



UNIVERSIDAD DEL BÍO-BÍO

FACULTAD DE INGENIERÍA

DEPTO. INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

“Desarrollo de una plataforma de docencia para control de motores DC y de pasos basada en un microcontrolador C2000”

Mauricio Javier Véjar Henríquez

Iván José Vivanco Escobar

SEMINARIO PARA OPTAR AL TÍTULO DE
INGENIERO DE EJECUCIÓN EN ELECTRÓNICA

CONCEPCIÓN – CHILE

2017



UNIVERSIDAD DEL BÍO-BÍO

FACULTAD DE INGENIERÍA

DEPTO. INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

“Desarrollo de una plataforma de docencia para control de motores DC y de pasos basada en un microcontrolador C2000”

Mauricio Javier Véjar Henríquez.

Iván José Vivanco Escobar.

Eugenio Wernekinck Armstrong.

Pedro Melín Coloma.

John Correa Toloza.

AGRADECIMIENTOS

Sin duda ha sido un camino largo y difícil, el cual está llegando a su fin, nos costó bastante llegar hasta donde estamos y de seguro no lo habríamos logrado de no ser por la ayuda de varias personas que jugaron un rol fundamental en esta travesía, que nos brindaron apoyo moral, el cual para nosotros fue muy importante en el momento que pensamos en bajar los brazos, gracias por las palabras de aliento y por el ánimo que nos dieron para seguir adelante.

Gracias además, a todas las personas que nos cooperaron desinteresadamente para lograr terminar nuestro seminario, gracias a los profesores que nos orientaron, a nuestros amigos que aportaron con algunas ideas, y en especial, a Don Pedro Inostroza por toda su disposición, amabilidad y ayuda desinteresada.

RESUMEN

El trabajo desarrollado se basa en un kit de control de motores, DRV8412EVM, desarrollado por Texas Instruments (TI), el cual permite experimentar el control de movimiento de tres motores; dos de corriente directa (DC) y un motor de pasos (STEPPER). El control de estos no es simultáneo, por lo que se puede controlar un tipo de motor a la vez.

El trabajo se presenta dividido en tres partes.

La primera se encuentra focalizada al desarrollo de una guía de funcionamiento del software de control, la cual muestra en detalle los pasos a seguir para su puesta en marcha.

La segunda, enfocada al desarrollo de una guía para las experiencias de control propuestas para ambos tipos de motores. Las cuales están debidamente detalladas.

Y la tercera, en la cual se detalla el diseño y construcción de la maqueta, el cual contempla la seguridad del kit, una fácil conexión, y la aplicación de las experiencias propuestas.

ÍNDICE

AGRADECIMIENTOS.....	3
RESUMEN.....	4
ÍNDICE	5
ÍNDICE DE FIGURAS.....	7
ÍNDICE DE TABLAS	8
INTRODUCCIÓN.....	9
OBJETIVOS	10
1. Objetivo General	10
2. Objetivos específicos.....	10
MARCO TEÓRICO	11
Motores DC.	11
Motor de Pasos.	13
Características comunes de los motores de pasos.	14
Tipos de motores de pasos.	14
Puente H.....	15
CAPÍTULO 1: DESCRIPCIÓN DEL HARDWARE	17
Hardware del Kit.....	17
Driver Integrado DRV8412	21
Modos de Funcionamiento del Circuito.....	22
Tarjeta de Control TMS320F28035.	24
Módulo PWM	25
Módulo eQEP	26
Convertidor Analógico-Digital	26
CAPÍTULO 2: DESCRIPCIÓN DEL SOFTWARE PARA PROGRAMAR	29
Code Composer Studio (CCS)	29
ControlSUITE	30
Interfaz de prueba rápida (DRV8412GUIv5)	31
CAPITULO 3: PROGRAMACIÓN DE PERFILES DE TRABAJO.	32
Bloques macro.....	32
Niveles de compilación incremental de cada experiencia.	35
CAPITULO 4: RESULTADOS Y DISCUSIÓN.	37

Trabajo con motor de pasos.	41
Trabajo con motor DC.	48
DISCUSIÓN.....	52
CONCLUSIONES.	53
Sumario y Metodología.	53
Conclusión.	53
Sugerencias a trabajos futuros.....	54
REFERENCIAS.	55
ANEXO A: MANUAL DE PUESTA EN MARCHA DEL DRV8412-C2-KIT	57
Instrucciones de Configuración de Hardware	57
Configuración de software para los proyectos de DRV8412-EVM.....	60
Instalando Code Composer y controlSuite.....	60
Configurar Code Composer Studio para trabajar con la DRV8412-EVM	60
➤ Problema Frecuente.....	64
Configurando un Proyecto	65
Compilar y Cargar el Proyecto.....	66
➤ Problema frecuente.	66
Configurar la Ventana Watch.	67
Ejecutar el Código.	68
ANEXO B: GUIA DE LABORATORIO.....	69
EXPERIENCIAS DE CONTROL.....	69
Experiencia N°1: Simulación de planta con motor de pasos.....	69
Experiencia N°2: Simulación de planta con motor DC	72
ANEXO C: CÓDIGO SECUENCIA DE ASCENSO Y DESCENSO	75
ANEXO D: CODIGO SECUENCIA CINTA TRANSPORTADORA.....	76
ANEXO E: DATOS TÉCNICOS (DRV8412).....	77
ANEXO F: ESQUEMA DISEÑO GABINETE	80
ANEXO G: ESQUEMA DISEÑO MAQUETA ELEVADOR	81
ANEXO H: ESQUEMA DISEÑO CINTA TRANSPORTADORA	82

ÍNDICE DE FIGURAS

Figura 1: “Estructura más popular Motor DC”	11
Figura 2: “Esquema funcionamiento Motor DC”	12
Figura 3: “Diagrama estructural del motor de pasos”	13
Figura 4: “Esquema circuito puente completo”	16
Figura 5: “Los dos estados básicos del circuito”	16
Figura 6: “DRV8412-C2-KIT”	17
Figura 7: “Vista isométrica tarjeta de control”	18
Figura 8: “Tarjeta de potencia”	20
Figura 9: “Diagrama simplificado de una aplicación del driver DRV8412”	22
Figura 10: “Conexión de los motores DC a los puentes integrados”	23
Figura 11: “Tarjeta de control TMS320F28035 (vista anterior)”	24
Figura 12: “Tarjeta de control TMS320F28035 (vista posterior)”	24
Figura 13: “Esquema de la F28035 y sus periféricos”	25
Figura 14: “Detalle del ciclo de trabajo en la onda PWM”	26
Figura 15: “Diagrama de bloques del módulo ADC”	27
Figura 16: “Valor digital convertido en el ADC”	27
Figura 17: “Diagrama de flujo control de Motor DC”	33
Figura 18: “Diagrama de flujo control de motor de pasos”	34
Figura 19: “Vista en perspectiva diseño gabinete”	37
Figura 20: “Diagrama conexionado eléctrico del gabinete”	38
Figura 21: “Vista en perspectiva diseño elevador”	39
Figura 22: “Vista en perspectiva diseño cinta transportadora”	40
Figura 23: “Esquema representación alturas torre y caja”	41
Figura 24: “Datos técnicos obtenidos de la interfaz de prueba rápida”	42
Figura 25: “Gráfico desplazamiento en el tiempo del elevador”	44
Figura 26: “Tensiones de salida puentes H y Corrientes de cada fase del motor de pasos al trabajar en Medio Paso, 40% velocidad”	45
Figura 27: “Tensiones de salida puentes H y Corrientes de cada fase del motor de pasos al trabajar en Cuarto de Paso, 40% velocidad”	45
Figura 28: “Tensiones de salida puentes H y Corrientes de cada fase del motor de pasos al trabajar en 16 Micropasos, 40% velocidad”	46

Figura 29: “Tensiones de salida puentes H y Corrientes de cada fase del motor de pasos al trabajar en Medio Paso, 40% velocidad, sentido giro horario”	46
Figura 30: “Tensiones de salida puentes H y Corrientes de cada fase del motor de pasos al trabajar en Medio Paso, 40% velocidad, sentido giro antihorario”	47
Figura 31: “Tensiones de salida puentes H y Corrientes de cada fase del motor de pasos al trabajar en 8 Micropasos, 5% velocidad”	47
Figura 32: “Señal de corriente de realimentación obtenida con secuencia sugerida en el motor DC 2”	48
Figura 33: “Tensión y corriente medida en el motor al ejecutar secuencia sugerida en el motor DC 2”	49
Figura 34: “Tensiones y corriente del motor DC 2 al aplicar una tensión de control constante”	50
Figura 35: “Señales de tensiones y corriente al modificar la carga del motor DC 2”	51
Figura 36: “Configuración de Switch tarjeta de control”	57
Figura 37: “Configuración de Hardware para experiencia 2xDC_Motor”	58
Figura 38: “Configuración de Hardware para experiencia Stepper”	59
Figura 39: “Workspace Launcher”	60
Figura 40: “Creando una configuración de tarjeta”	61
Figura 41: “Configurando una nueva tarjeta”	62
Figura 42: “Agregando todos los proyectos de DRV8412-EVM al workspace”	63
Figura 43: “Pestaña avanzada de TI XDS100”	64
Figura 44: “Elegiendo la configuración F2803x_RAM”	65
Figura 45: “Configurando la ventana Watch”	67
Figura 46: “Variables de la ventana Watch”	70
Figura 47: “Indicación para editar programa”	71
Figura 48: “Variables de la ventana Watch (Brushed DC)”	73
Figura 49: “Indicación para editar programa”	74

ÍNDICE DE TABLAS

Tabla 1: “Módulos usados en el sistema incremental de compilación (Brushed DC) [7]”	35
Tabla 2: “Módulos usados en el sistema incremental de compilación (Stepper) [8]”	36
Tabla 3: “Datos obtenidos de la interfaz de prueba rápida”	43

INTRODUCCIÓN

Las técnicas de control de motores DC son herramientas que se utilizan para controlar la velocidad, el torque y el suministro de potencia de los motores de corriente continua.

La naturaleza de la carga que se acopla al eje del motor define el comportamiento de esta máquina. Para el caso de una carga liviana, el motor desarrollara una velocidad relativamente alta y un torque de giro bajo. Por el contrario, si se dispone de una carga pesada o difícil de mover, el motor se moverá a una velocidad menor y entregara más torque, pues una mayor carga lo exige.

Para poder observar de mejor forma el comportamiento de estos motores al variar la carga, propusimos una maqueta didáctica, la cual favorece la interacción entre el alumno y el kit de forma sencilla y facilitara el aprendizaje de forma práctica, además de ser un método mucho más llamativo.

La base de este trabajo surge a partir de la necesidad de observar de forma practica el comportamiento de los motores en algunas asignaturas, como por ejemplo, control de movimiento; además nuestro trabajo lo impulsa el hecho de existir en el departamento varios kit de control de motores, los cuales se adquirieron con el motivo de implementarlos en dichas asignaturas, pero que, hasta ahora no existía una propuesta real de uso.

Cabe destacar que el control de la tarjeta del kit es realizado por una DSP (Procesador Digital de Señales) el cual mediante ondas PWM (Modulación por Ancho de Pulso) regula la velocidad de los motores.

OBJETIVOS

1. Objetivo General

Desarrollar una maqueta que permita experimentar en el control de movimientos punto a punto, con motores DC y de pasos con distintos perfiles, utilizando un kit de control de motores (microcontroladores C2000 de Texas Instruments) disponible en el departamento.

2. Objetivos específicos

- Diseñar y construir una maqueta de control para motor DC y motor de pasos basada en el microcontrolador C2000.
- Desarrollar un documento base de funcionamiento de la maqueta y del software de control.
- Desarrollar una experiencia de control para el motor DC y una para el de pasos.

MARCO TEÓRICO

Motores DC.

Un motor de corriente directa, o motor DC, convierte la energía eléctrica en energía mecánica. La conversión de energía en un motor eléctrico se debe a la interacción entre una corriente eléctrica y un campo magnético. Un campo magnético, que se forma entre los dos polos opuestos de un imán, es una región donde se ejerce una fuerza sobre determinados metales o sobre otros campos magnéticos. Un motor eléctrico aprovecha este tipo de fuerza para hacer girar un eje, transformándose así la energía eléctrica en movimiento mecánico.

Los dos componentes básicos de todo motor eléctrico son el rotor y el estator (Figura 1).

El rotor es una pieza giratoria, un electroimán móvil, con varios salientes laterales, que llevan cada uno a su alrededor un bobinado por el que pasa la corriente eléctrica.

El estator, situado alrededor del rotor, formado por un imán o un electroimán fijo, cubierto con un aislante. Al igual que el rotor, dispone de una serie de salientes con bobinados eléctricos por los que circula la corriente.

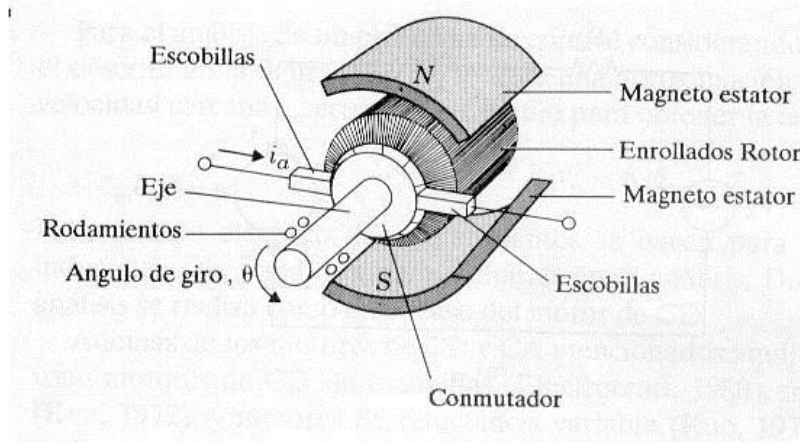


Figura 1: “Estructura más popular Motor DC”

Fuente: Apuntes clase Control Electrónico de Motores Fraccionarios

Cuando se introduce una espira de hilo de cobre en un campo magnético y se conecta a una batería, la corriente pasa en un sentido por uno de sus lados y en sentido contrario por el lado opuesto. Así, sobre los dos lados de la espira se ejerce una fuerza, en uno de estos hacia arriba y en el otro hacia abajo (observar Figura 2). Si la espira de hilo va montada sobre el eje metálico, empieza a dar vueltas hasta alcanzar la posición vertical. Entonces, en

esta posición, cada uno de los hilos se encuentra situado en el medio entre los dos polos, y la espira queda retenida. Para que la espira siga girando después de alcanzar la posición vertical, es necesario invertir el sentido de circulación de la corriente. Para conseguirlo, se emplea un conmutador o colector, que en el motor eléctrico más simple, el motor de corriente continua, está formado por dos chapas de metal con forma de media luna, que se sitúan sin tocarse, como las dos mitades de un anillo, y que se denominan delgas.

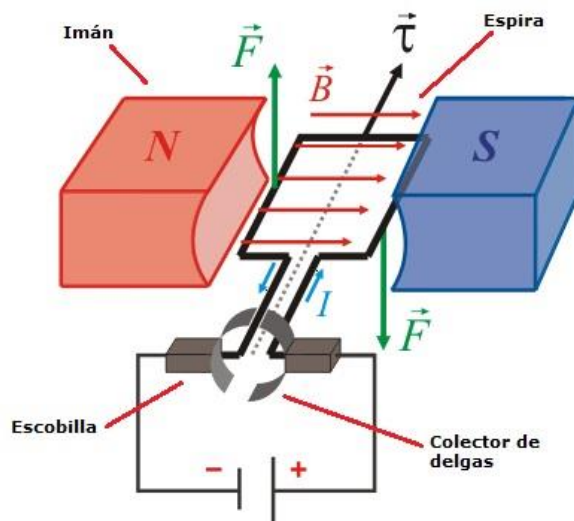


Figura 2: “Esquema funcionamiento Motor DC”
Fuente: www.automatismoindustrial.com

Los dos extremos de la espira se conectan a las dos medias lunas. Dos conexiones fijas, unidas al bastidor del motor y llamadas escobillas, hacen contacto con cada una de las delgas del colector, de forma que, al girar la armadura, las escobillas contactan primero con una delga y después con la otra. Cuando la corriente eléctrica pasa por el circuito, la armadura empieza a girar y la rotación dura hasta que la espira alcanza la posición vertical. Al girar las delgas del colector con la espira, cada media vuelta se invierte el sentido de circulación de la corriente eléctrica. Esto quiere decir que la parte de la espira que hasta ese momento recibía la fuerza hacia arriba, ahora la recibe hacia abajo, y la otra parte al contrario. De esta manera la espira realiza otra media vuelta y el proceso se repite mientras gira la armadura. El esquema descrito corresponde a un motor de corriente continua, el más simple dentro de los motores eléctricos, pero que reúne los principios fundamentales de este tipo de motores [11].

Motor de Pasos.

Los motores de pasos tienen un comportamiento del todo diferente al de los motores de corriente continua. En primer lugar, no giran libremente por sí mismos. Los motores de pasos, avanzan girando por pequeños pasos. También difieren de los motores de CC en la relación entre velocidad y torque (un parámetro que también es llamado "par motor" y "par de giro"). Los motores de CC no son buenos para ofrecer un buen torque a baja velocidad sin la ayuda de un mecanismo de reducción. Los motores de pasos, en cambio, trabajan de manera opuesta: su mayor capacidad de torque se produce a baja velocidad.

Los motores de pasos tienen una característica adicional: el torque de detención (que se puede ver mencionado también como "par de detención", e incluso par/torque "de mantenimiento"), que no existe en los motores de CC, hace que un motor de pasos se mantenga firmemente en su posición cuando no está girando. Esta característica es muy útil cuando el motor deja de moverse y, mientras está detenido, la fuerza de carga permanece aplicada a su eje. Se elimina así la necesidad de un mecanismo de freno.

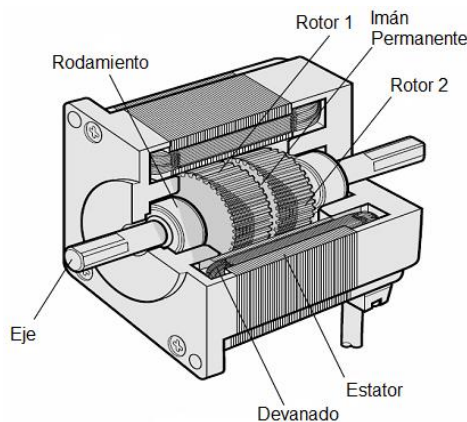


Figura 3: "Diagrama estructural del motor de pasos"
Fuente: [12]

Si bien es cierto que los motores de pasos funcionan controlados por un pulso de avance, el control de un motor de pasos no se realiza aplicando en directo este pulso eléctrico que lo hace avanzar. Estos motores tienen varios bobinados que, para producir el avance de ese paso, deben ser alimentados en una adecuada secuencia. Si se invierte el orden de esta secuencia, se logra que el motor gire en sentido opuesto. Si los pulsos de alimentación no se proveen en el orden correcto, el motor no se moverá apropiadamente. Puede ser que zumbe y no se mueva, o puede ser que gire, pero de una manera tosca e irregular.

Esto significa que hacer girar un motor de pasos no es tan simple como hacerlo con un motor de corriente continua, al que se le entrega una corriente y listo. Se requiere un circuito de

control, que será el responsable de convertir las señales de avance de un paso y sentido de giro en la necesaria secuencia que energice los bobinados.

Características comunes de los motores de pasos.

Un motor de pasos se define por estos parámetros básicos:

Voltaje

Los motores de pasos tienen una tensión eléctrica de trabajo. Este valor viene impreso en su carcasa o por lo menos se especifica en su hoja de datos. Algunas veces puede ser necesario aplicar un voltaje superior para lograr que un determinado motor cumpla con el torque deseado, pero esto producirá un calentamiento excesivo y/o acortará la vida útil del motor.

Resistencia eléctrica:

Otra característica de un motor de pasos es la resistencia de los bobinados. Esta resistencia determinará la corriente del motor, y su valor afecta la curva de torque del motor y su velocidad máxima de operación.

Grados por paso y pasos por revolución:

Este factor define la cantidad de grados que rotará el eje para cada paso completo. Una operación de medio-paso o semi-paso (half step) del motor duplicará la cantidad de pasos por revolución al reducir la cantidad de grados por paso. Cuando el valor de grados por paso no está indicado en el motor, es posible contar a mano la cantidad de pasos por vuelta, haciendo girar el motor y sintiendo por el tacto cada "diente" magnético. Los grados por paso se calculan dividiendo 360 (una vuelta completa) por la cantidad de pasos que se contaron. Las cantidades más comunes de grados por paso son: $0,72^\circ$, $1,8^\circ$, $3,6^\circ$, $7,5^\circ$, 15° y hasta 90° . A este valor de grados por paso usualmente se le llama la resolución del motor. En el caso de que un motor no indique los grados por paso en su carcasa, pero sí la cantidad de pasos por revolución, al dividir 360 por ese valor se obtiene la cantidad de grados por paso. Un motor de 200 pasos por vuelta, por ejemplo, tendrá una resolución de $1,8^\circ$ por paso [13].

Tipos de motores de pasos.

Los motores de pasos se dividen en dos categorías principales: de imán permanente y de reluctancia variable. También existe una combinación de ambos, a los que se les llama híbridos.

Los de **imán permanente** son los que poseen un imán que aporta el campo magnético para la operación.

Los motores del tipo de **reluctancia variable**, en cambio, poseen un rotor de hierro dulce que en condiciones de excitación del estator, y bajo la acción de su campo magnético, ofrece menor resistencia a ser atravesado por su flujo en la posición de equilibrio. Su mecanización es similar a los de imán permanente y su principal inconveniente radica en que en condiciones de reposo (sin excitación) el rotor queda en libertad de girar y, por lo tanto, su posicionamiento de régimen de carga dependerá de su inercia y no será posible predecir el punto exacto de reposo. El tipo de motor de reluctancia variable consiste en un rotor y un estator cada uno con un número diferente de dientes. Ya que el rotor no dispone de un imán permanente, gira libremente si no tiene corriente alimentándolo, o sea que no tiene torque de detención.

Los motores **híbridos** combinan las mejores características de los de reluctancia variable y de imán permanente. Se construyen con estatores multi-dentados y un rotor de imán permanente. Los motores híbridos estándar tienen 200 dientes en el rotor y giran en pasos de 1,8 grados. Existen motores híbridos con configuraciones de 0,9° y 3,6°. Dado que poseen alto torque estático y dinámico y se mueven a muy altas velocidades de pulso, se los utiliza en una amplia variedad de aplicaciones industriales [13].

Puente H.

