control de tres motores a pasos de forma simultánea.

Terminado el diagrama lógico se hizo el mapeo de entradas y salidas digitales para validar si el microcontrolador es capaz de manejar todos los elementos de hardware, es decir, controlar 6 motores a pasos con los interruptores de final de

carrera correspondiente. El mapeo realizado se muestra en la Figura 20.

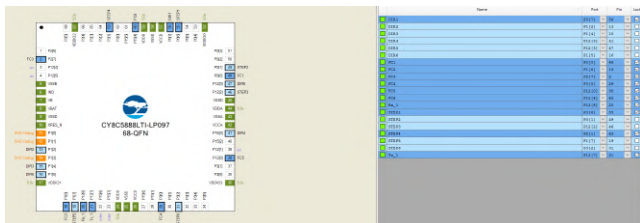


Figura 20. Mapeo de pines físicos.

El uso de recursos del diagrama final propuesto permite el manejo de 6 motores a pasos, los finales de carrera de todos, así como el uso de comunicación UART teniendo los resultados de consumo que se muestra en la Figura 21.

Digital	
DMA Channels	0 / 24
Timer	0 / 4
UDB	94.3 %
Macrocells	142 / 192
Unique P...	362 / 384
Datapath Cells	4 / 24
Status Cells	5 / 24
Control Cells	15 / 24

Figura 21. Recursos consumidos por el diagrama lógico propuesto.

Si bien el diagrama lógico es en donde se realiza la mayoría del control del driver, también es necesario algunos elementos de programación estructurada para la activación, manejo y control de las interrupciones. El código estructurado se encuentra en la Figura 22.

```
#include "project.h"
#include "stdio.h"
#include "ControlMotor.h"

CY_ISR_PROTO(FC1_Interrupt);
CY_ISR_PROTO(FC3_Interrupt);

int flag=0;

int main(void)
{
    CyDelay(1000);
    FC1_ISR_StartEx(FC1_Interrupt);
    FC3_ISR_StartEx(FC3_Interrupt);
    CyGlobalIntEnable;

    UART_1_Start();
    MotorSteps(1, 400, 0);

    for(;;)
    {
        if(flag==1)
        {
            MotorSteps(1, 400, 1);
            flag=0;
        }
    }

    CY_ISR (FC1_Interrupt)
    {
        StopMotor(1);
        STEP4_Write(1);
    }

    CY_ISR (FC3_Interrupt)
    {
        flag=1;
    }
}
```

Figura 22. Código estructurado para el control de motores a pasos por hardware.

#### IV. DISCUSIÓN, CONCLUSIÓN Y RECOMENDACIONES

Los motores a pasos son los actuadores más utilizados en sistemas de control de movimiento en donde la precisión es de vital importancia. Es por ello, que existen una gran cantidad de ejemplos para su control. Sin embargo, con la llegada de nuevos dispositivos con mejores prestaciones como lo son los sistemas PSOC es necesario buscar alternativas de programación con el fin de tener una visión diferente del uso de estos microcontroladores.

Las opciones de diseño propuestas, al ser generadas con lógica combinatorial, pueden implementarse incluso con elementos TTL que se utilizan de forma didáctica en la enseñanza de la electrónica digital, lo que puede abrir un abanico de posibilidades para el uso de la plataforma PSOC como instrumento para la generación de material didáctico.

Se recomienda, para futuros proyectos derivados de este, mejorar el consumo de recursos, pues si bien es posible el manejo de múltiples actuadores y elementos de comunicación, se deja poco espacio en memoria para el uso de otros bloques de programación en caso de ser necesarios.

#### V. AGRADECIMIENTOS

Al Instituto Tecnológico Superior de Lerdo, especialmente a las divisiones de Ingeniería en Sistemas Automotrices e Ingeniería en electrónica, así como el departamento de investigación y desarrollo tecnológico por las facilidades brindadas para la realización de este proyecto.

#### VI. REFERENCIAS

- [1] H. R. Domínguez, «Control de coordenadas en XY de un mecanismo de motores a pasos y un microcontrolador,» Jóvenes en la Ciencia, vol. 3, n° 2, pp. 2273-2277, 2017.
- [2] L. Corona, Sensores y actuadores. Aplicaciones con Arduino, México: Patria, 2014.
- [3] S. Jennings, «Motores a pasos,» Informador Técnico, pp. 47-58, 2002.
- [4] J. M. V. Díaz, Microcontroladores PIC. Principios y aplicaciones, México: ASEUC, 2009.
- [5] M. Rashid, Electrónica de Potencia, México: Pearson Educación, 2004.
- [6] M. V. O. Machay, Diseño y construcción de una máquina cnc router de tres grados de libertad utilizando arduino uno y el driver drv8825., Quito: ISTVN, 2020.
- [7] R. Ashby, Designer's guide to the Cypress PSoC, Newnes, 2005.
- [8] A. Kitagawa, «A Study of Programmable System on Chip (PSoC),» Journal of Physics: Conference Serie, vol. 1899, n° 1, 2021.

## VII. BIOGRAFÍA



**Machado Díaz Eduardo.** Nació en la ciudad de Gómez Palacio, Dgo. México el 29 de enero de 1994. Obtuvo el grado de Ingeniero en Electrónica con especialidad en Mecatrónica y automatización en el año de 2017 en el Instituto Tecnológico Superior de Lerdo. Egresado de la maestría en Ingeniería Mecatrónica en el año de 2018 del mismo instituto en donde participó en proyectos de investigación enfocados en el control automático.

Actualmente es docente en el Instituto Tecnológico Superior de Lerdo como docente en la división de Sistemas Automotrices en las áreas de electrónica, automatización industrial y sistemas robóticos. Sus líneas de investigación son los sistemas de control automático, la instrumentación y el diseño de interfaces.



**Coto Fuentes Hesner.** Ingeniero en Automática egresado del Instituto Superior Politécnico José Antonio Echeverría, La Habana, Cuba. Maestro en Diseño de Sistemas Electrónicos por el Centro de Investigaciones en Microelectrónica, La Habana, Cuba, y Doctor en Ciencias en Ingeniería Eléctrica con especialización en Instrumentación Electrónica por el Instituto Tecnológico de la Laguna, Torreón, Coahuila, México.

En su trayectoria como docente - investigador se ha especializado en las áreas de Instrumentación Virtual y desarrollo de sistemas electrónicos aplicados a la medicina, la industria y el medio ambiente