Un puente H es un circuito electrónico del tipo inversor, que permite controlar el sentido de giro de un motor de corriente directa o excitar las fases de los motores de paso. Está compuesto por cuatro interruptores, dispuestos de dos en cada pierna. La tensión máxima que se puede obtener a la salida es aproximadamente igual a la tensión que se tiene a la entrada del puente (ya que existe una pequeña caída de tensión en los switch).

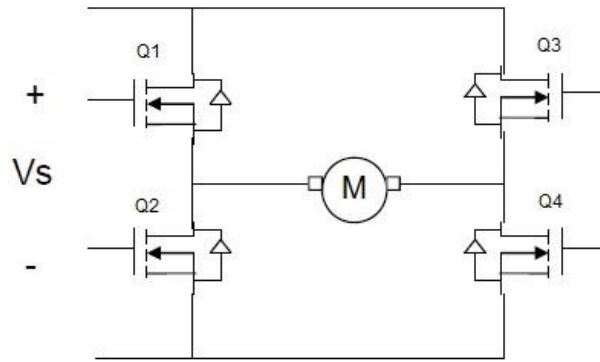


Figura 4: “Esquema circuito puente completo”
Fuente: Elaboración propia

La activación de los interruptores se hace por la compuerta de los MOSFET. Los diodos que los acompañan permiten a la corriente circular en sentido inverso cada vez que se conmute la tensión, puesto que el motor está compuesto por bobinas que luego de la conmutacion se opondrán a que la corriente varíe.

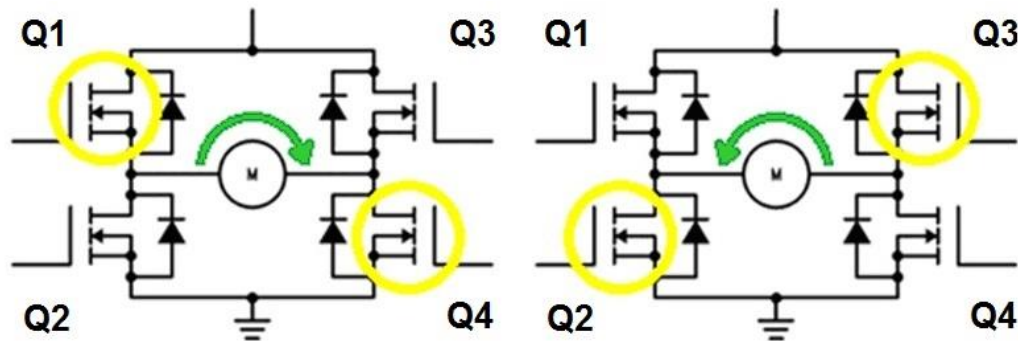


Figura 5: “Los dos estados básicos del circuito”
Fuente: Elaboración propia

Cuando los interruptores Q1 y Q4 (ver Figura 5) están cerrados (Q2 y Q3 abiertos) se aplica una tensión positiva en el motor, haciéndolo girar en un sentido. Abriendo los interruptores Q1 y Q4 (cerrando Q2 y Q3), la tensión se invierte, permitiendo el giro en sentido inverso del motor [14].

CAPÍTULO 1: DESCRIPCIÓN DEL HARDWARE

En este capítulo se describen los componentes del kit (DRV8412-C2-KIT) utilizado, desarrollado por Texas Instruments. El kit se separa en dos placas, la primera tiene el DSP con la que se realiza la programación y control desde el PC, y la segunda tiene el circuito de potencia con los puentes H donde son conectados los respectivos motores, posteriormente se hará una descripción de ambas. Se hace énfasis en el driver integrado de la placa de potencia y en los periféricos de la tarjeta de control. La explicación de funcionamiento y de la estructura interna del microcontrolador C2000 no es parte de este trabajo.

Hardware del Kit

DRV8412-C2-KIT es un kit de evaluación para el control de motores desarrollado por Texas Instruments. Incluye lo necesario para trabajar con dos Motores DC o un Motor de Pasos: Tarjeta de Control C2000 Piccolo F28035, Driver de motor DRV8412, Fuente de poder, guía de inicio rápido, software y motores.



Figura 6: "DRV8412-C2-KIT"
Fuente: [4]

A continuación se describen sus componentes.

- **Tarjeta de control CC2803X ISO DIMM REV 1.3.** Esta tarjeta realiza todas las operaciones de control. En ella se generan las señales PWM que comandan el funcionamiento de los MOSFET que se encuentran en la tarjeta de potencia. En la

memoria del microchip F28035 incorporado en la tarjeta de control, se ejecuta el código necesario para permitir que el kit funcione con la guía de comienzo rápido. Más adelante se explicarán los periféricos que incluye.



Figura 7: “Vista isométrica tarjeta de control”

Fuente: www.express-technology.com

▪ **Tarjeta de potencia DRV8412EVM REV F:**

- DRV8412: Es el driver integrado de la tarjeta de potencia, capaz de controlar un motor de pasos, dos motores DC o un motor DC con una configuración de puente H paralelo. Tiene las siguientes características:
 - Máxima tensión de entrada: 52 (V) DC
 - Máxima corriente de salida: 6 (A) peak con 3.5 (A) continuos.
 - Frecuencia de switching del driver de hasta 500 (kHz) [4].
- Sensor de corriente puente H, AB-CD: Cada pierna de los puente H tiene realimentación, la que consiste en una etapa de censado de corriente por derivación para mantener los valores de control.
- Switch de reinicio del puente H, AB-CD: En caso de alguna falla inesperada la placa trae dos switch de reinicio, uno para cada puente H.
- Jumper de modo DRV8412: Para una correcta configuración de la placa de potencia se debe verificar la posición de los jumper (la cual se describe más adelante).

- Puertos para codificador en cuadratura: La placa trae puertos para conectar un codificador en cuadratura de manera opcional, el cual se usa para medir posición.
- Ranura para tarjeta de control: La placa de potencia trae una ranura para insertar la tarjeta de control CC2803X ISO DIMM REV 1.3.
- Control de alimentación JP1: En caso de ser necesario se puede modificar la alimentación que se le da a la tarjeta de control desde la placa de potencia, ya que se puede alimentar por medio de la tensión suministrada por la misma placa de potencia o por medio de una alimentación externa.
- Control de alimentación +12 V: Si se desea alimentar la tarjeta de control con una fuente externa es aquí donde debe conectarse, pero, no olvidar que la fuente a conectar debe ser de 12 (V).
- Conectores puente H, AB-CD: Conectan los motores a utilizar, hay que tener en cuenta que motores se usaran para configurar la conexión de manera correcta.
- Bus de control de tensión regulada: Aquí la tensión regulada y controlada es dispersada por toda la placa.
- Bus de conexión DC: Alimentación de la placa de 24 (V).

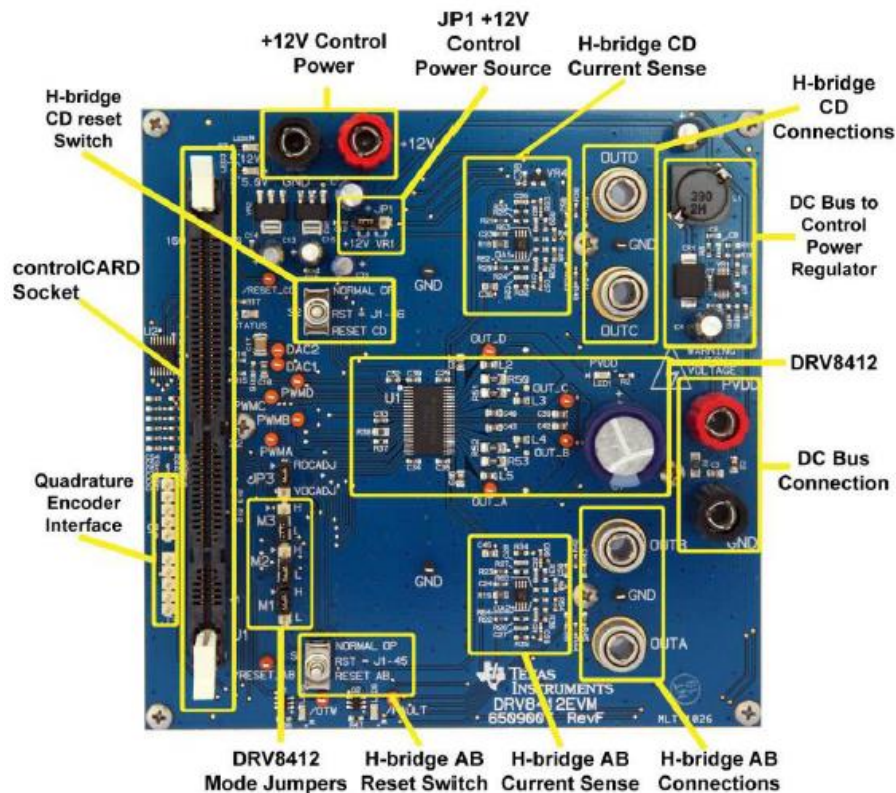


Figura 8: “Tarjeta de potencia”
Fuente: [4]

- Los siguientes motores están incluidos en el kit.
 - Motor de Pasos Bipolar: 4.2 (A) por fase, 8 terminales, 1.8°/paso
 - Dos Motores DC Brushed: 12 (V), 0.92 (A)
- Fuente de alimentación la cual se conecta a la tensión monofásica de 220 (V), que posteriormente se reduce a una tensión continua de 24 (V).
- Adaptador mini-USB a USB, este adaptador es el puente de comunicación entre el computador y la tarjeta de control.
- Como apoyo extra se encuentra una guía impresa, en la cual se describen algunos puntos importantes para la puesta en marcha del kit.
- Además se incluyen adaptadores para enchufes de tipo norteamericano y ucraniano, los que no se ocupan en este proyecto, pero que teniendo los adaptadores correctos pueden ser de ayuda.

Driver Integrado DRV8412

La placa de potencia tiene el driver integrado DRV8412 con puente completo dual, que tiene un gran sistema de protección.

El DRV8412 requiere de dos fuentes de alimentación, una de 12 (V) para GNDD y VDD, y otra de 50 (V) para PVDD. El driver puede operar a una frecuencia de switcheo de hasta 500 (kHz). El dispositivo tiene un sistema de protección de cortocircuito, de sobre-corriente, de bajo-voltaje y protección térmica. El circuito limitador de corriente previene que el dispositivo se apague durante el estado transitorio de la carga, como la puesta en marcha del motor [5].

La Figura 9 muestra de forma simplificada, un ejemplo de conexión del chip. Como se puede observar, el controlador se conecta a los pines PWM (A, B, C, D) por los cuales se reciben las señales de control de funcionamiento de los puentes H. Los pines RESET (AB, CD) son los que permiten realizar un reset por software desde el controlador. Los pines FAULT y OTW, que son activos bajos, son por los que se tiene una señal si hay alguna falla o sobrecarga detectada por los sensores, su función es proteger el controlador PWM y el sistema de control. Los pines M1, M2, M3 dependiendo de su estado, establecen el modo de funcionamiento de los puentes. Los pines OUT (A, B, C, D) entregan la señal de salida de los puentes y los pines PVDD (A, B, C, D) son por los cuales se alimenta cada puente. Para mayor información revisar el Anexo E, que incluye los datos técnicos del chip DRV8412.

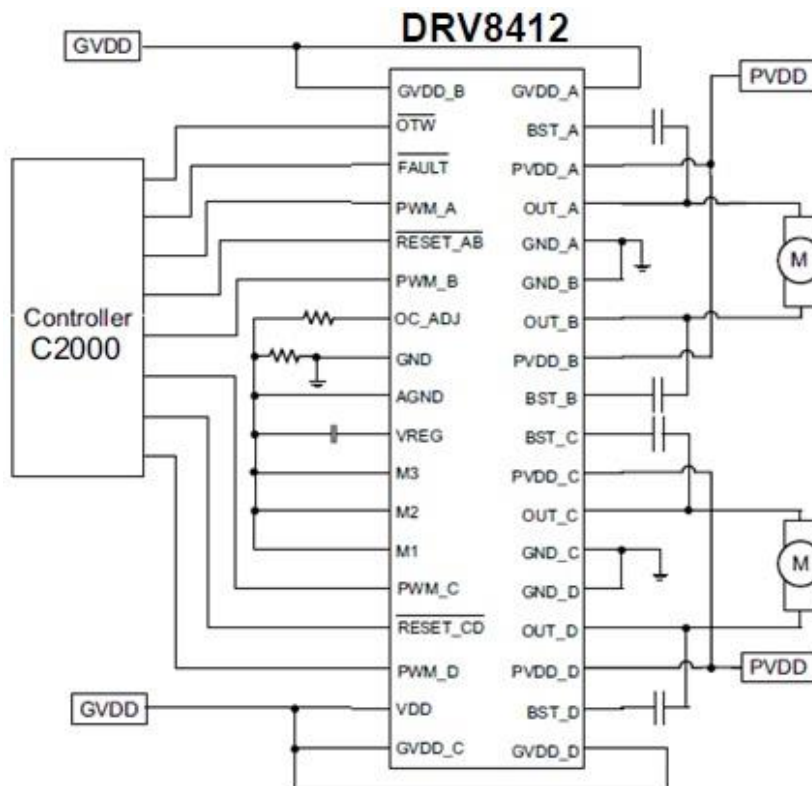


Figura 9: “Diagrama simplificado de una aplicación del driver DRV8412”
Fuente: [5]

Modos de Funcionamiento del Circuito.

El DRV8412 tiene cuatro modos diferentes de operación:

Modo 1. Dos puentes completos (dos entradas PWM para cada puente completo) o cuatro medios puentes con limitador de corriente.

Modo 2. Dos puentes completos (dos entradas PWM para cada puente completo) o cuatro medios puentes con apagado de enclavamiento si hay sobre-corriente.

Modo 3. Puente completo Paralelo con limitador de corriente.

Modo 4. Dos puentes completos (una entrada PWM para cada puente completo) con limitador de corriente.

En el **Modo 1 y 2**, PWM_A controla medio puente (A), y PWM_B controla la otra mitad (B).

En el modo puente completo paralelo (**Modo 3**), PWM_A controla los medios puentes A y B, PWM_B controla los medios puentes C y D, mientras que los pines PWM_C y PWM_D no son usados.

Los medios puentes A y B están sincronizados internamente, como ocurre también con los medios puentes C y D. OUT_A y OUT_B se conectan juntos al igual que OUT_C y OUT_D.

En el **Modo 4**, una señal PWM controla un puente completo. PWM_A controla los medios puentes A y B, y PWM_C controla los medios puentes C y D. En este modo la operación del medio puente B es complementaria a la del A, al igual que el medio puente D es complementario al C [5].

La Figura 10 muestra el circuito interno del chip DRV8412, y como es la conexión de los puentes H a los motores DC (para más detalle revisar el diagrama interno del chip en el Anexo E).

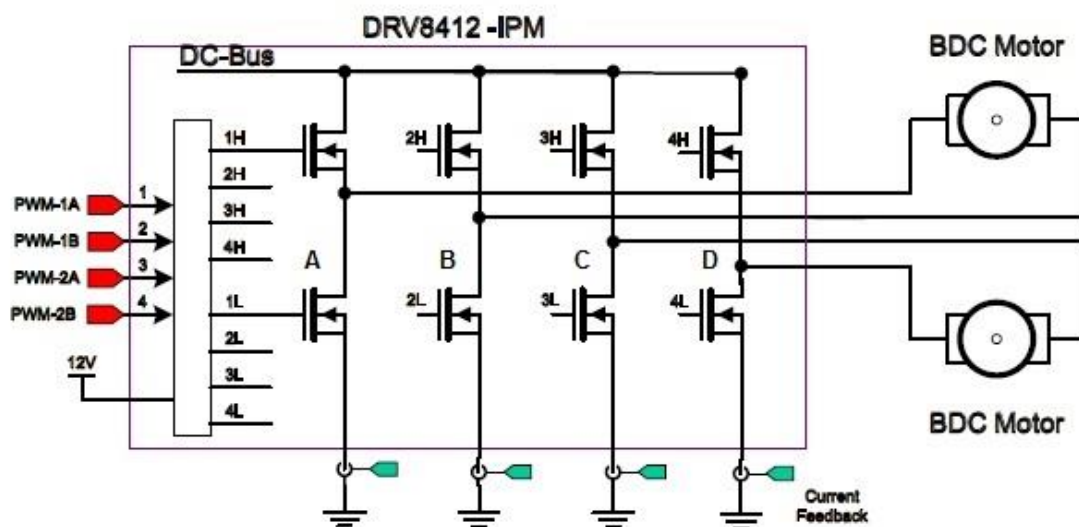


Figura 10: “Conexión de los motores DC a los puentes integrados”
Fuente: [4]

Tarjeta de Control TMS320F28035.

El Procesador Digital de Señales, de ahora en adelante nombrado como DSP (Digital Signal Processor), es el elemento fundamental en el sistema de control, ya que es el encargado de las tareas más importantes para el control del sistema. Dichas tareas son la adquisición de datos para su posterior procesamiento, y la generación de señales para poder llevar a cabo un algoritmo de control implementado por el usuario para los motores de DC y Stepper.

El DSP utilizado para el control del sistema es el TMS320F28035 del fabricante Texas Instruments, que se encontraba disponible en el departamento de electricidad y electrónica. Dicha tarjeta tiene un formato DIMM de 100 contactos y se conecta a la placa DRV8412 DMC. La tarjeta mencionada se muestra en la Figura 11 y la Figura 12.

El DSP es accesible y programable a través del puerto JTAG desde un PC, que debe tener instalado el software Code Composer Studio v4, el que permite programar el microprocesador mediante el lenguaje C.



Figura 11: “Tarjeta de control TMS320F28035 (vista anterior)”

Fuente: Elaboración propia

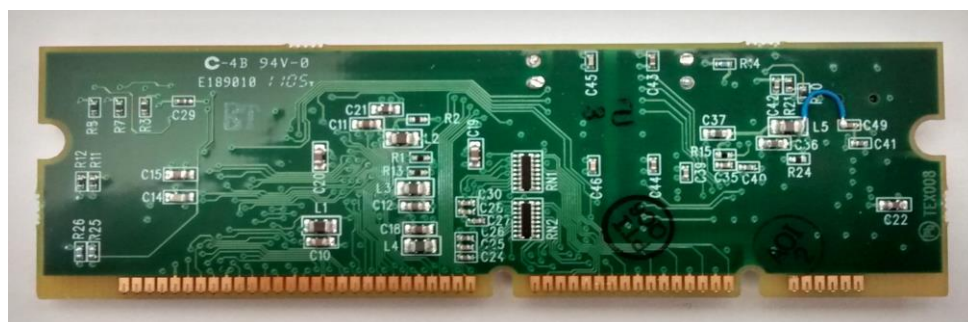


Figura 12: “Tarjeta de control TMS320F28035 (vista posterior)”

Fuente: Elaboración propia

A continuación se estudian y señalan los periféricos del DSP más importantes que se usan para el control de los motores, como son los módulos de generación PWM, el módulo eQEP y el convertidor analógico-digital (ADC, por sus siglas en ingles). Se hace una explicación superficial de cada uno de estos, ya que las configuraciones y valores que tienen por defecto son suficientes para las experiencias de control a desarrollar. En la Figura 13 se muestra el esquema de la tarjeta de control F28035, con los periféricos conectados a la CPU a través del bus.

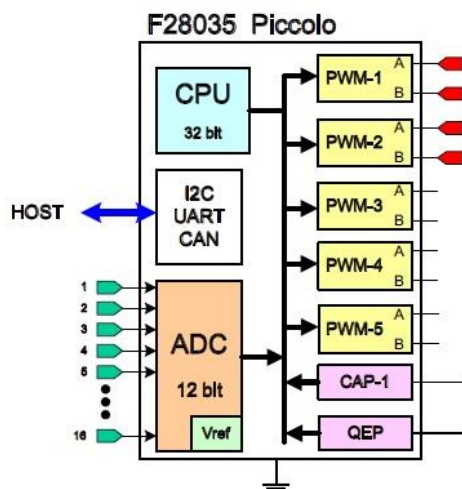


Figura 13: “Esquema de la F28035 y sus periféricos”

Fuente: [4]

Módulo PWM

Este módulo es el encargado de generar las señales PWM, del inglés “Pulse Width Modulation”, es decir, modulación por ancho de pulsos. Estas señales son necesarias para implementar el control de velocidad de los motores, ya que está basado en el disparo de hasta 12 ramas de IGBTs, usando las señales PWM para tal propósito [2].

Dos parámetros básicos son los que se podrán controlar a la hora de generar las señales: la frecuencia de la señal, que dependerá de la velocidad de control que se quiera realizar posteriormente; y el *duty cycle*, o ciclo de trabajo, que nos indica el tiempo que está activa la señal en relación al periodo de la misma. Su valor se encuentra comprendido entre 0 y 1 (0% a 100% del periodo). Este ciclo de trabajo viene detallado en la Figura 14.

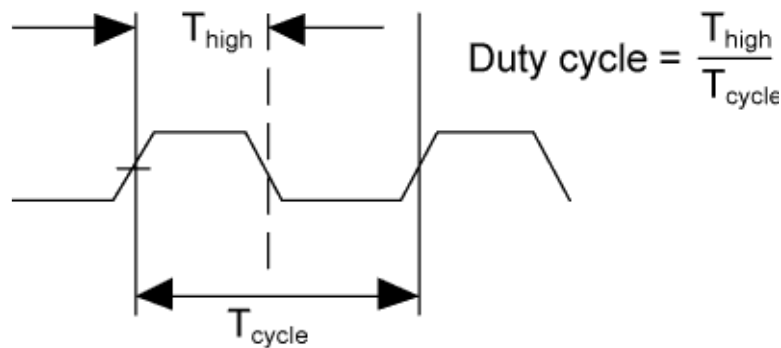


Figura 14: “Detalle del ciclo de trabajo en la onda PWM”

Fuente: electronics.stackexchange.com

Donde:

T_{high} = Es el tiempo en que la señal está activa

T_{cycle} = Es el periodo de la señal.

El DSP posee seis módulos de generación PWM independientes, obteniendo de cada módulo a su vez dos salidas de la misma frecuencia, pero con un ciclo de trabajo independiente para cada una de estas. Para el sistema que se utiliza es necesario generar cuatro señales PWM, para controlar los disparos de los IGBTs de cada una de las fases que posee el motor a controlar.

Módulo eQEP

Las siglas eQEP provienen del inglés “enhanced quadrature encoder pulse”, es decir, codificador de pulsos de cuadratura mejorado. Este módulo se utiliza para recibir y medir señales procedentes de un encoder, y obtener así información sobre la posición, velocidad y sentido de giro de una maquina rotativa [2].

En las experiencias a desarrollar, no se utiliza un encoder, aun así, están disponibles los puertos de entrada en la placa de potencia (DRV8412 EVM) para los codificadores si es que se requiere.

Convertidor Analógico-Digital

El convertidor analógico-digital (ADC) es un periférico capaz de convertir una señal analógica de tensión en una señal con valor en binario, de manera que pueda ser leída y manipulada por el microprocesador del sistema. Para ello necesita previamente una unidad de Sample and Hold (S/H), de muestreo y poder convertirlo posteriormente en el convertidor.

La tarjeta de control incluye dos módulos independientes, cada uno con 8 señales susceptibles de ser medidas. Cada módulo posee un multiplexor para acceder al único convertidor, y una unidad propia de Sample and Hold. Desde ambas unidades de S/H se accede al convertidor analógico-digital.

El esquema del periférico previamente explicado, viene detallado en la Figura 15. El convertidor analógico-digital es de 12 bits, por ello, se obtendrá un valor digital a la salida en el rango expuesto en la Figura 16.

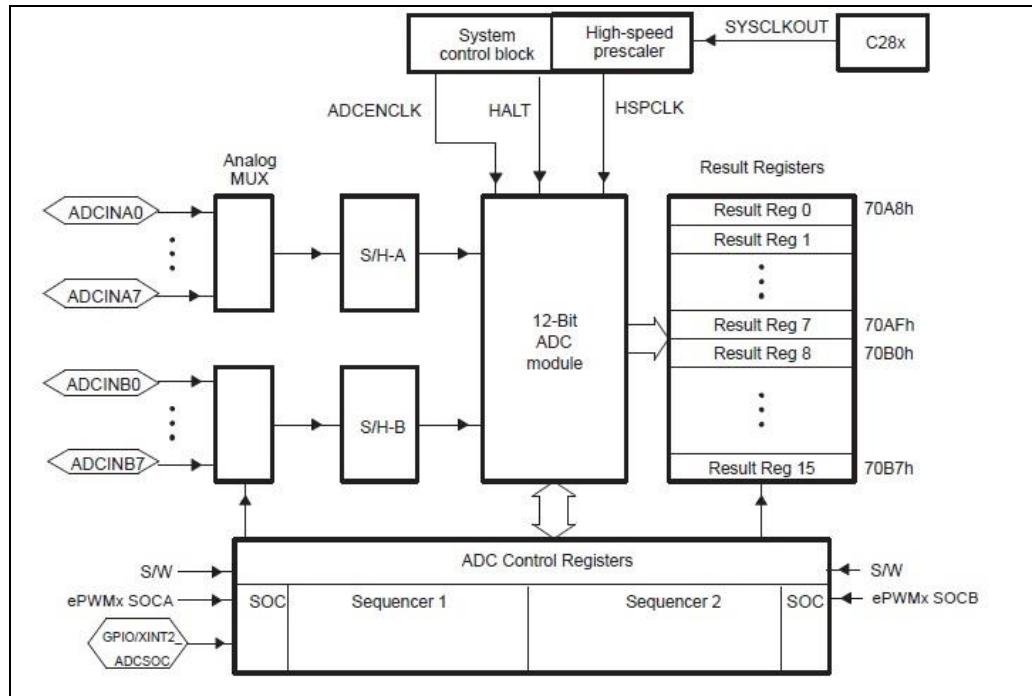


Figura 15: “Diagrama de bloques del módulo ADC”
Fuente: Texas Instruments Datasheet SPRU812A

Digital Value = 0,	when input ≤ 0 V
Digital Value = $4096 \times \frac{\text{Input Analog Voltage} - \text{ADCLO}}{3}$	when 0 V < input < 3 V
Digital Value = 4095,	when input ≥ 3 V

Figura 16: “Valor digital convertido en el ADC”
Fuente: Texas Instruments Datasheet SPRU812A

El registro “ADCLO”, visto en la figura anterior, corresponde a la referencia de tensión analógica con la que se compara la de entrada. Los registros “ADCTRL1”, “ADCTRL2” y “ADCTRL3” son los registros de control del ADC, encargados de configurar el modo de

conversión que se utilizara posteriormente. Cabe destacar también el registro “ADCMAXCONV” que fija el número de conversiones que se realizaran en cada secuencia.

La conversión es posible realizarla de diversas formas. El modo por defecto, en el que se va a usar el convertidor en este trabajo, es el modo “cascada”. En este modo, se muestran los 16 canales del registro como si fueran un único módulo, leyendo en una sola secuencia los 16 canales en orden, y almacenándolos en sus respectivos registros. Este modo es el más rápido, porque las unidades S/H funcionan en paralelo (ver Figura 15), con lo que se reduce el tiempo de conversión total a la mitad [6].

CAPÍTULO 2: DESCRIPCIÓN DEL SOFTWARE PARA PROGRAMAR

Este capítulo se enfoca al estudio de la herramienta de software utilizada para la implementación de algoritmos en el DSP, y de los componentes que intervienen en las funciones de una aplicación, que en ocasiones ocurren de manera transparente al usuario.

Para un correcto funcionamiento del kit se necesita tener los programas necesarios, además de contar con un computador capaz de trabajar a las exigencias de operación de los programas requeridos. Se debe tener en cuenta que no todas las versiones de los programas funcionan de manera efectiva, y tampoco funciona en los sistemas operativos mas recientes. Es por ello que se sugiere la versión de cada programa para ser instalada, además de la versión de Windows ideal para no tener problemas. Existen versiones de CCS que dicen ser compatible hasta con Windows 10, pero a la hora de compilar aparece un error que no se puede corregir, para evitar estos percances se hacen las sugerencias.

Code Composer Studio (CCS)

Code Composer Studio es un ambiente de desarrollo integrado que actúa como interfase de comunicación entre la tarjeta de desarrollo y la computadora, permitiendo programar al DSP de una manera más sencilla. Incorpora los siguientes módulos: Un compilador C, ensamblador, enlazador, simulador, depurador, etc., con características adicionales para visualización gráfica. Gracias al editor que proporciona, CCS permite la manipulación de programas en lenguaje C/C++, ensamblador o incluso el manejo de proyectos completos para la construcción de aplicaciones [9].

Code Composer Studio facilita el uso de diversas ventanas de observación y permite establecer puntos de ruptura dentro del programa, puntos de prueba para datos de I/O. También ofrece capacidades gráficas para monitoreo de datos, además de determinar el rendimiento de tiempo que se invierte en una porción del programa, entre otras características.

Para familiarizar al usuario con la escritura eficiente del código C/C++, Texas Instruments ofrece una colección de archivos o carpetas que pueden ejecutarse desde CCS y facilitarán la edición y mantenimiento de programas.

Dependiendo de las necesidades que se tengan, estos archivos se pueden usar como herramientas de aprendizaje que incluyen ejemplos desde Code Composer Studio. Los ejemplos explican los pasos que se requieren para inicializar el dispositivo y utilizar los periféricos. Además, se puede copiar y modificar su estructura con el fin de experimentar diferentes configuraciones de los periféricos. También se pueden usar como código base de una plataforma de desarrollo e integrarse dentro de un proyecto, usando las secciones necesarias de código y descartando el resto del programa. El usuario debe tener un nivel básico de conocimiento en lenguaje C, acceso a los ejemplos en cada una de las carpetas y la habilidad de cargar y ejecutar código usando Code Composer Studio.

El contenido de esta herramienta es muy extenso, por lo que sólo se comenta superficialmente el manejo de CCS. En el *Anexo A* se proporciona la *Guía Básica de Operación* que le permite al usuario instalar y usar la versión 4 de CCS, para generar, depurar, compilar y ejecutar el programa que se utiliza en este trabajo. Además, se encontrarán los pasos para administrar un proyecto, agregar programas de ejecución de una aplicación específica, controlar las variables en tiempo real, etc. En el CD que se anexa, se encuentra más información sobre el manejo e instalación de Code Composer Studio que complementa la información aquí descrita, tal como tutoriales, librerías, aplicaciones reales para uso en electrónica de potencia, manuales de programación, etc.

ControlSUITE

ControlSUITE es un conjunto de herramientas de infraestructura de software y software, diseñados para minimizar el tiempo de desarrollo. Desde los driver específicos del dispositivo y soporte del software para un completo sistema de ejemplos en sofisticados sistemas de aplicaciones, controlSUITE provee las librerías matemáticas, las bibliotecas de aplicaciones y utilidades además de ejemplos en cada etapa de desarrollo y evaluación; cabe destacar que el software “controlSUITE” es universal, es decir, está equipado para brindar apoyo en todos los kit de desarrollo de Texas Instruments, eso significa que, al instalarlo se debe (si se desea) seleccionar que partes del paquete instalar, ya que no se usan todas.

Interfaz de prueba rápida (DRV8412GUIv5)

La interfaz de prueba rápida es un software con el cual se puede hacer una prueba de funcionamiento del kit, y se pueden realizar aplicaciones básicas de movimiento, esta interfaz es capaz de mover los motores de escobilla de manera simultánea e independiente, controlando la velocidad de estos mediante un control de tensión o de corriente. La interfaz brinda también una gráfica en la que se observa el comportamiento de la corriente que circula por el motor.

Al igual que los motores de escobillas, se puede probar el motor de pasos. La interfaz es capaz de controlar la velocidad de este motor, la que es medida en pasos por segundo. Se debe destacar que se tiene control completo del motor ya que se puede hacer girar para ambos sentidos, y su rango de operación va desde -800 a 800 (pps). Por otro lado, provee la opción de limitar la corriente mediante un rango de operación, el cual va de 0.0 a 6.0 (A). Incluye una sección en la que se puede hacer girar al motor por pasos, la velocidad en la que se haga, y la configuración que adopte para hacerlo, ya que mediante la interfaz se puede configurar, ya sea, en paso completo, medio paso, cuarto de paso, 8 micro-pasos/paso, 16 micro-pasos/paso, 32 micro-pasos/paso, 64 micro-pasos/paso y 128 micro-pasos/paso.

ControlSUITE incluye dos versiones de la interfaz (DRV8412GUIv2 y DRV8412GUIv9) con las que no se pudo trabajar por errores de conectividad. Por el contrario, la guía de interfaz que incluye el CD funciona correctamente con el kit de desarrollo.

CAPITULO 3: PROGRAMACIÓN DE PERFILES DE TRABAJO.

En este capítulo se presenta la forma que tiene el programa (código) con el que se trabaja en las distintas aplicaciones de control de movimiento de los motores.

Las experiencias que se plantean para el control del motor DC y el de pasos, se basan en los ejemplos que incluye el software ControlSUITE.

Bloques macro.

Para trabajar con la tarjeta de control, Texas Instruments, utiliza la librería DMC (de sus siglas en inglés, control digital de motor) en la serie de dispositivos C2000. Esta librería está compuesta de las funciones C (o macros) que representan como bloques modulares una agrupación de funciones matemáticas y de configuración de los distintos periféricos. Estos bloques se clasifican en tres categorías: Transformación-Estimación, Control y Controladores de periféricos [7].

Al mismo tiempo, el programa se compila utilizando niveles incrementales. Esto se hace, ya que independiente de cuanta planificación se haga en un proyecto de ingeniería, lo más probable es que el programa final no funcione la primera vez, porque se pasa por alto una variable dependiente o una condición. Esto haría que se destine mucho tiempo en encontrar y resolver la falla.

Para evitar esto, el programa viene estructurado en niveles de compilación, que permiten al usuario depurar desde el nivel más bajo (sólo lo esencial) hasta el nivel más alto (todo el programa) y verificar paso a paso el correcto funcionamiento del código. Si se encuentra alguna falla en un nivel permite corregirla sin riesgo de perder la configuración de los niveles anteriores.

En la Figura 17 se muestra el diagrama de flujo del programa que se utiliza para trabajar con el Motor DC.

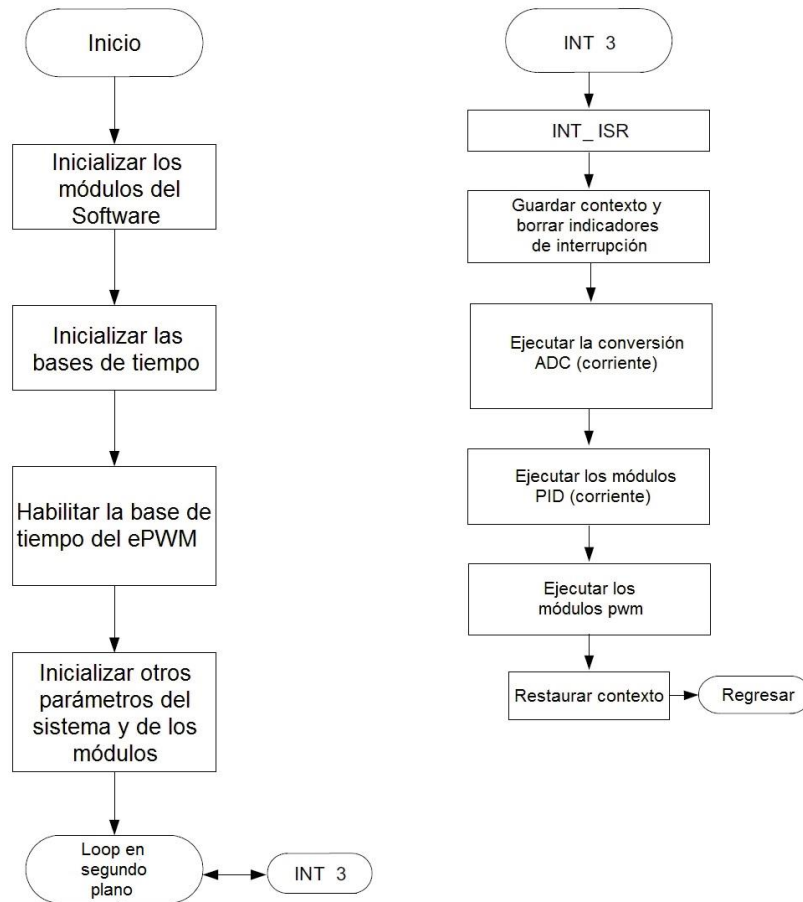


Figura 17: “Diagrama de flujo control de Motor DC”

Fuente: [7]

Como se puede observar, está estructurado en dos columnas, el proceso y la interrupción.

Lo primero que se hace al iniciar el programa es inicializar los módulos con los que trabajará el software.

Seguidamente, se inicia la referencia temporal que se tendrá en las distintas secciones del código.

A continuación se habilita la base de tiempo del ePWM que dentro de todas las tareas que se hacen en esta sección, las más importantes serían: Configurar la frecuencia o el período del contador, sincronizar las bases de tiempo entre los distintos módulos, configurar el comportamiento del contador cuando el dispositivo este detenido por una emulación, sincronizar el reloj del sistema con el de los módulos. Se inician posteriormente los parámetros secundarios del sistema y de los módulos.

Luego se suspende temporalmente la ejecución del proceso y se dirige a la interrupción.

Lo primero que aparece es INT_ISR. ISR es rutina de servicio de interrupción, de sus siglas en inglés, es una subrutina que la CPU debe ejecutar como respuesta a una determinada petición de interrupción de su tarea actual, que en este caso al ser INT, es por una instrucción que genera el programa.

Después se guarda el contexto, para permitir la interrupción y que posteriormente se continúe desde este punto y además se borran las banderas de interrupción, ya que si no se hace, la CPU estará siempre interrumpida.

Posteriormente, se ejecuta la conversión análogo-digital del valor de corriente que entrega el sensor. Se trabaja este dato en el módulo PID para luego pasar al módulo PWM y así controlar la velocidad del motor.

Terminado lo anterior, se restaura el contexto y se regresa.

La Figura 18, muestra el diagrama de flujo del programa que se utiliza para trabajar con el motor de pasos.

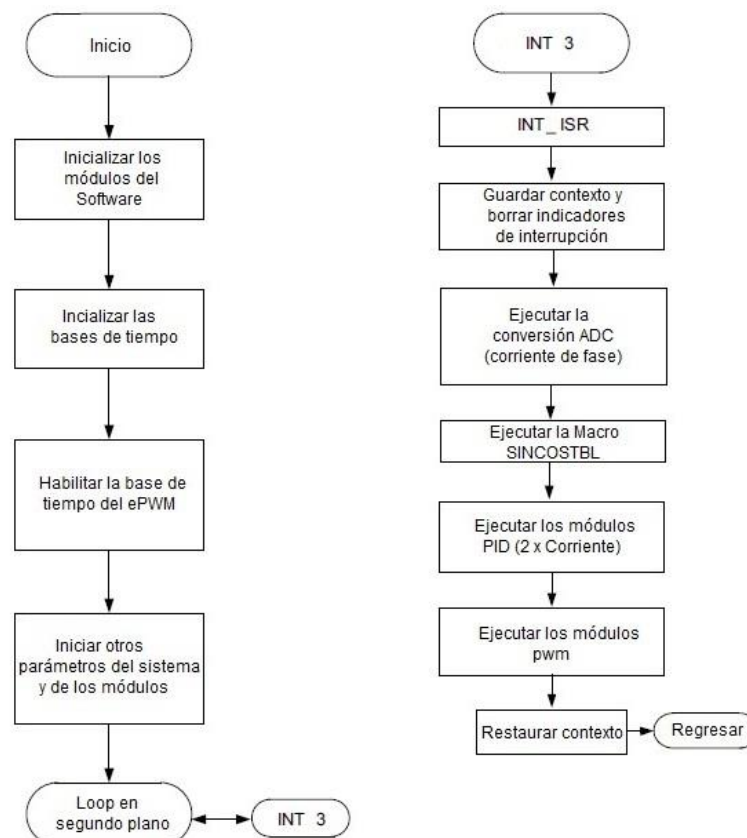


Figura 18: “Diagrama de flujo control de motor de pasos”

Fuente: [8]

Como se observa, los dos diagramas son similares. Lo que se agrega en el segundo es la macro SINCOSTBL. Esto se utiliza cuando se hace control por micropasos. Al aplicar una función seno en una fase y una función coseno en otra, se tiene una rotación suave y un posicionamiento preciso.

Niveles de compilación incremental de cada experiencia.

Como se mencionó anteriormente, el proyecto tiene una estructura de compilación en niveles incrementales, para este caso, consisten en cuatro niveles. Los cuatro niveles están diseñados para testear y verificar la correcta operación de los principales módulos utilizados en el sistema. La Tabla 1 resume los módulos probados y utilizados para cada nivel de compilación del sistema al trabajar con el motor DC.

Tabla 1: “Módulos usados en el sistema incremental de compilación (Brushed DC) [7]”

Módulo de Software	Nivel 1	Nivel 2	Nivel 3	Nivel 4
PWMDAC_MACRO	(1)	(1)	(1)	(1)
RC_MACRO	(1)	(1)	(1)	(1)
PWM_MACRO	(2)	(1)	(1)	(1)
A/D CONVERSION		(2)	(1)	(1)
PID_MACRO			(2)	(1)

(1) Este módulo es usado en este nivel.

(2) Este módulo es probado en este nivel.

- **Nivel 1:** En este nivel, se mantiene el motor desconectado, y se hace un mínimo chequeo del sistema. Se testea la macro que genera PWM y se observa la señal de salida.
- **Nivel 2:** Esta fase verifica la conversión análogo-digital. Se conecta el motor a la placa, pero se hacen pruebas sin carga en el eje. Sólo se puede testear un motor a la vez.
- **Nivel 3:** Este nivel verifica la regulación de corriente realizada por el módulo PID_REG3. Se varía la carga en el eje del motor para realizar las pruebas.
- **Nivel 4:** En este nivel se conectan los dos motores al dispositivo. Se verifica el concepto de multi-axis, con los dos motores girando simultáneamente y en control de corriente.

Para las experiencias que se realizarán con los motores DC, será de interés trabajar con los niveles dos y cuatro.

Al igual que el anterior, se tendrá una estructura de compilación en niveles en el programa de control del motor de pasos, que en este caso son tres niveles. La Tabla 2 resume los módulos activos y en prueba para cada nivel de compilación del sistema.

Tabla 2: “Módulos usados en el sistema incremental de compilación (Stepper) [8]”

Módulo de Software	Nivel 1	Nivel 2	Nivel 3
PWMDAC_MACRO	(1)	(1)	(1)
RC_MACRO	(1)	(1)	(1)
RG_MACRO	(1)	(1)	(1)
SINCOSTBL_MACRO	(2)	(1)	(1)
PWM_MACRO	(2)	(1)	(1)
Conversión A/D		(2)	(1)
PID_MACRO			(2)

(1) Este módulo es usado en este nivel.

(2) Este módulo es probado en este nivel.

- **Nivel 1:** En este nivel se mantiene el motor desconectado. Se hace una mínima revisión de las interrupciones del sistema, del módulo sin/cos y del generador PWM como también de los puentes H de la DRV8412.
- **Nivel 2:** En este nivel se verifica la conversión análogo-digital y se conecta el motor a la placa.
- **Nivel 3:** En este nivel se verifica la regulación de corriente realizada por el módulo PID_REG3.

Para las experiencias que se realizarán con el motor de pasos, será esencial trabajar con el nivel 3, ya que éste habilita todas las Macro del programa, por lo que se tendrá un mejor control del motor y se podrá trabajar en micropasos.

CAPITULO 4: RESULTADOS Y DISCUSIÓN.

Una vez planteada la forma de trabajo se procede a construir las tres maquetas.

- La primera es el gabinete. Este contiene la tarjeta de control, la tarjeta de potencia y los motores. Los ejes de los motores dan al exterior para que se pueda acoplar a estos las cargas con las que se quiera trabajar, que en este caso son las otras dos maquetas. A un costado tienen los bornes para conectar el adaptador de corriente, y cuatro interruptores que su función es cambiar la conexión de un tipo de motor a otro (de DC a Stepper y viceversa) sin tener que desconectar los cables de los bornes de la tarjeta de potencia. La caja está cubierta por un acrílico transparente para poder observar los LED indicadores de operación y falla.

La Figura 19 muestra un bosquejo del diseño que se realizó. El esquema detallado del diseño se encuentra en el ANEXO F: ESQUEMA DISEÑO GABINETE

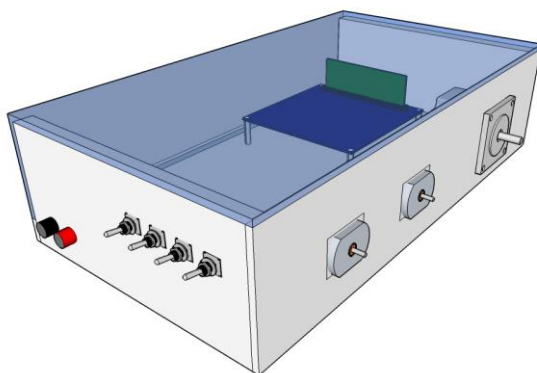


Figura 19: “Vista en perspectiva diseño gabinete”
Fuente: Elaboración propia

En la Figura 20 se muestra el diagrama de conexionado eléctrico del gabinete:

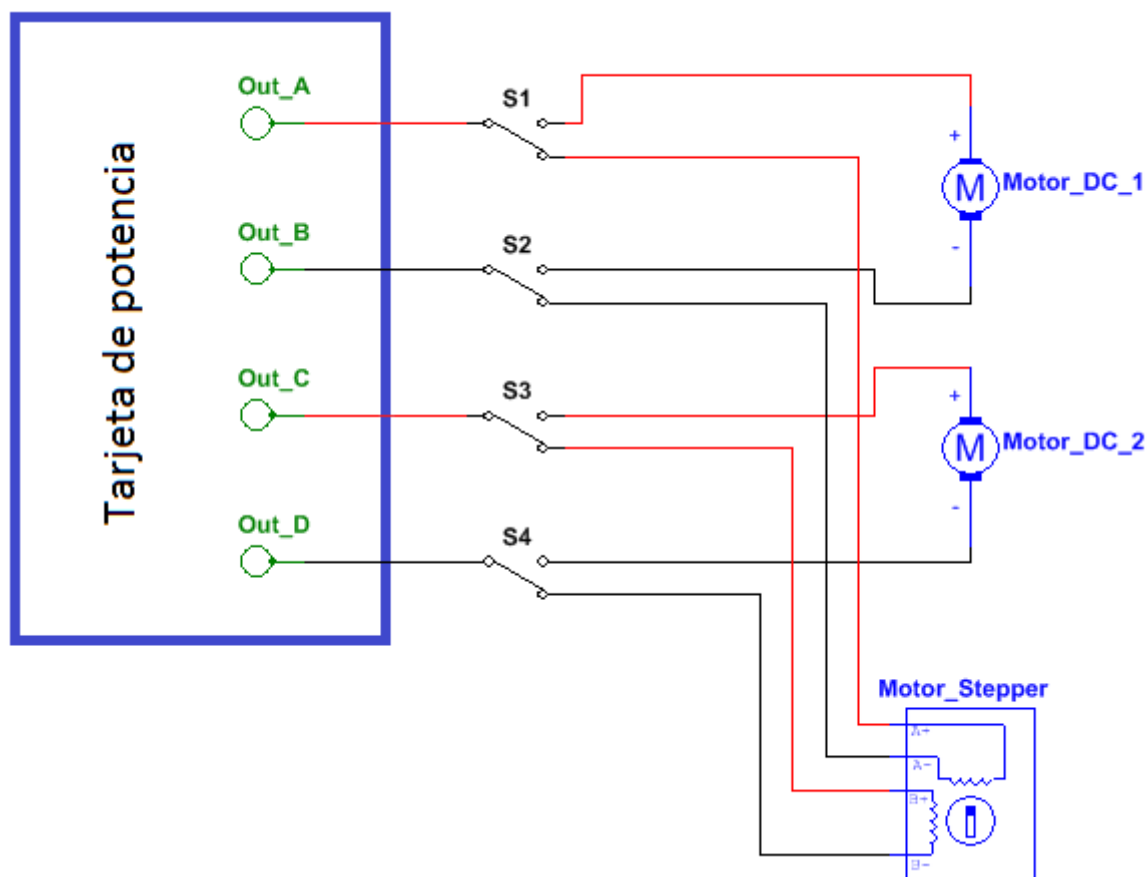


Figura 20: “Diagrama conexionado eléctrico del gabinete”
Fuente: Elaboración propia.

En la Figura 20 se observa que los switches cumplen el papel de seleccionar el tipo de motor a utilizar para trabajar, pero para tener un correcto funcionamiento se debe asegurar que antes de comenzar los 4 switch deben estar en la misma posición (dependiendo de si se usara los motores DC o el motor Stepper) para obtener un correcto funcionamiento y evitar errores.

Cabe destacar que los interruptores son de tipo selector, ya que de esta forma es más sencillo cambiar del motor de pasos a los motores DC o viceversa, esto nos evita el proceso de abrir el gabinete y cambiar manualmente el conexionado de los motores cada vez que se requiera.

Las otras dos maquetas son las de experiencias de control. Éstas se definieron considerando el tipo de motor y sus aplicaciones típicas. Para el motor de pasos se definió una maqueta que simula un elevador, ya que de esta forma se puede apreciar el movimiento secuencial del motor, además de la variabilidad de carga aplicada con la que se busca obtener el comportamiento del motor al ser sometido a diferentes tipos de carga.

Para el motor DC se definió una maqueta que simula una cinta transportadora, ya que esta es una aplicación típica que se asocia al motor DC por la relación velocidad/tensión, además, con ella es posible observar el comportamiento del motor al ser sometido a variaciones de carga y velocidad.

Cabe destacar que en ambas maquetas es posible hacer un control punto a punto, en la maqueta del elevador es de posición, mientras que en la maqueta de la cinta transportadora es de velocidad.

- La maqueta del elevador, consiste en una torre de medio metro por la que se desplaza un carro de forma vertical. Consta de tres poleas y un contrapeso. El eje del motor se conecta en la parte inferior de la maqueta, a un costado de la torre.

La Figura 21 muestra un bosquejo del elevador. El esquema detallado se encuentra en ANEXO G: ESQUEMA DISEÑO MAQUETA ELEVADOR

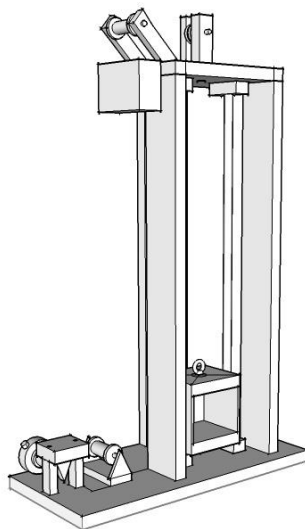


Figura 21: “Vista en perspectiva diseño elevador”
Fuente: Elaboración propia

- La tercera maqueta es la cinta transportadora, que se une por medio de poleas al eje del motor, tiene un ángulo de elevación fijo.

La Figura 22 muestra el bosquejo de la cinta transportadora. El esquema detallado se encuentra en el ANEXO H: ESQUEMA DISEÑO CINTA TRANSPORTADORA

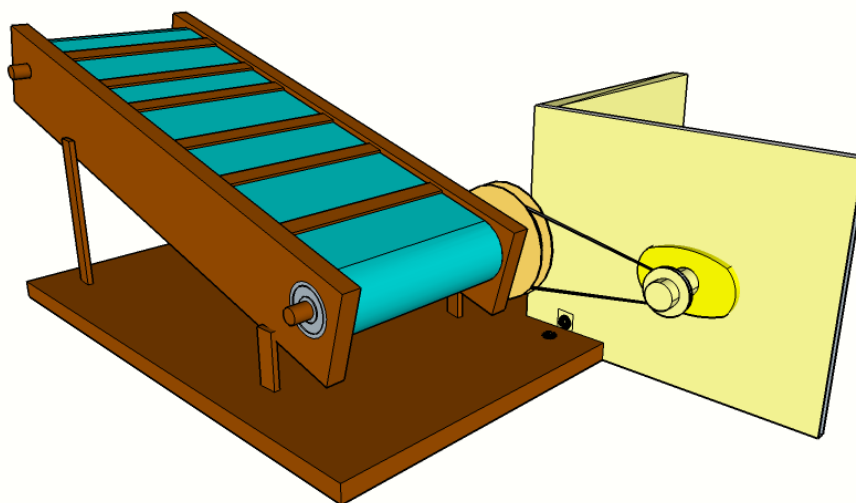


Figura 22: “Vista en perspectiva diseño cinta transportadora”
Fuente: Elaboración propia

En la Figura 22 se observa cómo se acoplan ambas maquetas a través de una escuadra atornillable, para la transmisión de movimiento se utiliza una correa que une la polea del “motor DC 2” y el eje de la cinta transportadora, cabe destacar que para reducir de cierta manera la velocidad aplicada desde el motor hacia la cinta se fabrica una polea más grande para el eje de la maqueta, estas dos poleas tienen una relación 2:1, lo que significa que la polea del motor tiene la mitad de la circunferencia que la polea del eje de la cinta, esto en la práctica se puede entender de la siguiente forma: por cada vuelta que gire el eje de la cinta, el eje del motor habrá girado 2 vueltas.

Para la construcción de esta maqueta fue necesaria la implementación de rodamientos que se integraron en los ejes de la cinta transportadora logrando así reducir el roce considerablemente, por otro lado las poleas aportan un roce el cual se debe a que no están bien centradas, produciendo un efecto de elipse al momento de girar. Para la cinta se utilizó inicialmente goma, pero debido a que ésta se pegaba a la tabla de base el motor no lograba moverla, por lo que se implementó género con el cual se obtuvo un mejor resultado.

Trabajo con motor de pasos.

Para definir la secuencia de desplazamiento que se usa al trabajar con el motor de pasos, primero se tiene que medir el desplazamiento total que hace la caja en la maqueta y cuál es la altura de la caja para establecer cuantos niveles tendrá el elevador, refiriéndose a nivel como un movimiento y una pausa.

La caja del elevador tiene una altura de 9 (cm) y la altura de la torre es de 43 (cm).

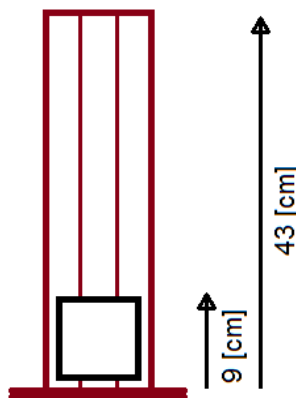


Figura 23: “Esquema representación alturas torre y caja”

Fuente: Elaboración propia

Las medidas que se le dieron a la maqueta fueron cálculos estimados, considerando que debían ser más de dos niveles de desplazamiento porque se necesita hacer una secuencia de control y al tener solamente dos niveles sería más bien una plataforma de elevación que un ascensor teniendo un accionamiento de todo nada.

Con los datos que se tiene se determina que el desplazamiento total es de 34 (cm), por lo que al dividir el desplazamiento por la altura de la caja se tiene aproximadamente 3. Entonces, para que sea uniforme, el desplazamiento de la caja en cada nivel es de 11 (cm).

Alto caja= 9 (cm)	9+11+11+11=42 (cm)
Alto torre= 43 (cm)	
Desplazamiento por nivel= 11 (cm)	

Ahora se debe determinar cuanto giro o cuantas revoluciones debe hacer el eje del motor para que haya un desplazamiento de 11(cm).

El cilindro que va unido al eje tiene un perímetro de 5.5 (cm)

1 Revolución → 5.5 (cm)

2 Revoluciones → 11 (cm)

Por lo que se deduce que cada 2 revoluciones, la caja se desplaza un “piso”.

Con ayuda de la interfaz de prueba rápida [10], se obtienen los distintos valores con los que se trabajará.

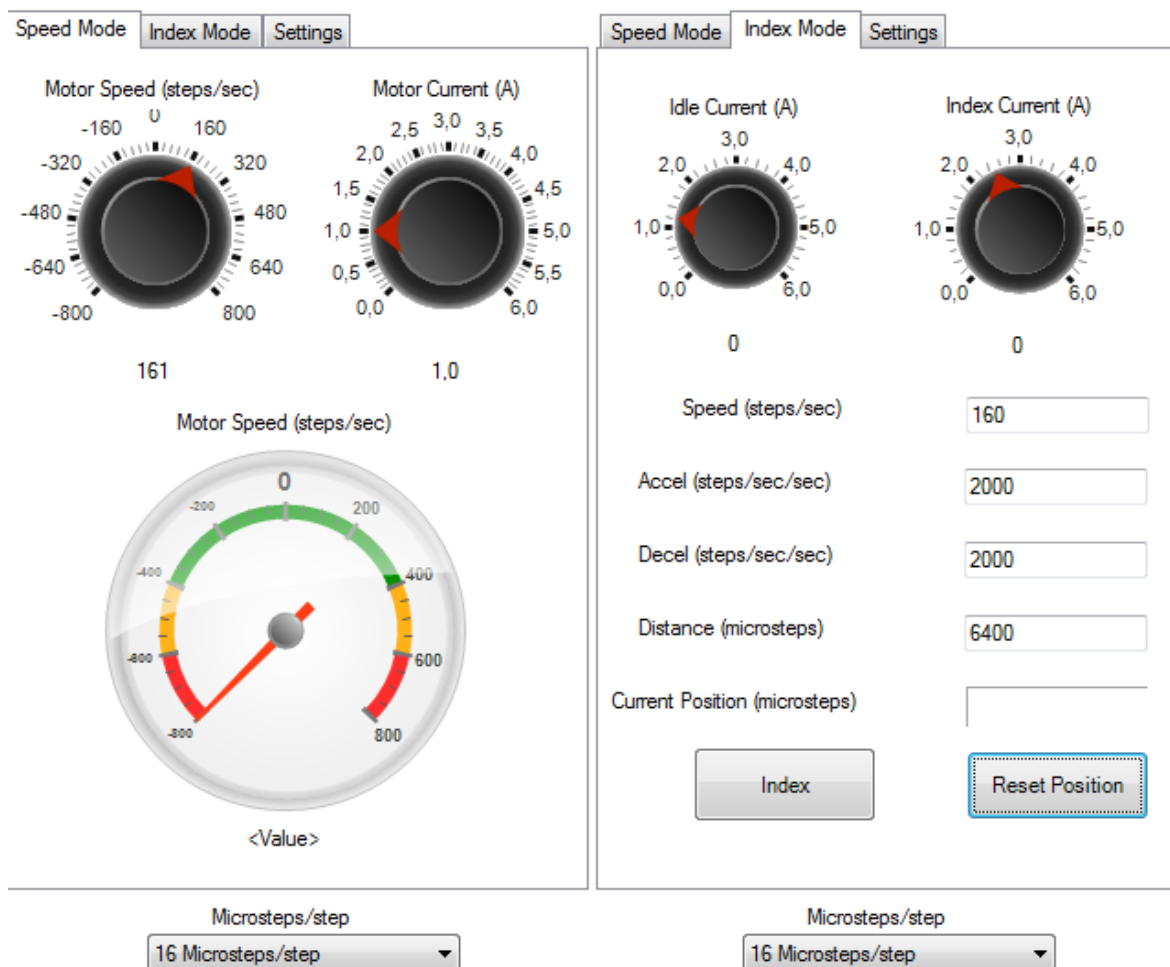


Figura 24: “Datos técnicos obtenidos de la interfaz de prueba rápida”
Fuente: DRV8412GUIv5

En la Figura 24 muestra la velocidad máxima del motor que es de 800 (pasos/segundo), ésta la define el programa. Para que tenga un ascenso y descenso suave, se utiliza el 20% de la velocidad máxima que son 160 (pasos/segundo).

En cada paso gira 1.8° por lo que una revolución es equivalente a 200 pasos.

Se configura para que trabaje en 16 (micropasos). Con lo que ahora el motor debe dar 3200 pasos para girar una revolución.

Se escogieron estos valores porque con estos el motor tenía un movimiento suave y no zumbaba.

Como no se dispone de codificador óptico para medir el desplazamiento, se calcula el tiempo de movimiento, de los datos que se tienen:

Tabla 3: “Datos obtenidos de la interfaz de prueba rápida”

Paso	1.8°
Revolución	200 pasos
Velocidad	160 (pasos/segundo)
Configuración	16 (micropasos)
Desplazamiento	2 Revoluciones

Se hacen los siguientes cálculos:

$$\begin{aligned}
 &200 \text{ (pasos)} \cdot 16 = 3200 \text{ (micropasos)} \\
 &3200 \text{ (micropasos)} \cdot 2 = 6400 \text{ (micropasos)} \\
 &16 \cdot 160 \text{ (pasos/segundo)} = 2560 \text{ (micropasos/segundo)} \\
 &t = 6400 \text{ (micropasos)} / 2560 \text{ (micropasos/segundo)} \\
 &t = 2.5 \text{ segundos}
 \end{aligned}$$

La explicación de los cálculos es la siguiente:

Como se mencionó anteriormente se configura el motor en 16 micropasos, por eso una revolución es igual a 3200 micropasos. Para que la caja del ascensor se desplace 11 (cm) (en ascenso o descenso) el cilindro en el eje debe dar 2 revoluciones, porque las dos revoluciones son igual a 6400 micropasos. La modificación también afecta a la velocidad, por lo que ahora es 2560 (micropasos/segundo). Al dividir los dos valores se obtiene el tiempo de desplazamiento, de esta forma se obtiene que con 2.5 segundos el eje gira 2 revoluciones, o sea, que hay un desplazamiento de 11 (cm) de la caja.

De forma experimental no se logra que el eje gire 2 revoluciones en ese tiempo, ya que no se está considerando la aceleración, pero se resuelve al sumar 0.4 segundos.

La secuencia considera una pausa de 5 segundos luego de cada movimiento. De manera gráfica se muestra el desplazamiento de la caja del elevador, en la Figura 25. El gráfico posición-tiempo se elaboró con mediciones directas de distancia en la maqueta, considerando el tiempo de movimiento (2.9 segundos) y las pausas.

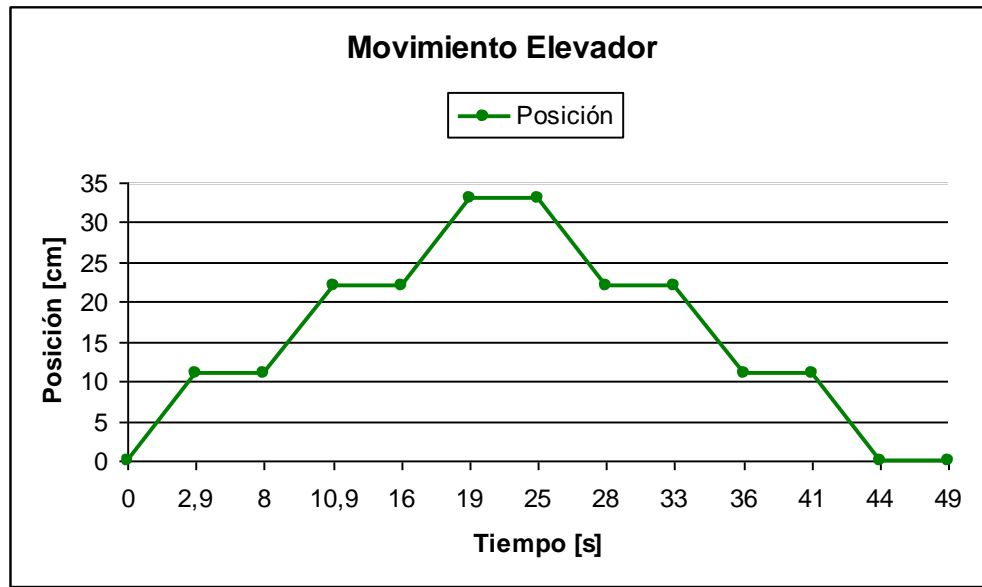


Figura 25: "Gráfico desplazamiento en el tiempo del elevador"
Fuente: Elaboración propia

Una vez determinados todos los datos y definida la secuencia de desplazamiento, se escribe el código que se muestra en el ANEXO C: CÓDIGO SECUENCIA DE ASCENSO Y DESCENSO y se agrega al proyecto del motor de pasos en el software Code Composer Studio. El procedimiento de la experiencia de control punto a punto con el motor de pasos, se muestra en el ANEXO B: GUÍA DE LABORATORIO. **Error! No se encuentra el origen de la referencia.**

En la Figura 26, Figura 27 y Figura 28, se muestran las tensiones y corrientes del motor de pasos medidas con el osciloscopio. Se varía la cantidad de pasos por paso manteniendo la velocidad del motor y la amplitud de la corriente.

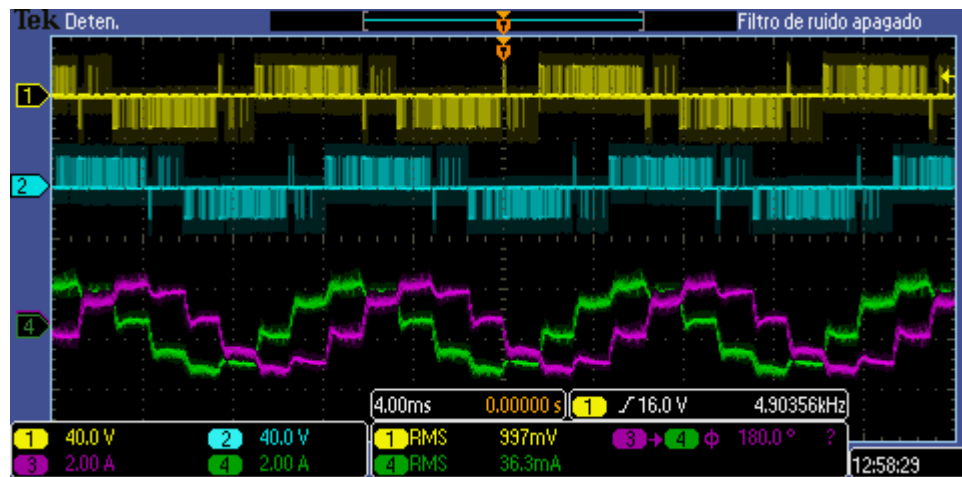


Figura 26: “Tensiones de salida puentes H y Corrientes de cada fase del motor de pasos al trabajar en **Medio Paso**, 40% velocidad”

Fuente: Medición Osciloscopio Tektronix MSO 2024B

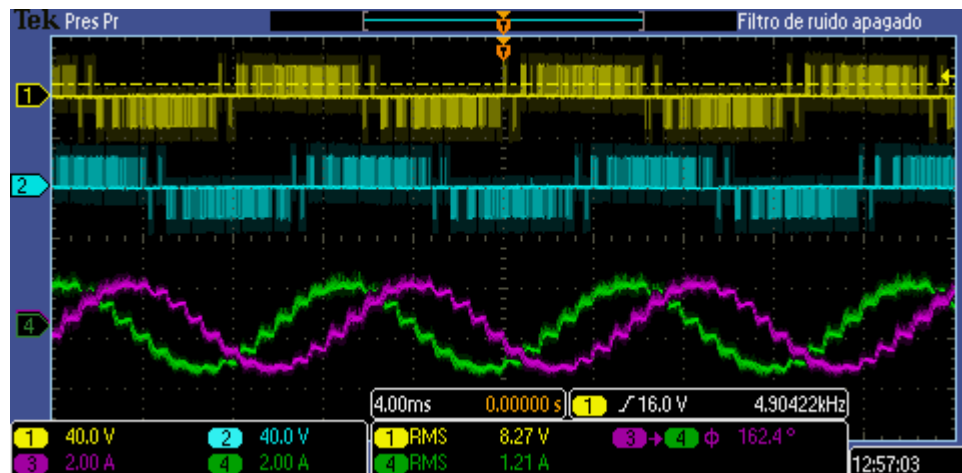


Figura 27: “Tensiones de salida puentes H y Corrientes de cada fase del motor de pasos al trabajar en **Cuarto de Paso**, 40% velocidad”

Fuente: Medición Osciloscopio Tektronix MSO 2024B

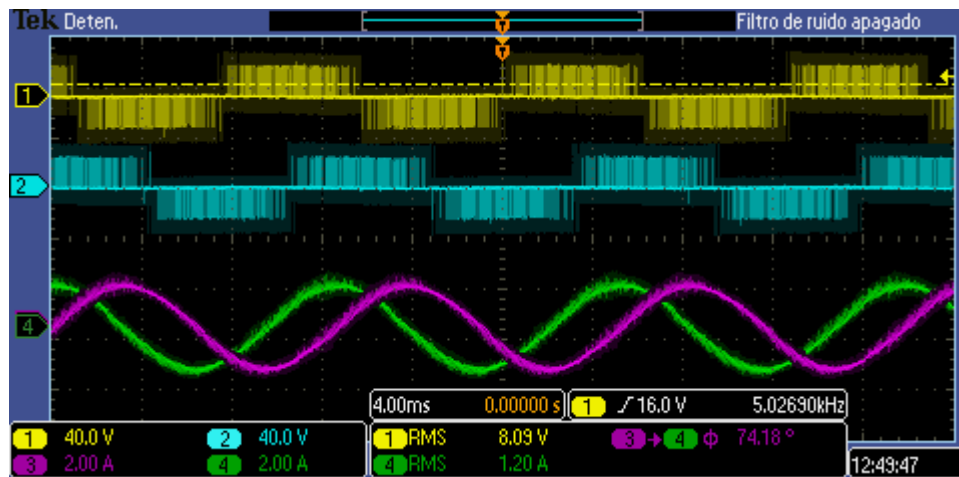


Figura 28: “Tensiones de salida puentes H y Corrientes de cada fase del motor de pasos al trabajar en **16 Micropasos**, 40% velocidad”

Fuente: Medición Osciloscopio Tektronix MSO 2024B

En las figuras se observa que la tensión de salida de cada puente H es una señal que varia tanto su ancho de pulso como su estado alto y bajo. Al hacerlo a una tasa lo suficientemente rápida, la corriente en el motor parece comportarse como una señal analógica. Las corrientes de fase corresponden a dos sinusoides desfasadas una con respecto a la otra para obtener el control por micropasos en el motor. Los pasos por paso son notorios en la onda a valores bajos, al aumentar la cantidad de micropasos la forma de la onda tiende a volverse más continua y este efecto se refleja en el movimiento del eje del motor que se vuelve más suave.

Al comparar la Figura 29 con la Figura 30 se observa el efecto que tiene cambiar el sentido de giro del motor de pasos, manteniendo la corriente y velocidad.

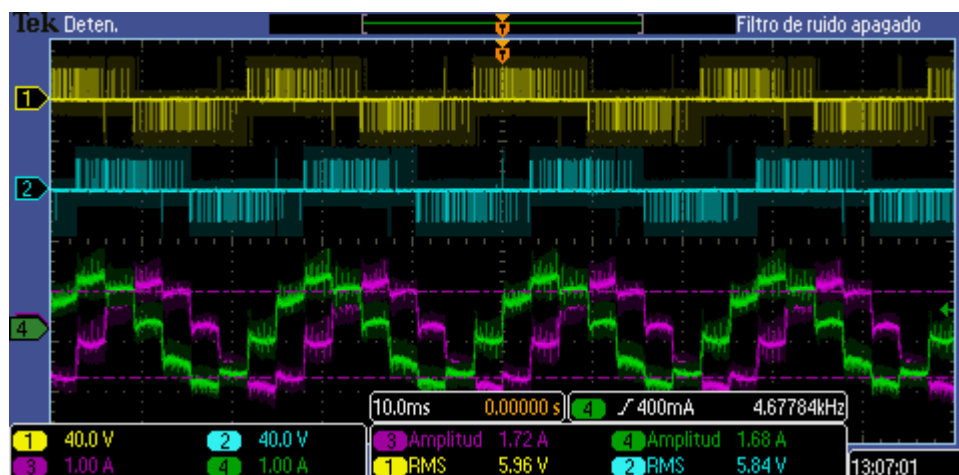


Figura 29: “Tensiones de salida puentes H y Corrientes de cada fase del motor de pasos al trabajar en **Medio Paso**, 40% velocidad, sentido giro horario”

Fuente: Medición Osciloscopio Tektronix MSO 2024B

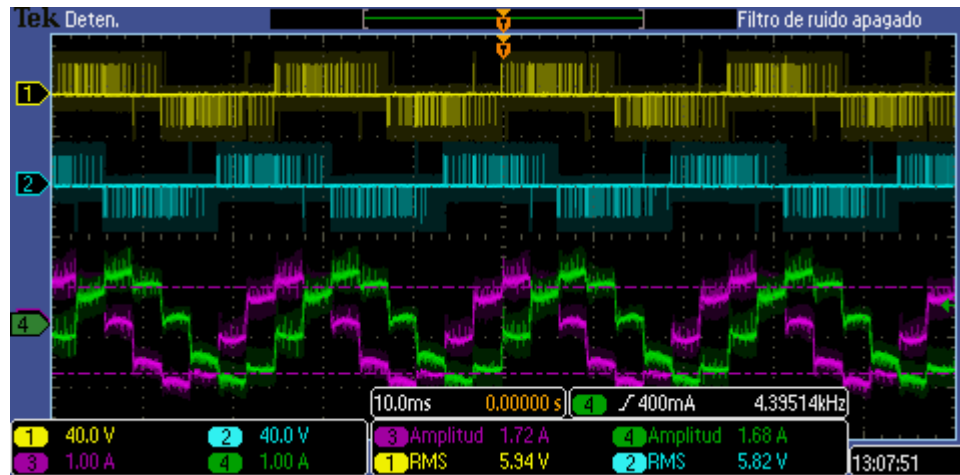


Figura 30: “Tensiones de salida puentes H y Corrientes de cada fase del motor de pasos al trabajar en **Medio Paso**, 40% velocidad, sentido giro antihorario”
Fuente: Medición Osciloscopio Tektronix MSO 2024B

Como se observa, se invierte el desfase de las señales de corriente, o sea, la señal verde está en atraso respecto a la señal morada, al girar el eje del motor en sentido antihorario.

La Figura 31 muestra la señal de corriente al trabajar en 8 micropasos, pero con una velocidad del 5%.

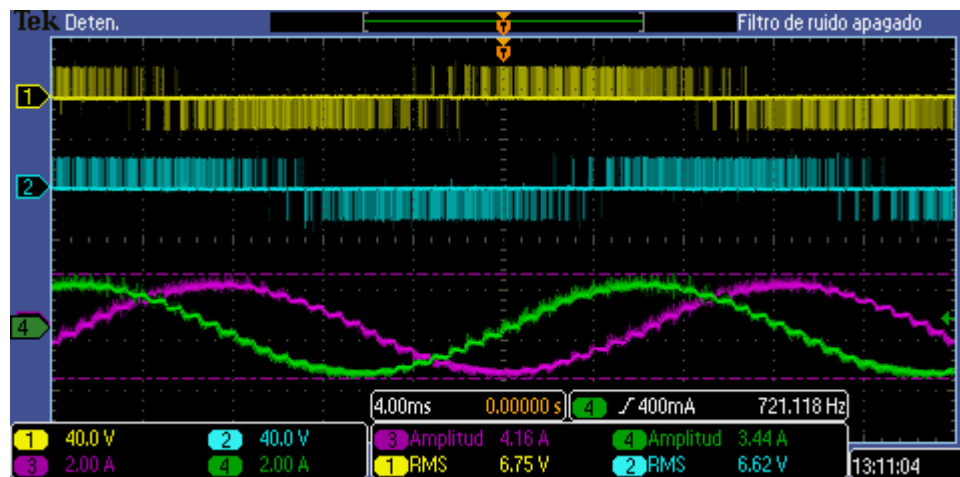


Figura 31: “Tensiones de salida puentes H y Corrientes de cada fase del motor de pasos al trabajar en **8 Micropasos**, 5% velocidad”
Fuente: Medición Osciloscopio Tektronix MSO 2024B

Se observa que al reducir la velocidad de giro, la frecuencia de la corriente también lo hace, comprobando que la frecuencia de salida del PWM depende directamente de la velocidad de referencia que se ingresa al sistema. El cambio de frecuencia es gradual, ya que la velocidad de referencia tiene asociada una rampa en el programa.

Trabajo con motor DC.

El motor DC tiene un funcionamiento más simple que el de pasos, refiriéndose a que son menos variables de control a tomar en consideración (solamente IRef1 y VRef1).

Se plantea una experiencia con una cinta transportadora, ya que existe una fuerte dependencia entre la velocidad del motor de corriente continua y el voltaje aplicado en sus terminales, sobre todo para el caso de imanes permanentes. El control de la velocidad se hace al variar el ciclo de trabajo, que como se explicó anteriormente, es controlar la cantidad de energía que se enviará a una carga (en este caso el motor DC).

La Figura 32 muestra la señal de corriente de realimentación obtenida aplicando la secuencia sugerida al motor 2 de la plataforma de trabajo.

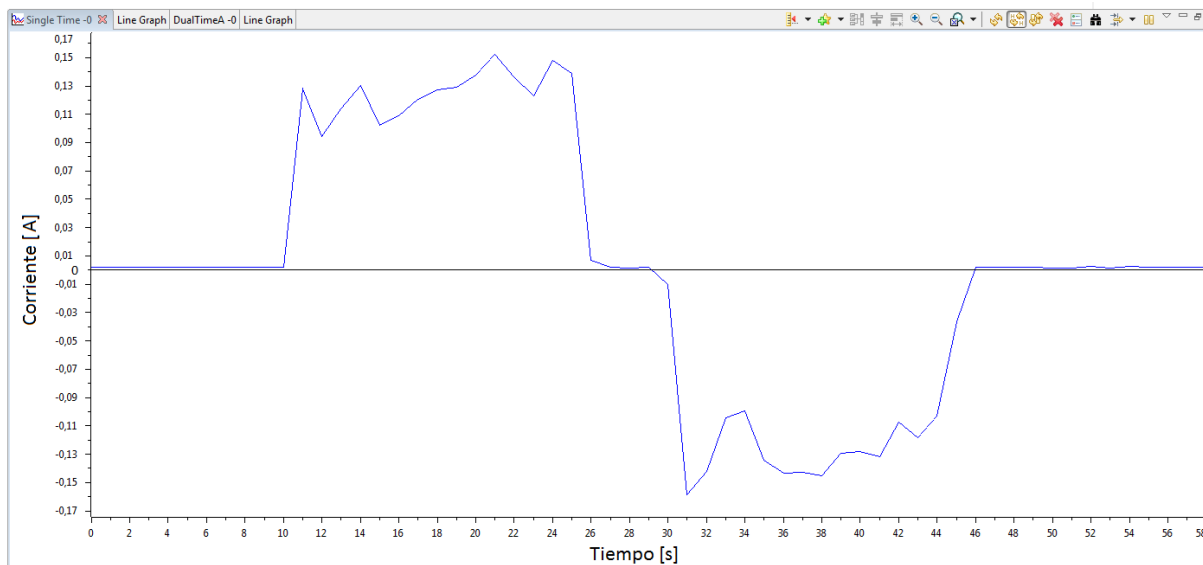


Figura 32: “Señal de corriente de realimentación obtenida con secuencia sugerida en el motor DC 2”
Fuente: Elaboración propia a través de Code Composer Studio.

La señal de la Figura 32 se obtiene conectando el motor DC 2 a la cinta transportadora y ejecutando el código con la secuencia dada en el “ANEXO D: CODIGO SECUENCIA CINTA TRANSPORTADORA”. Al observar el gráfico se logra determinar que la señal no es estable en su valor alto y bajo debido al roce existente en la cinta al momento de girar, esto produce un efecto de carga variable obteniendo como resultado un comportamiento variable de la corriente.

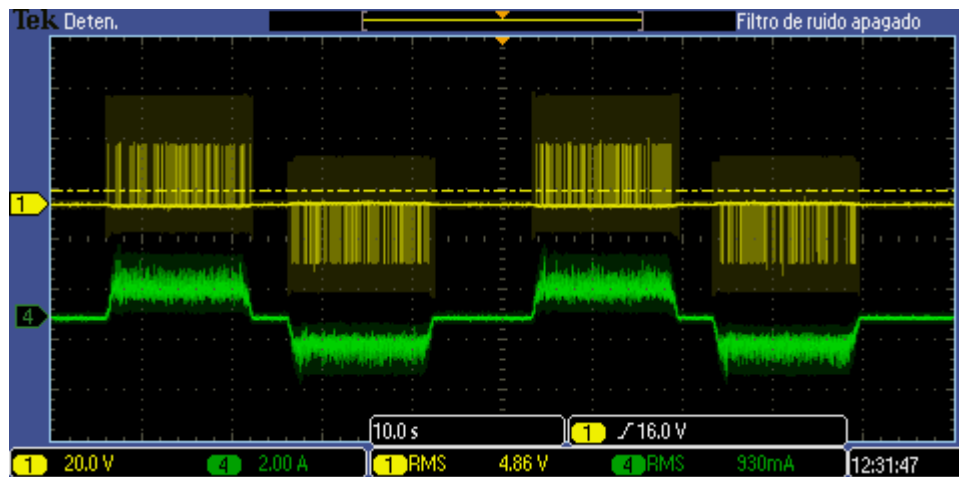


Figura 33: “Tensión y corriente medida en el motor al ejecutar secuencia sugerida en el motor DC 2”

Fuente: Medición Osciloscopio Tektronix MSO 2024B.

La Figura 33 muestra la señal de tensión (en amarillo) y corriente (en verde) al aplicar la secuencia sugerida en el motor DC 2, al igual que la Figura 32 la corriente se mantiene inestable en los niveles altos y bajos.

Cabe destacar que la Figura 32 y la Figura 33 se obtienen de la misma secuencia aplicada por lo que existe una directa relación entre estas.

En caso que se requiera aumentar un la velocidad que se le dará al motor, es necesario realizar una regla de tres simple con la cual es posible calcular el valor numérico con el que se debe reemplazar en la secuencia, este valor se puede aumentar al 8% o 9% lo cual sería suficiente. Para ello puede apoyarse en la siguiente relación:

$$\begin{array}{l} 7\% = 1174405.12 \\ 9\% = X \end{array}$$

Donde X será el valor a modificar en la secuencia para aumentar la tensión aplicada al motor DC y con ello aumentar la velocidad.

Se sugiere no sobrepasar el 9% de capacidad del motor DC al usarlo en conjunto con la maqueta, ya que este aumentaría demasiado su velocidad tomando en cuenta que una cinta transportadora no gira a gran velocidad.

La Figura 34 muestra dos señales de tensión y una señal de corriente al someter al motor DC 2 a una tensión constante:

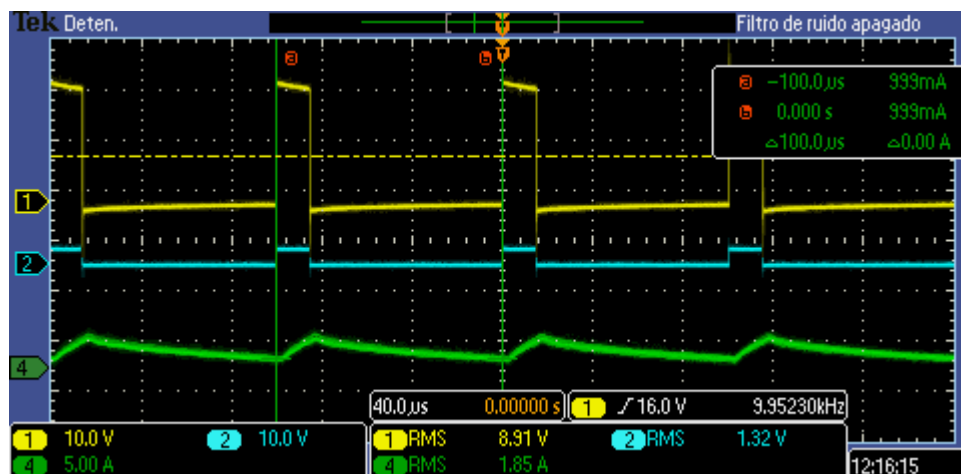


Figura 34: “Tensiones y corriente del motor DC 2 al aplicar una tensión de control constante”

Fuente: Medición Osciloscopio Tektronix MSO 2024B.

En la Figura 34 se muestra la tensión aplicada al motor DC 2 (señal 1, amarilla), la señal de control PWM_C antes de ingresar al puente H (señal 2, azul) y la señal de corriente en el motor DC 2 (señal 4, verde) se puede apreciar como aumenta la corriente mientras la señal de tensión esta en alto, y luego comienza a disminuir hasta llegar al mismo valor en el cual inició reflejando el efecto de carga y descarga de las bobinas.

La Figura 35 muestra el comportamiento de la corriente del motor DC 2 al mantenerlo con una tensión de referencia constante y aplicarle una carga más pesada:

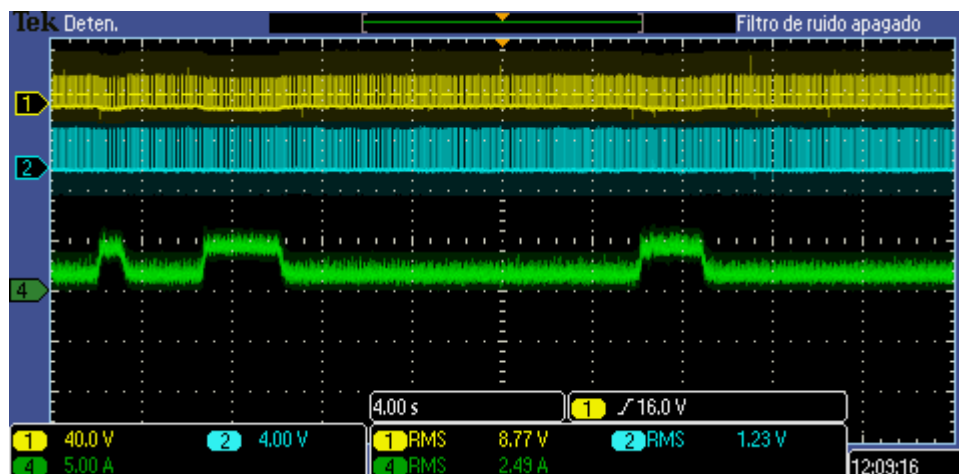


Figura 35: “Señales de tensiones y corriente al modificar la carga del motor DC 2”
Fuente: Medición Osciloscopio Tektronix MSO 2024B.

En la Figura 35 se muestra aplicada una tensión constante al motor (señal 1, amarilla), además se muestra la señal PWM_C antes de llegar al puente H (señal 2, azul) y la señal de corriente en el motor. En la figura se observa que al someter el motor a una carga más pesada aumenta la corriente para mantener el torque, luego al disminuir la carga la corriente disminuye pues el motor ya no la requiere. Cabe destacar que para lograr visualizar este efecto en el osciloscopio fue necesario aumentar la escala de tiempo, con lo que se logró visualizar el comportamiento de la corriente por un periodo de 40 segundos aproximadamente.

Al igual que el motor de pasos, al trabajar con este motor es posible variar en tiempo real desde el programa CCS la tensión o corriente según se requiera y dependiendo el nivel de compilación (ver capítulo 3).

Si bien es cierto la complejidad en tema de control es mucho menor en comparación con la experiencia propuesta para el motor de pasos, nos encontramos con una difícil tarea, la cual era disminuir el roce de giro en la cinta transportadora, lo que hacía que fuera difícil moverla. Debido a los escasos materiales necesarios para la construcción de la cinta, y al poco conocimiento de construcción técnica de estas, demoramos más tiempo del planificado en

llevarla a cabo, pero conseguimos reducir el roce de cierta manera agregando rodamientos a la maqueta, lo que facilitó el giro.

Además nos encontramos con otro problema, el cual era la conexión entre el eje del motor y el eje de la cinta transportadora, después de algún tiempo pensando en una solución, optamos por hacer la conexión mediante engranes, para lo cual recurrimos a la tecnología, e hicimos los datos técnicos requeridos para mandar a hacerlos con una impresora 3D. Al montar los engranes y poner en marcha la cinta observamos que producía ruido excesivo y los engranes no hacían un perfecto acople, por lo que finalmente optamos por hacer el acople mediante poleas, el cual resultó ser más sencillo y producía un mejor funcionamiento.

DISCUSIÓN

Se implementó el procedimiento que propone el fabricante para verificar el correcto funcionamiento de los distintos módulos del sistema (PWM, ADC, SINCOS, PID), además de verificar el funcionamiento del hardware.

Se trabaja con valores bajos de corriente y tensión, ya que al tener tres poleas y un contrapeso, el motor no debe hacer mucho esfuerzo para mover la carga, solo lo necesario para romper el equilibrio de la maqueta.

Para la maqueta de la cinta transportadora se realizó el control por tensión, ya que, al ser una carga liviana le genera menos esfuerzo al motor. Por el contrario, en el control por corriente necesariamente debe tener una carga en el eje del motor, si no, genera demasiado ruido en el motor, además de sobrecalentarlo, ya que tiende a la máxima velocidad.

CONCLUSIONES.

Sumario y Metodología.

En el presente seminario se hace un estudio del kit DRV8412-2-KIT basado en el microcontrolador C2000. Se abordan aspectos teóricos y prácticos en cuanto a su funcionamiento y configuración física; se describe el Driver de potencia y el DSP, con sus respectivas herramientas de hardware y software, además de la implementación práctica en tareas de control de movimiento en motores de pasos y de corriente directa.

La metodología que se utilizó para realizar este trabajo fue la siguiente: Primero fue necesario recopilar información del sistema, para luego desarrollar el documento base de funcionamiento de la maqueta y del software de control. Lo siguiente es comprender la estructura del software que se carga en el DSP, se determina en que sección se agrega la secuencia de control y cuáles serán las experiencias que se propondrán. Ya terminado lo anterior se procede a diseñar y construir las maquetas para experimentar el control de velocidad y posición con los dos tipos de motores. Se realizan las distintas pruebas y se desarrolla el informe del trabajo.

Conclusión.

Durante el proceso de prueba de las maquetas para las experiencias propuestas, se concluye de forma teórica y práctica que el comportamiento del motor se define por la naturaleza de la carga a la cual éste esté sometido, ya que se observó que al poner una carga liviana en la cinta transportadora esta desarrolla una alta velocidad, luego al cambiar la carga por una más pesada, ésta disminuye su velocidad aumentando el torque. Por otro lado la relación velocidad/torque no está presente en el motor de pasos. Como prueba de esto, en la experiencia del elevador se observó que al modificar la carga no varía el tiempo de ejecución de la secuencia, por lo que la velocidad de ascenso y descenso se mantiene.

Al hacer las mediciones se determinó que la velocidad del motor de pasos está limitada por la cantidad de micropasos que se le configure, ya que al usar una velocidad alta con una cantidad de micropasos elevada las bobinas no logran energizarse completamente, produciendo que el motor no se mueva y zumbe, o puede que gire pero de forma tosca e irregular.

Sugerencias a trabajos futuros.

Luego de finalizado el trabajo se proponen las mejoras que en el futuro se pueden realizar basándose en el mismo. A continuación se comentan brevemente las más importantes:

El control de velocidad y posición que se realizó fue en lazo abierto variando la tensión y la frecuencia, por lo que una mejora significativa sería utilizar un codificador óptico acoplado al eje del motor. Con esto se cierra el lazo de control y se puede controlar directamente la posición y la velocidad agregando la sección correspondiente en el código del programa.

Muchas de las funciones que tiene el DSP no se utilizan en el control de motor con este driver. Una mejora significativa sería construir una versión personalizada del circuito de la tarjeta de control dejando solo lo esencial para trabajar con el chip DRV8412. Por ejemplo, del ADC solo se usan 4 de las 16 entradas que este tiene, solo se utilizan 2 módulos PWM para controlar los dos puentes H, además no se utilizan todas las GPIO (entrada/salida de propósito general) de la tarjeta. Esto abriría un gran abanico de posibilidades para empezar a desarrollar circuitos de procesamiento digital de señales en la universidad para aplicaciones específicas.

REFERENCIAS.

- [1] PARRA, R. F. y REYES, E. F. 2014. Diseño y construcción de módulos de entrenamiento para control de motores DC. Seminario para optar al título de Ingeniero de Ejecución en Electrónica. Concepción-Chile, Universidad del Bío-Bío, Facultad de Ingeniería.
- [2] PÉREZ ROMERO, J. 2015. Diseño de un sistema empotrado basado en DSP C2000 de Texas Instruments para control de armario de potencia de 12 fases. Trabajo Fin de Grado Ingeniero de las Tecnologías Industriales. Sevilla-España, Universidad de Sevilla, Departamento de Ingeniería Electrónica.
- [3] GONZÁLEZ ORTEGA, ALFREDO. 2008. Estudio de Sistemas Digitales de Altas Prestaciones para Aplicaciones en Electrónica de Potencia. Tesis de Maestría en Ciencias. Cuernavaca, Morelos-México. Centro Nacional de Investigación y Desarrollo Tecnológico, Departamento de Ingeniería Electrónica.
- [4] TEXAS INSTRUMENTS. DRV8412-C2-KIT Hardware Reference Guide, No. TIDU314. Mayo 2014.
- [5] TEXAS INSTRUMENTS. DRV84x2 Dual Full-Bridge PWM Motor Driver, No. SLES242G. Diciembre 2009.
- [6] TEXAS INSTRUMENTS. TMS320x2833x Analog-to-Digital ConverterADC Module Reference Guide, No SPRU812A. Septiembre 2007.
- [7] TEXAS INSTRUMENTS. Dual-Axis Current Control of Brushed DC Motors Application Report, No. TIDA012. Mayo 2014.
- [8] TEXAS INSTRUMENTS. Dual-Axis Current Control of Brushed DC Motors Application Report, No. TIDA013. Mayo 2014.
- [9] TEXAS INSTRUMENTS. DRV8412-C2-KIT How to Run Guide, No. DRV8412-C2-KIT_HowToRunGuide. Noviembre 2010.
- [10] TEXAS INSTRUMENTS. DRV8412 Digital Motor Control Kit Quick Start Guide, No. QSG-DRV8412-C2-KIT-GUI. Mayo 2011.

- [11] CASA FACIL. 2009. Motor Eléctrico [en línea] Argentina. <http://www.casafacil.com.ar/electricidad/motor_electrico.php> [consulta: 14 marzo 2017]
- [12] ORIENTAL MOTOR. 2016. Basics of Motion Control [en línea] Estados Unidos. <<http://www.orientalmotor.com/technology/articles/2phase-v-5phase.html>> [consulta: 18 marzo 2017]
- [13] EDUARDO JULIO CARLETTI. 2007. Motores paso a paso Características básicas [en línea] Buenos Aires, Argentina. <http://robots-argentina.com.ar/MotorPP_basico.htm> [consulta: 14 marzo 2017]
- [14] WIKIPEDIA. 2016. Puente H (electrónica) [en línea] Estados Unidos. <[https://es.wikipedia.org/wiki/Puente_H_\(electr%C3%B3nica\)](https://es.wikipedia.org/wiki/Puente_H_(electr%C3%B3nica))> [consulta: 18 marzo 2017]

ANEXO A: MANUAL DE PUESTA EN MARCHA DEL DRV8412-C2-KIT

A continuación se entrega una ayuda básica para operar el kit C2000, en donde se explica la configuración de hardware y software, además de como cargar y ejecutar un programa. Si se desea tener una explicación detallada de todas las funciones del software CCS o de la programación en lenguaje “C”, puede informarse por fuentes externas.

Instrucciones de Configuración de Hardware

Para los proyectos de “Control de Motor”, se usa como alimentación la fuente de 24V DC incluida en el kit. La configuración de las dos placas incluidas en el kit se muestra en los siguientes pasos:

1. Instalar en la DRV8412-EVM (Figura 37), el Jumper de JP1 en la posición VR1 para configurar la fuente interna de 12V.
2. Verificar que los switch de la Tarjeta de Control se encuentren en la configuración que muestra la Figura 36

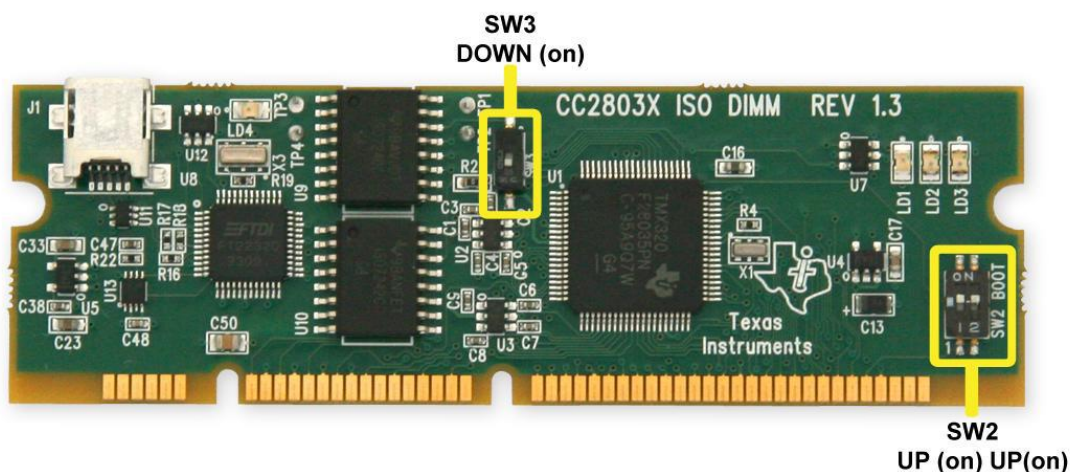


Figura 36: “Configuración de Switch tarjeta de control”

3. Insertar la Tarjeta de Control en la ranura J1 de la DRV8412-EVM (Figura 37). Empujar verticalmente hacia abajo usando presión en ambos bordes de la tarjeta hasta que los clips se cierren. (para retirar la tarjeta simplemente se abren los clips de retención).
4. Conectar el cable USB al J1 en la Tarjeta de Control (Figura 36). Esto activará la emulación aislada de JTAG en el dispositivo C2000. LD4 en la Tarjeta de Control se encenderá.

5. Conectar la fuente de 24 (V) a los terminales PVDD y GND de la DRV8412-EVM (Figura 37). Ahora LD2 y LD3, en la placa base DRV8412, se encenderán. LD1 en la Tarjeta de Control se encenderá para indicar que la Tarjeta de Control está recibiendo energía de la placa.

Como referencia, la Figura 37 y la Figura 38 muestran las conexiones de motores y la configuración de jumper para las distintas experiencias.

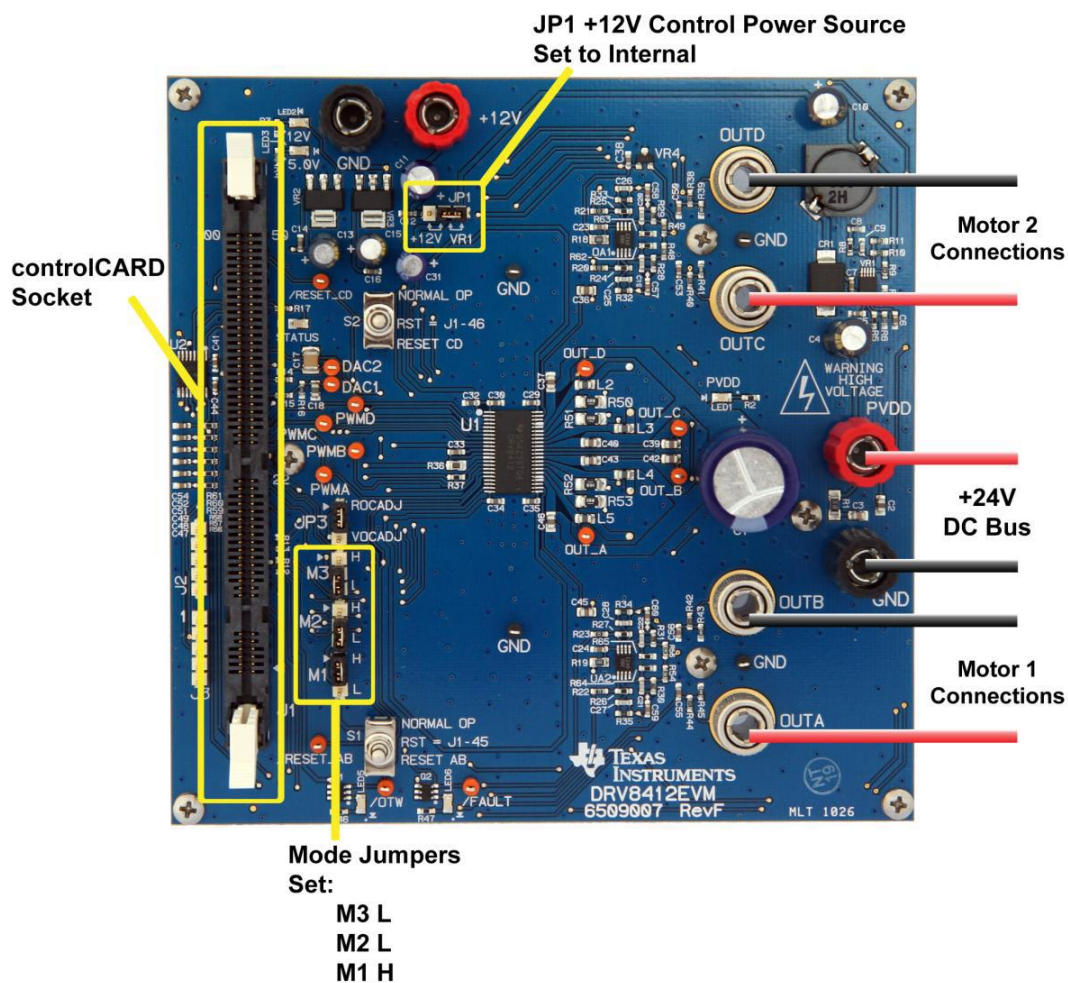


Figura 37: “Configuración de Hardware para experiencia 2xDC_Motor”

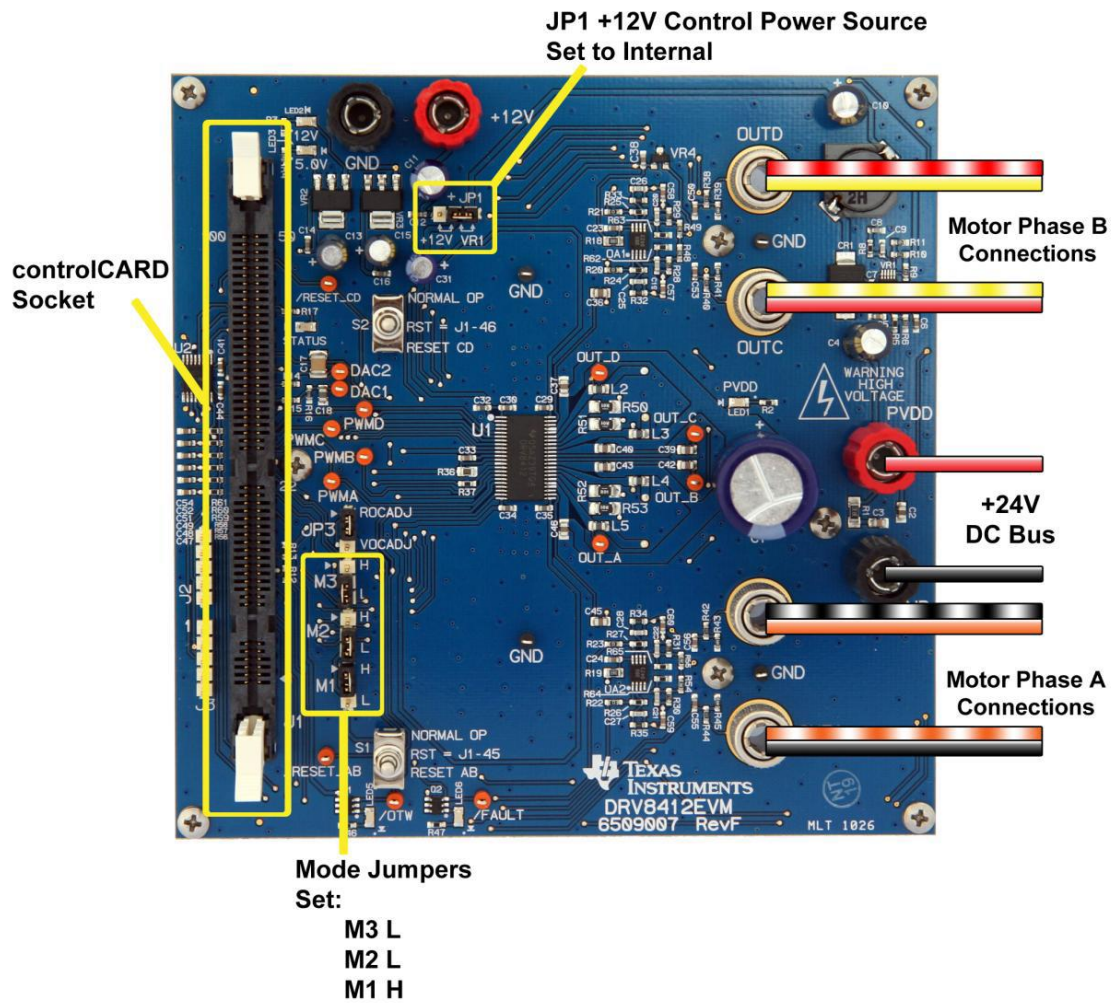


Figura 38: “Configuración de Hardware para experiencia Stepper”

Configuración de software para los proyectos de DRV8412-EVM

Instalando Code Composer y controlSuite

1. Si aún no está instalado, instalar Code Composer v4 desde el CD incluido con el kit.
2. Desde el CD ejecutar controlsuiteinstaller.exe. Es recomendable hacer una instalación típica (todo el contenido).

Configurar Code Composer Studio para trabajar con la DRV8412-EVM

3. Abrir “Code Composer Studio v4”.
 4. Una vez abierto, aparecerá el “workspace launcher” que pedirá seleccionar la ubicación del proyecto. (la ubicación del workspace, es donde se guardara en el disco duro las configuraciones de usuario, de proyectos y archivos guardados).
- Clic en el botón “Browse...”
 - Crear la ruta a continuación si es necesario hacer una nueva carpeta.
 - “C:\Documents and Settings\My Documents\CCSv4_workspaces\DRV8412-EVM_workspace”
 - Desmarcar la casilla que dice “Use this as the default and do not ask again”.
 - Click en “OK”.

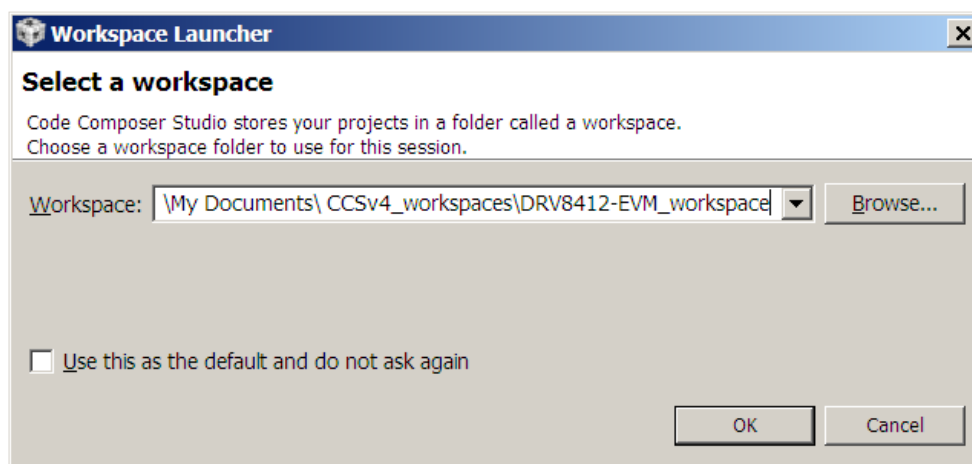


Figura 39: “Workspace Launcher”

5. Lo siguiente es configurar Code Composer para saber cual MCU esta conectada. Click en “Target -> New Target Configuration...”. Nombrar la nueva configuración xds100-f28035.ccxml o cualquier otro nombre dependiendo de la tarjeta. Clic en Finish.

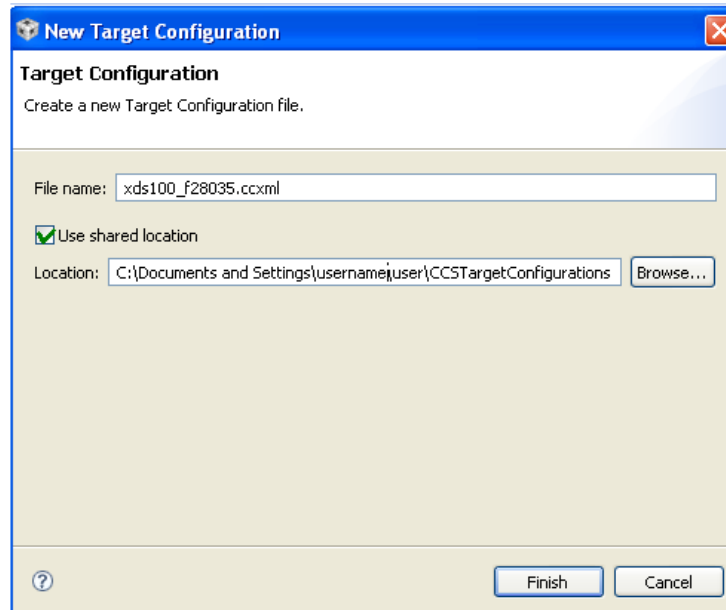


Figura 40: “Creando una configuración de tarjeta”

6. Esto abre una nueva pestaña como se ve en la Figura 41. Seleccionar e ingresar las opciones que se muestran:

- Connection – Texas Instruments XDS100v1 USB Emulator
- Device – TMS320F28035
- Click en Save
- Cerrar la pestaña xds100-f28035.ccxml

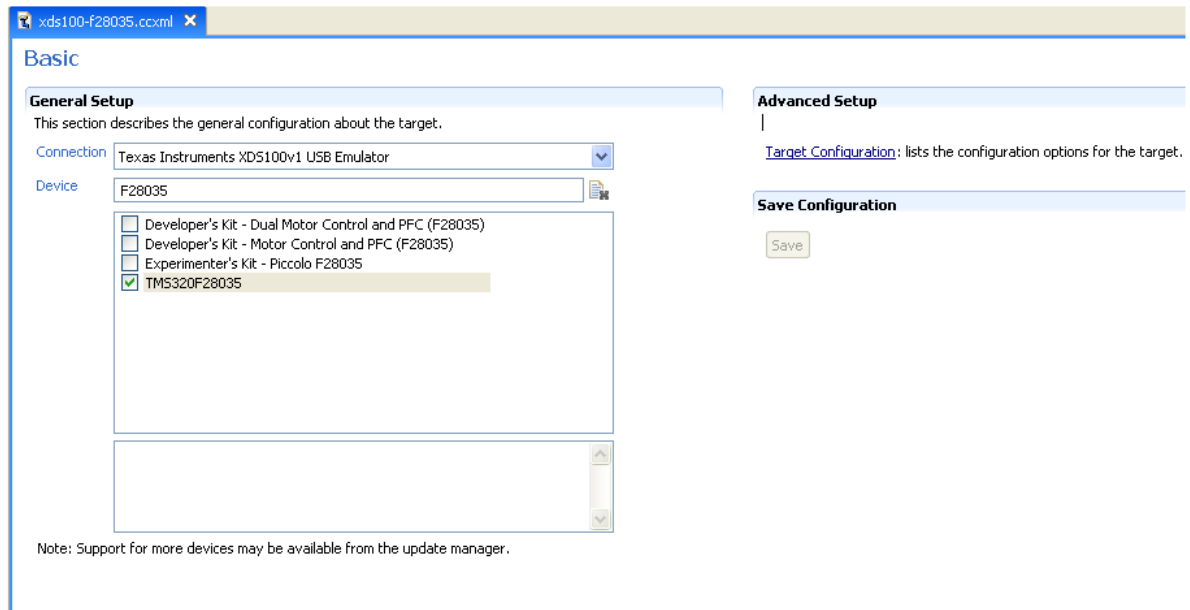


Figura 41: “Configurando una nueva tarjeta”

7. Si es la primera vez que se usa Code Composer, la configuración se establece como configuración por defecto.

8. Agregar todos los proyectos de control de motor al actual “workspace” haciendo clic en “Project->Import Existing CCS/CCE Eclipse Project”.
- Seleccionar la ubicación del DRV8412-EVM (Figura 42). Esta es:
“C:\ti\controlSUITE\development_kits\DRV8412-EVM”

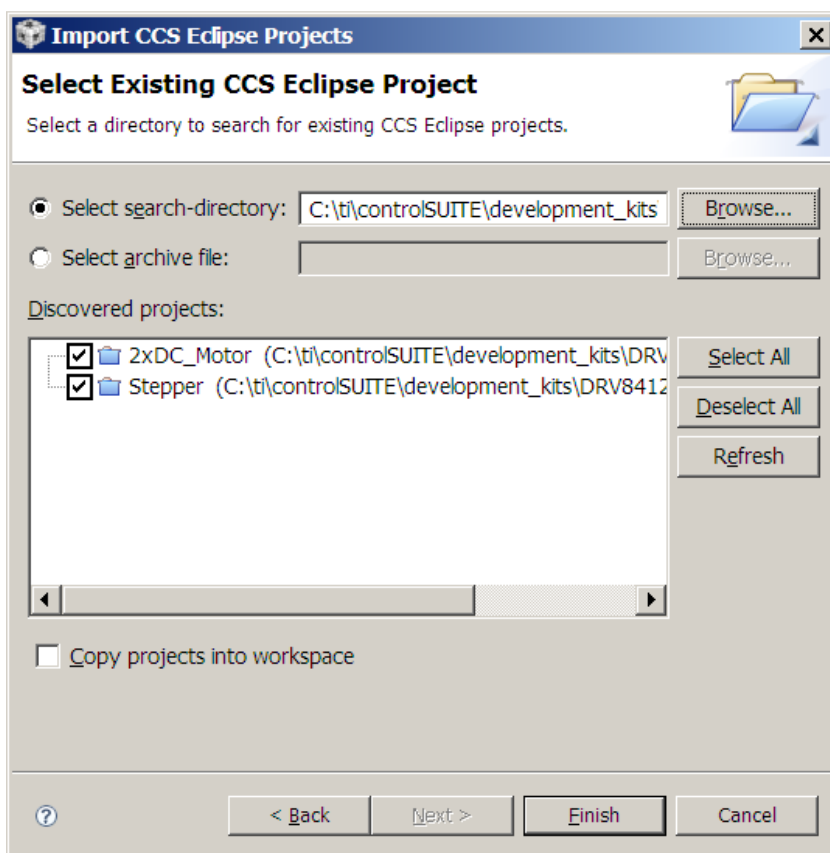


Figura 42: “Agregando todos los proyectos de DRV8412-EVM al workspace”

- Click en Finish, esto copiará todos los proyectos relevantes para el kit en el workspace. Si solo se necesita un proyecto en particular, desmarcar las casillas de los otros proyectos.

➤ **Problema Frecuente.**

Si se tiene dificultad para comunicar la F28035, es posible que el serial-ti-USB no este habilitado por defecto. Favor de seguir los siguientes pasos en orden para habilitar el puerto serial virtual.

- Conectar la F28035 a un puerto USB disponible en la computadora.
- Abrir el Administrador de Dispositivos de Windows y expandir la sección de Controladores de Bus Serie Universal.
- Para cada entrada de TI XDS100, abrir el dialogo *Propiedades* y seleccionar la pestaña *Opciones Avanzadas* (Figura 43). Click para habilitar “Load VCP” como se muestra a continuación.

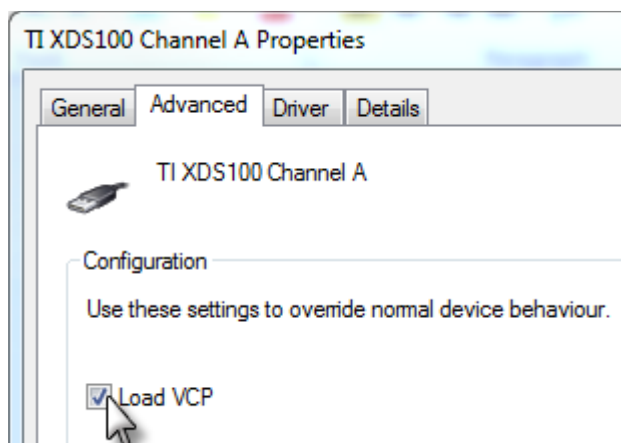


Figura 43: “Pestaña avanzada de TI XDS100”

- Cerrar el Administrador de Dispositivos de Windows y desconectar la F28035.
- Reconectar la F28035, es posible que se solicite instalar los controladores, ubicados en la carpeta “Drivers”.
- Para verificar si la instalación esta correcta, abrir el Administrador de Dispositivos de Windows y verificar que dos nuevos puertos serie se muestren bajo la sección Puertos (COM & LPT).

Configurando un Proyecto

1. Expandir la estructura de archivos del proyecto que se desea ejecutar, desde la pestaña C/C++ Projects. Dar Clic derecho en el nombre del proyecto y seleccionar “Set as Active Project”.
2. Asumiendo que es la primera vez que se usa Code Composer, *xds 100-F28035* será establecido como la configuración por defecto de la tarjeta. Para verificar esto, observar que en la estructura expandida del proyecto el archivo *xds100-f28035.ccxml* tiene un [Active/Default] escrito al lado. Al ir a “View->Target Configuration” se pueden editar las configuraciones existentes, cambiar la por defecto o activar una configuración.
3. Cada proyecto puede ser configurado para generar código y ser ejecutado en flash o en RAM. Se puede seleccionar cualquiera de los dos, sin embargo, para las experiencias de laboratorio se usará la configuración RAM la mayoría de las veces (se selecciona la configuración FLASH para producción). Como se muestra en la Figura 44, hacer clic derecho en un proyecto individual y elegir “Active Build Configuration -> F28035x_RAM”.

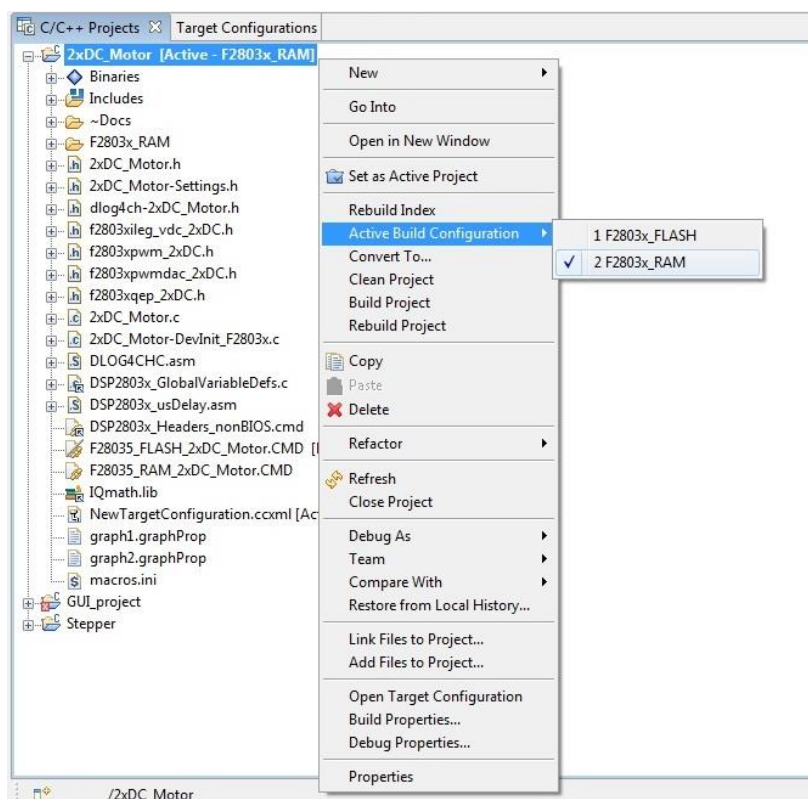





Figura 44: “Elijiendo la configuración F2803x_RAM”

Compilar y Cargar el Proyecto

1. Los programas de Control de Motor de Texas Instruments proporcionan compilaciones incrementales, donde diferentes componentes / bloques macro del sistema se unen uno a uno para formar el sistema completo. Esto ayuda en la depuración paso a paso y la comprensión del sistema. Desde la pestaña C/C++ Project abrir el archivo: *[Nombre-del-proyecto]-Settings.h* y asegurarse que el BUILDLEVEL este establecido en el nivel que se requiera trabajar y guardar el archivo. Las explicaciones en detalle de los niveles de compilación (Build Level) para cada proyecto, están en el *Capítulo 3*.
2. Ahora, click derecho en el Nombre del Proyecto y hacer clic en “Rebuild Project” y observar la ventana “Console”. Cualquier error en el proyecto será mostrado en la ventana “Console”.
3. Cuando se complete con éxito la compilación, hacer click en el botón  “Debug”, ubicado en la parte superior izquierda de la pantalla. La IDE automáticamente conectará a la tarjeta, cargará el archivo en el dispositivo y cambiará la vista de depuración.
4. Hacer click en el botón “Enable silicon real-time mode”  y en el botón “Enable polite real-time mode” . Esto permitirá al usuario editar y ver las variables en tiempo real. *¡No hacer RESET a la CPU sin desactivar las opciones de tiempo real!*

➤ Problema frecuente.

Suele suceder, que luego de compilar (rebuild), al hacer debug, se muestra una falla en pantalla, y además, se encenderá el LED FAULT en la placa base. Esto se debe a que no está conectado el motor adecuado para el proyecto en curso. Para solucionar esto, comprobar que los switch del gabinete se encuentren en la posición *DC* (hacia la izquierda), o *Stepper* (hacia la derecha), según se requiera.

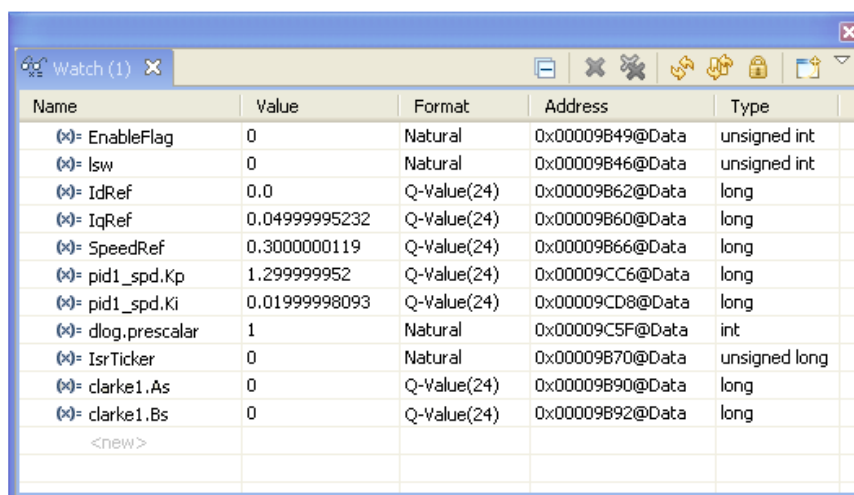
Si el LED FAULT sigue encendido, mover los switch S1 y S2 de la placa base a la posición superior (NORMAL OP) y luego a la posición media (RST), con esto se quita el estado de falla de la placa base.

Si el problema persiste, comprobar que el proyecto actual se encuentra *activo*.

Configurar la Ventana Watch.


Click en: View -> Watch en la barra del menú para abrir la *ventana Watch* y ver las variables que están siendo usadas en el proyecto. A continuación se muestra como agregar variables a la ventana Watch. Haciendo click derecho en la variable es posible cambiar el formato de número de la variable. También se puede seleccionar el “Q Format” apropiado para las variables que se deseen observar.

Más adelante, se mostrará que variables son necesarias agregar para cada proyecto. A continuación la Figura 45 muestra una ventana Watch típica.






Name	Value	Format	Address	Type
EnableFlag	0	Natural	0x00009B49@Data	unsigned int
lsw	0	Natural	0x00009B46@Data	unsigned int
IdRef	0.0	Q-Value(24)	0x00009B62@Data	long
IqRef	0.04999995232	Q-Value(24)	0x00009B60@Data	long
SpeedRef	0.3000000119	Q-Value(24)	0x00009B66@Data	long
pid1_spd.Kp	1.299999952	Q-Value(24)	0x00009CC6@Data	long
pid1_spd.Ki	0.01999998093	Q-Value(24)	0x00009CD8@Data	long
dlog.prescalar	1	Natural	0x00009C5F@Data	int
IsrTicker	0	Natural	0x00009B70@Data	unsigned long
clarke1.As	0	Q-Value(24)	0x00009B90@Data	long
clarke1.Bs	0	Q-Value(24)	0x00009B92@Data	long
<new>				

Figura 45: “Configurando la ventana Watch”

Hacer click en el botón “Continuous Refresh”  en la ventana Watch. Esto permite que la ventana se ejecute en el modo tiempo real. Haciendo click en la flecha hacia abajo de la misma ventana y seleccionando “Customize Continuous Refresh Interval” se puede editar la frecuencia de actualización de la ventana Watch. Nota: Elegir un intervalo muy rápido puede afectar el rendimiento.

Ejecutar el Código.

1. Se ejecuta el código al pulsar el botón *Run*  en la pestaña Debug.
2. El proyecto se ejecutará y los valores de la ventana Watch se mantendrán actualizados.
3. Una vez completado, hacer RESET al procesador  (Target->Reset->Reset CPU) y luego finalizar la sesión de depuración haciendo click en  (Target->Terminate All). Esto detendrá el programa y desconectará Code Composer de la MCU.

ANEXO B: GUIA DE LABORATORIO

EXPERIENCIAS DE CONTROL
Experiencia N°1: Simulación de planta con motor de pasos.

OBJETIVO:

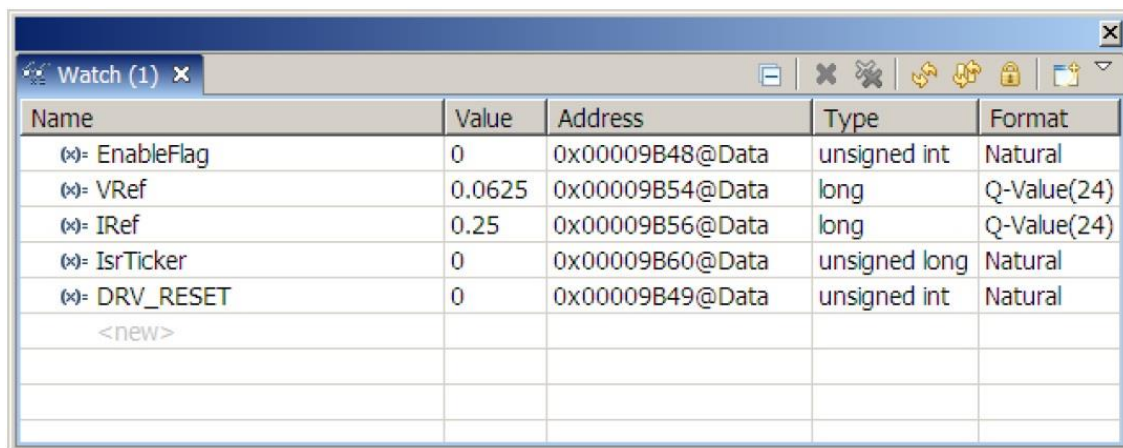
- Experimentar control punto a punto a través de la herramienta Code Composer Studio aplicado a la maqueta del elevador.

Procedimiento:

En esta experiencia se trabaja con el motor de pasos, por lo que antes de empezar, comprobar que los switch del gabinete se encuentren todos en la *posición Stepper* (a la derecha), para no tener problemas en la ejecución del programa.

Guiarse por los pasos genéricos de la guía básica de operación para configurar el hardware.

1. Seleccionar Stepper como proyecto activo.
2. Abrir “*Stepper-Settings.h*”
3. Seleccionar el nivel de compilación 3, para eso, editar el BUILDLEVEL a LEVEL3 (`#define BUILDLEVEL LEVEL3`).
4. Dar click derecho en el nombre del proyecto y hacer click en “*Rebuild Project*”.
5. Una vez que la compilación este completa, dar click en el botón *Debug* para cargar el código en el controlador.
6. Abrir una ventana Watch y agregar las variables que se muestran en la Figura 46 y seleccionar el formato apropiado para cada una.



Name	Value	Address	Type	Format
(x) EnableFlag	0	0x00009B48@Data	unsigned int	Natural
(x) VRef	0.0625	0x00009B54@Data	long	Q-Value(24)
(x) IRef	0.25	0x00009B56@Data	long	Q-Value(24)
(x) IsrTicker	0	0x00009B60@Data	unsigned long	Natural
(x) DRV_RESET	0	0x00009B49@Data	unsigned int	Natural
<new>				

Figura 46: “Variables de la ventana Watch”

7. Hacer Reset a la CPU y Restart.
8. Activar el modo tiempo-real y ejecutar (Run).
9. Dar el valor 1 a “EnableFlag” en la ventana Watch. La variable llamada “IsrTicker” empezará a aumentar, confirmando así que la interrupción funciona apropiadamente.
10. En el software, las variables importantes a ajustar son las siguientes:
 - IRef (Q24): Para cambiar la corriente en el motor en por-unidad.
 - SpeedRef (Q24): Para cambiar la velocidad eléctrica del motor.

En este nivel, la corriente se regula dinámicamente al usar el módulo PID_REG3.

Como prueba, fijar IRef en 0.25 y SpeedRef en 0.25. Dar otros valores para confirmar que el módulo PID funciona apropiadamente.

Adicionalmente, se puede modificar el valor de MICROSTEPS en el archivo *Stepper-Settings.h*. Se puede dar el valor de: Paso completo, medio paso, cuarto de paso o micro-pasos (de 8 a 128). Compilar el proyecto para que los cambios tengan efecto.

Luego de esta primera prueba, se procederá a utilizar la maqueta de elevación, para eso, asegurarse que el “Modo en Tiempo Real” este INACTIVO y que la sesión Debug este en STOP.

- A continuación, acoplar la maqueta al eje de motor de pasos y fijar el rodillo al eje con el tornillo.
- Alinear y atornillar la maqueta al gabinete con los pernos de fijación.

- Una vez hecho lo anterior, girar el eje en sentido horario hasta que la cuerda quede tensa y el carro permanezca en la base. Al cumplir con lo anterior, se realizará la edición del programa.
- Para que la maqueta funcione adecuadamente, se editará el programa y se agregará una secuencia de ascenso y descenso pausada. Esta secuencia se encuentra en el ANEXO C: CÓDIGO SECUENCIA DE ASCENSO Y DESCENSO, y se debe agregar entre las siguientes líneas de código del archivo “Stepper.c” desde el CCS, como se muestra en la Figura 47.

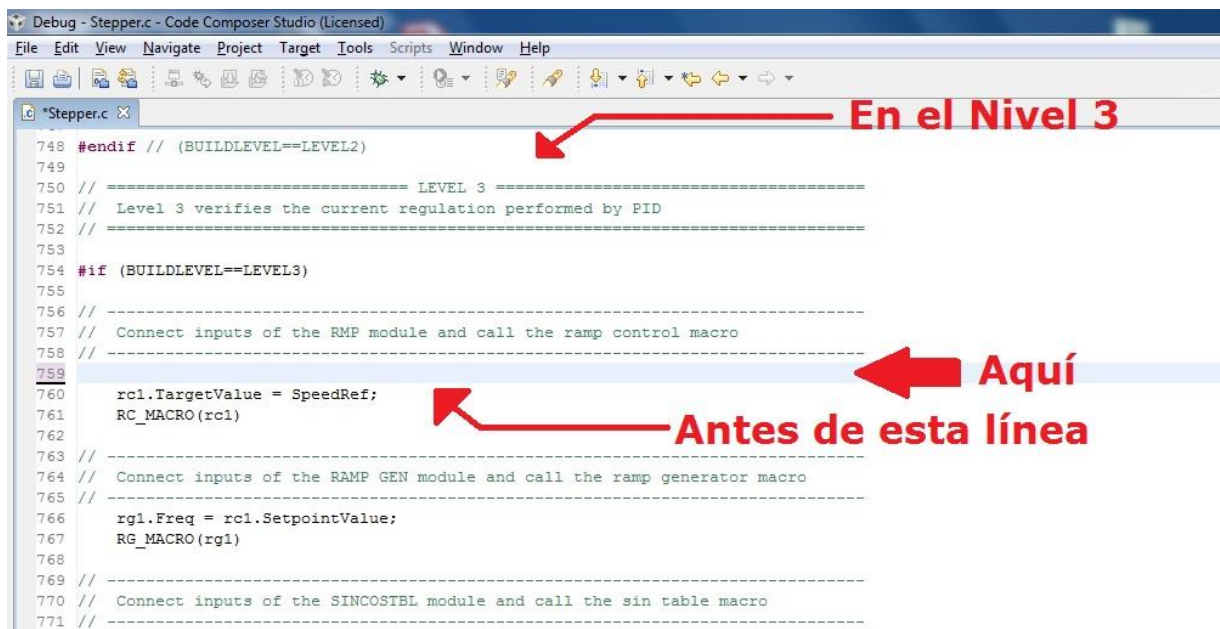


Figura 47: “Indicación para editar programa”

- Ya copiado y pegado el código de secuencia, se da en *Rebuild* en el proyecto activo, y luego en *Debug*. Al hacer esto se compilará y cargará el programa.
- Lo siguiente es activar el modo tiempo real, y dar en RUN.
- Para que inicie el movimiento, se cambiará el valor de las variables DRV_RESET a cero (0) y de *EnableFlag* a uno (1). Si todo esta bien, se observará que empieza a moverse el carro y se detendrá. El movimiento es en cuatro “pisos” y de uno en uno.

Experiencia N°2: Simulación de planta con motor DC

OBJETIVO:

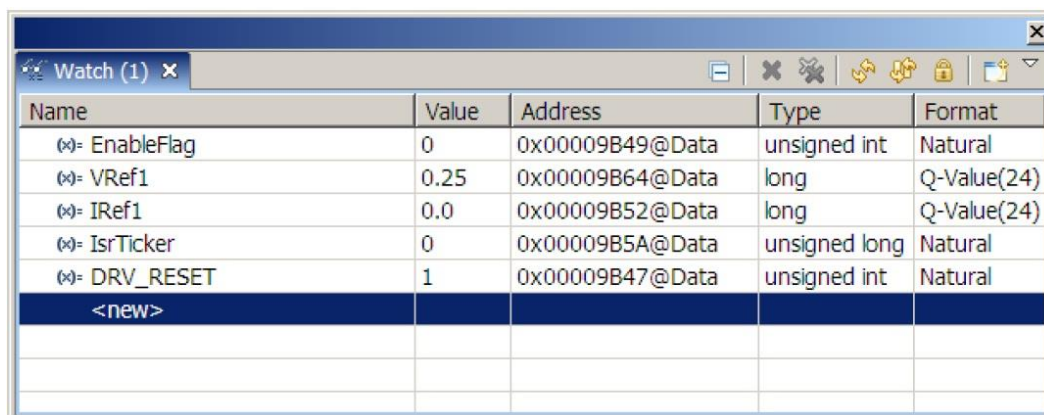
- Experimentar control punto a punto a través de la herramienta Code Composer Studio aplicado a la maqueta “cinta transportadora”.

Procedimiento:

Antes de empezar, comprobar que los switch del gabinete se encuentran todos en la posición DC (a la izquierda), para no tener problemas en la ejecución del programa.

En esta experiencia se hará una prueba en lazo abierto de control por tensión de un motor DC. Para realizar la prueba de control por tensión se desconecta la maqueta en caso de estar conectada. Seguir los pasos de la guía anterior para ejecutar el proyecto.

1. Seleccionar 2xDC_Motor como “active Project”.
2. Abrir 2xDC_Motor-Settings.h
3. Seleccionar la opción nivel 2 de compilación, para eso, editar el BUILDLEVEL a LEVEL2 (#define BUILDLEVEL LEVEL2).
4. Guardar el archivo.
5. Dar click derecho en el nombre del proyecto y hacer click en Rebuild Project.
6. Una vez que la compilación este completa, dar click en el botón Debug para cargar el código en el controlador.
7. Abrir una ventana Watch y agregar las variables que se muestran en la Figura 48 y elegir el Formato apropiado.



Name	Value	Address	Type	Format
EnableFlag	0	0x00009B49@Data	unsigned int	Natural
VRef1	0.25	0x00009B64@Data	long	Q-Value(24)
IRef1	0.0	0x00009B52@Data	long	Q-Value(24)
IsrTicker	0	0x00009B5A@Data	unsigned long	Natural
DRV_RESET	1	0x00009B47@Data	unsigned int	Natural
<new>				

Figura 48: “Variables de la ventana Watch (Brushed DC)”

8. Hacer Reset en la CPU y Restart.
9. Activar modo real-time y Run.
10. Dar el valor 1 a “Enable Flag” y 0 a “DRV_RESET” en la ventana Watch. La variable “IsrTicker” se incrementará automáticamente, esto confirma que la interrupción del sistema funciona correctamente.
11. Desde la misma ventana, fijar VRef1 en 0.2. El motor empezará a girar lentamente, si se desea se puede ajustar la velocidad del motor, solo se debe modificar el valor en la variable VRef1 con un rango de valores entre 0.0 y 1 los cuales equivalen a 0 y 100%, se debe tener en cuenta que al poner el valor de 0.02 el motor no se moverá, ya que el valor en tensión es muy pequeño.

Para dar una detención segura del sistema, fijar la variable DRV_RESET en 1, quitar el modo tiempo real del controlador (desactivarlo), y hacer un reset a el procesador.

Luego de finalizada la prueba, se procede a utilizar la maqueta de cinta transportadora, para eso, asegúrese que el “Modo en Tiempo Real” este INACTIVO y que la sesión Debug este en STOP.

- A continuación, acoplar la maqueta de la cinta transportadora al gabinete mediante la escuadra, ubicar la correa en las poleas de la cinta y del motor DC 2.
- Para lograr el movimiento punto a punto en la cinta es necesario agregar una secuencia al programa. Esta secuencia se encuentra en el ANEXO D: CODIGO SECUENCIA CINTA

TRANSPORTADORAANEXO D: CODIGO SECUENCIA CINTA TRANSPORTADORA, y se debe agregar entre las siguientes líneas del archivo “2xDC_Motor.c” desde el CCS, como se muestra en la figura:

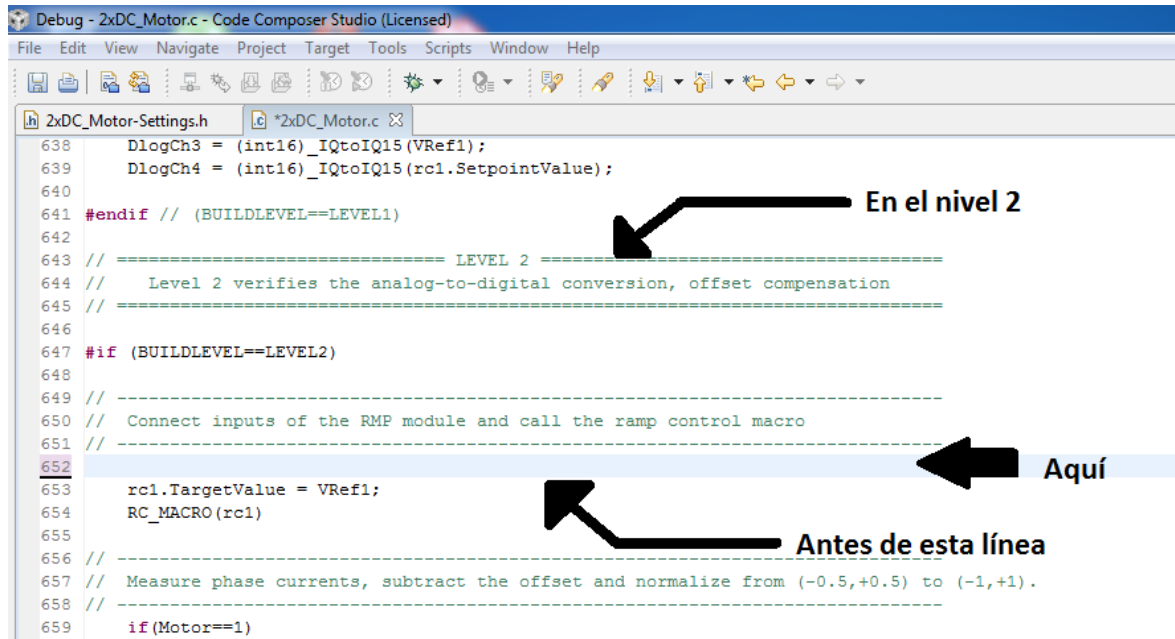


Figura 49: “Indicación para editar programa”

- Ya copiado y pegado el código de secuencia, se da Rebuild en el proyecto activo, y luego en Debug. Al hacer esto se compilará y cargará el programa.
- Lo siguiente es activar el modo tiempo real, y dar en RUN.
- Para que inicie el movimiento, se cambiará el valor de las variables DRV_RESET a cero (0) y de EnableFlag a uno (1). Si todo está bien, se observara que la cinta parte de forma acelerada para luego mantenerse a una velocidad baja, luego esta se detendrá, para proceder a girar en el sentido opuesto.

ANEXO C: CÓDIGO SECUENCIA DE ASCENSO Y DESCENSO

```

if (IsrTicker>0 && IsrTicker<28650){           // primer a segundo nivel
    SpeedRef=3355443.2;
}
if (IsrTicker>28650 && IsrTicker<80000){
    SpeedRef=0;
}
if (IsrTicker>80000 && IsrTicker<108650){       // segundo a tercer nivel
    SpeedRef=3355443.2;
}
if (IsrTicker>108650 && IsrTicker<160000){
    SpeedRef=0;
}
if (IsrTicker>160000 && IsrTicker<188650){     // tercer a cuarto nivel
    SpeedRef=3355443.2;
}
if (IsrTicker>188650 && IsrTicker<250000){
    SpeedRef=0;
}
if (IsrTicker>250000 && IsrTicker<278650){     //Baja del cuarto al tercero
    SpeedRef=-3355443.2;
}
if (IsrTicker>278650 && IsrTicker<330000){
    SpeedRef=0;
}
if (IsrTicker>330000 && IsrTicker<358650){     //Baja del tercero al segundo
    SpeedRef=-3355443.2;
}
if (IsrTicker>358650 && IsrTicker<410000){
    SpeedRef=0;
}
if (IsrTicker>410000 && IsrTicker<438650){     //Baja del segundo al primero
    SpeedRef=-3355443.2;
}
if (IsrTicker>438650 && IsrTicker<490000){
    SpeedRef=0;
}
// Fin Secuencia
    
```

ANEXO D: CODIGO SECUENCIA CINTA TRANSPORTADORA

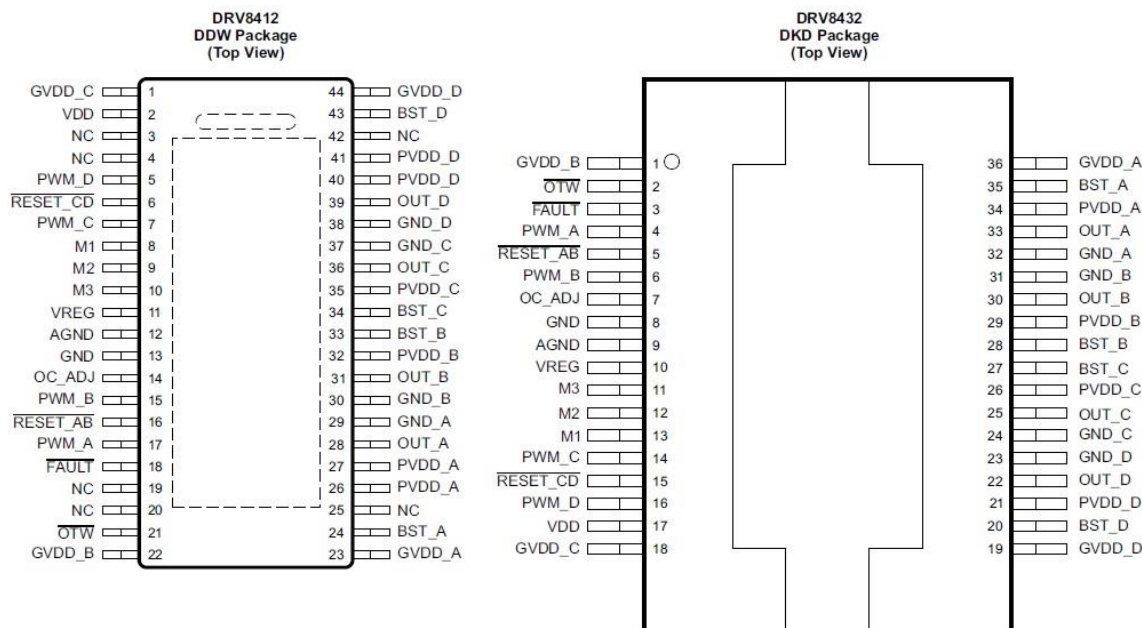
```

if (IsrTicker<15000 && IsrTicker>0){           //Inicia movimiento al 15%
VRef1 = 2516582.4;
}
if (IsrTicker<150000 && IsrTicker>15000){       //Mantiene el movimiento al 7%
VRef1 =1174405.12;
}
if (IsrTicker<200000 && IsrTicker>150000){      //Movimiento en 0%
VRef1 = 0000000;
}
if (IsrTicker<215000 && IsrTicker>200000){      //inicia movimiento al 15% sentido opuesto
VRef1 = -2516582.4;
}
if (IsrTicker<350000 && IsrTicker>215000){      //Mantiene el movimiento al 7%
VRef1 = -1174405.12;
}
if (IsrTicker>350000){                          //Se detiene
VRef1 = 0000000;
}
rc1.TargetValue = VRef1;
    RC_MACRO(rc1)

```

ANEXO E: DATOS TÉCNICOS (DRV8412)

Pin Configuration and Functions



Pin Functions

NAME	PIN		I/O TYPE ⁽¹⁾	DESCRIPTION
	DRV8412	DRV8432		
AGND	12	9	P	Analog ground
BST_A	24	35	P	High side bootstrap supply (BST), external capacitor to OUT_A required
BST_B	33	28	P	High side bootstrap supply (BST), external capacitor to OUT_B required
BST_C	34	27	P	High side bootstrap supply (BST), external capacitor to OUT_C required
BST_D	43	20	P	High side bootstrap supply (BST), external capacitor to OUT_D required
GND	13	8	P	Ground
GND_A	29	32	P	Power ground for half-bridge A
GND_B	30	31	P	Power ground for half-bridge B
GND_C	37	24	P	Power ground for half-bridge C
GND_D	38	23	P	Power ground for half-bridge D
GVDD_A	23	36	P	Gate-drive voltage supply
GVDD_B	22	1	P	Gate-drive voltage supply
GVDD_C	1	18	P	Gate-drive voltage supply
GVDD_D	44	19	P	Gate-drive voltage supply
M1	8	13	I	Mode selection pin
M2	9	12	I	Mode selection pin
M3	10	11	I	Reserved mode selection pin, AGND connection is recommended
NC	3, 4, 19, 20, 25, 42	—	—	No connection pin. Ground connection is recommended
OC_ADJ	14	7	O	Analog overcurrent programming pin, requires resistor to AGND
OTW	21	2	O	Overtemperature warning signal, open-drain, active-low. An internal pullup resistor to VREG (3.3 V) is provided on output. Level compliance for 5-V logic can be obtained by adding external pullup resistor to 5 V
OUT_A	28	33	O	Output, half-bridge A
OUT_B	31	30	O	Output, half-bridge B
OUT_C	36	25	O	Output, half-bridge C

(1) I = input, O = output, P = power, T = thermal

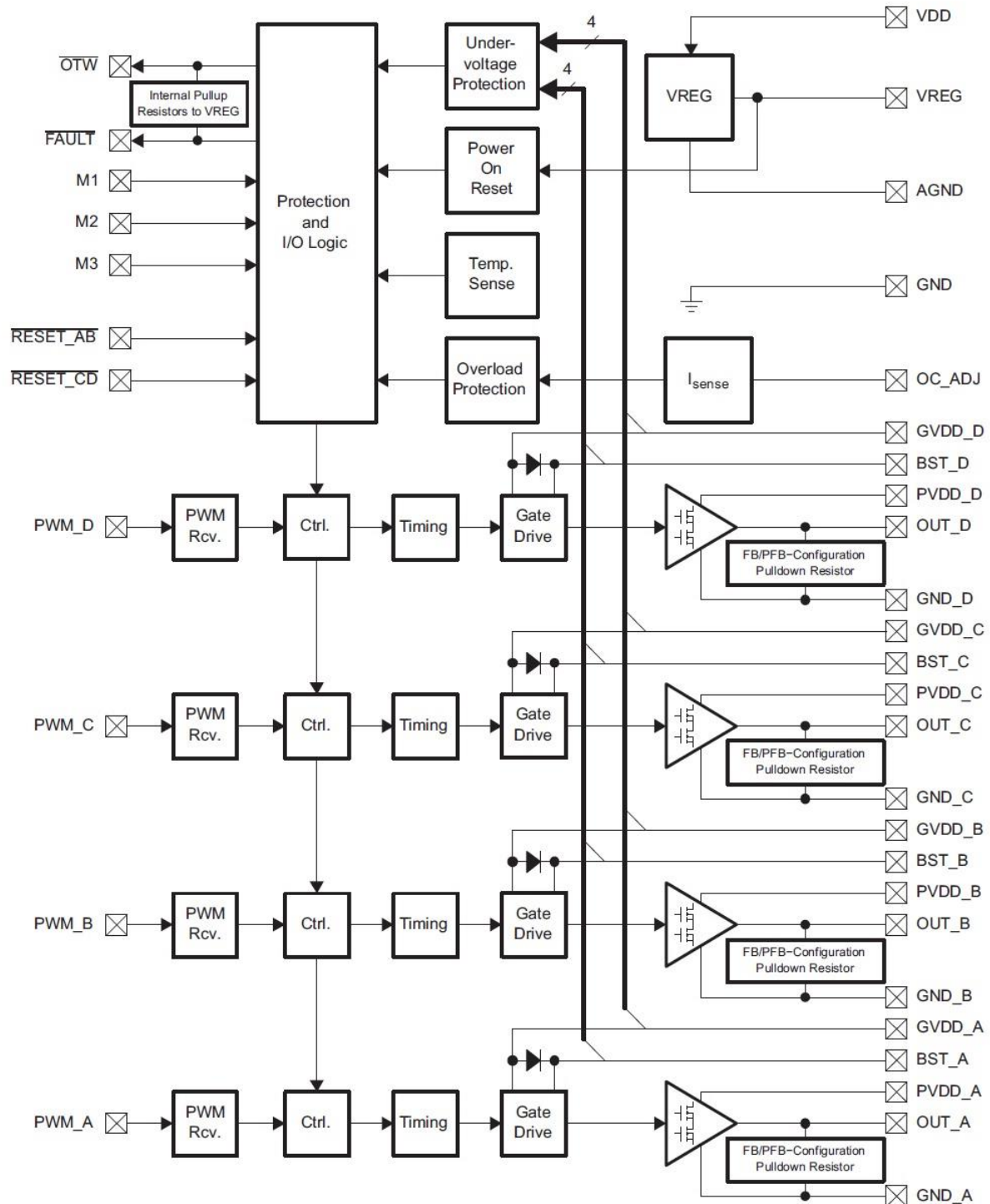
Pin Functions (continued)

PIN			I/O TYPE ⁽¹⁾	DESCRIPTION
NAME	DRV8412	DRV8432		
OUT_D	39	22	O	Output, half-bridge D
PVDD_A	26, 27	34	P	Power supply input for half-bridge A requires close decoupling capacitor to ground.
PVDD_B	32	29	P	Power supply input for half-bridge B requires close decoupling capacitor to ground.
PVDD_C	35	26	P	Power supply input for half-bridge C requires close decoupling capacitor to ground.
PVDD_D	40, 41	21	P	Power supply input for half-bridge D requires close decoupling capacitor to ground.
PWM_A	17	4	I	Input signal for half-bridge A
PWM_B	15	6	I	Input signal for half-bridge B
PWM_C	7	14	I	Input signal for half-bridge C
PWM_D	5	16	I	Input signal for half-bridge D
RESET_AB	16	5	I	Reset signal for half-bridge A and half-bridge B, active-low
RESET_CD	6	15	I	Reset signal for half-bridge C and half-bridge D, active-low
FAULT	18	3	O	Fault signal, open-drain, active-low. An internal pullup resistor to VREG (3.3 V) is provided on output. Level compliance for 5-V logic can be obtained by adding external pullup resistor to 5 V
VDD	2	17	P	Power supply for digital voltage regulator requires capacitor to ground for decoupling.
VREG	11	10	P	Digital regulator supply filter pin requires 0.1-μF capacitor to AGND.
THERMAL PAD	—	N/A	T	Solder the exposed thermal pad to the landing pad on the pcb. Connect landing pad to bottom side of pcb through via for better thermal dissipation. This pad should be connected to GND.
HEAT SLUG	N/A	—	T	Mount heat sink with thermal interface on top of the heat slug for best thermal performance.

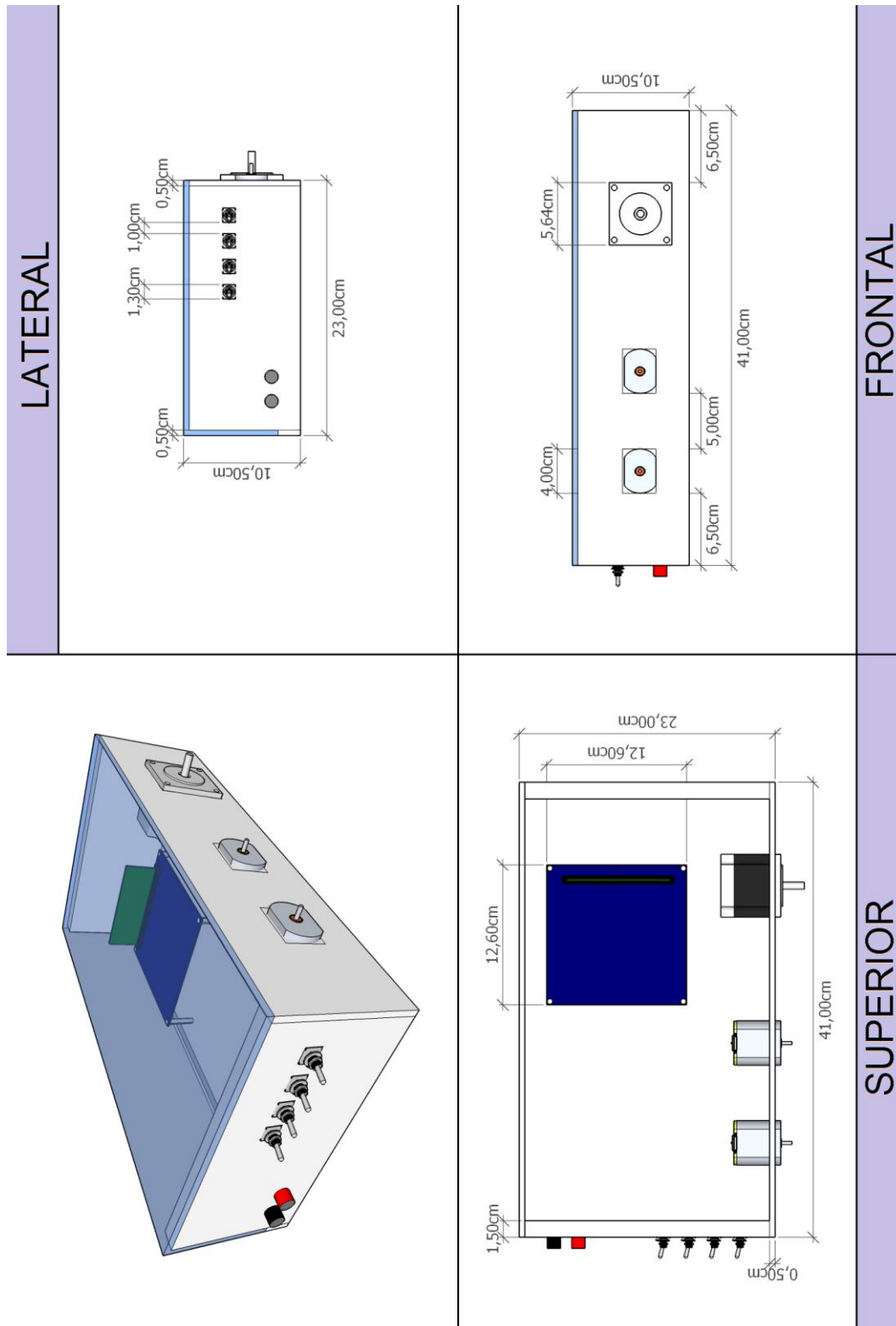
Mode Selection Pins

MODE PINS			OUTPUT CONFIGURATION	DESCRIPTION
M3	M2	M1		
0	0	0	2 FB or 4 HB	Dual full bridges (two PWM inputs each full bridge) or four half bridges with cycle-by-cycle current limit
0	0	1	2 FB or 4 HB	Dual full bridges (two PWM inputs each full bridge) or four half bridges with OC latching shutdown (no cycle-by-cycle current limit)
0	1	0	1 PFB	Parallel full bridge with cycle-by-cycle current limit
0	1	1	2 FB	Dual full bridges (one PWM input each full bridge with complementary PWM on second half bridge) with cycle-by-cycle current limit
1	x	x	Reserved	

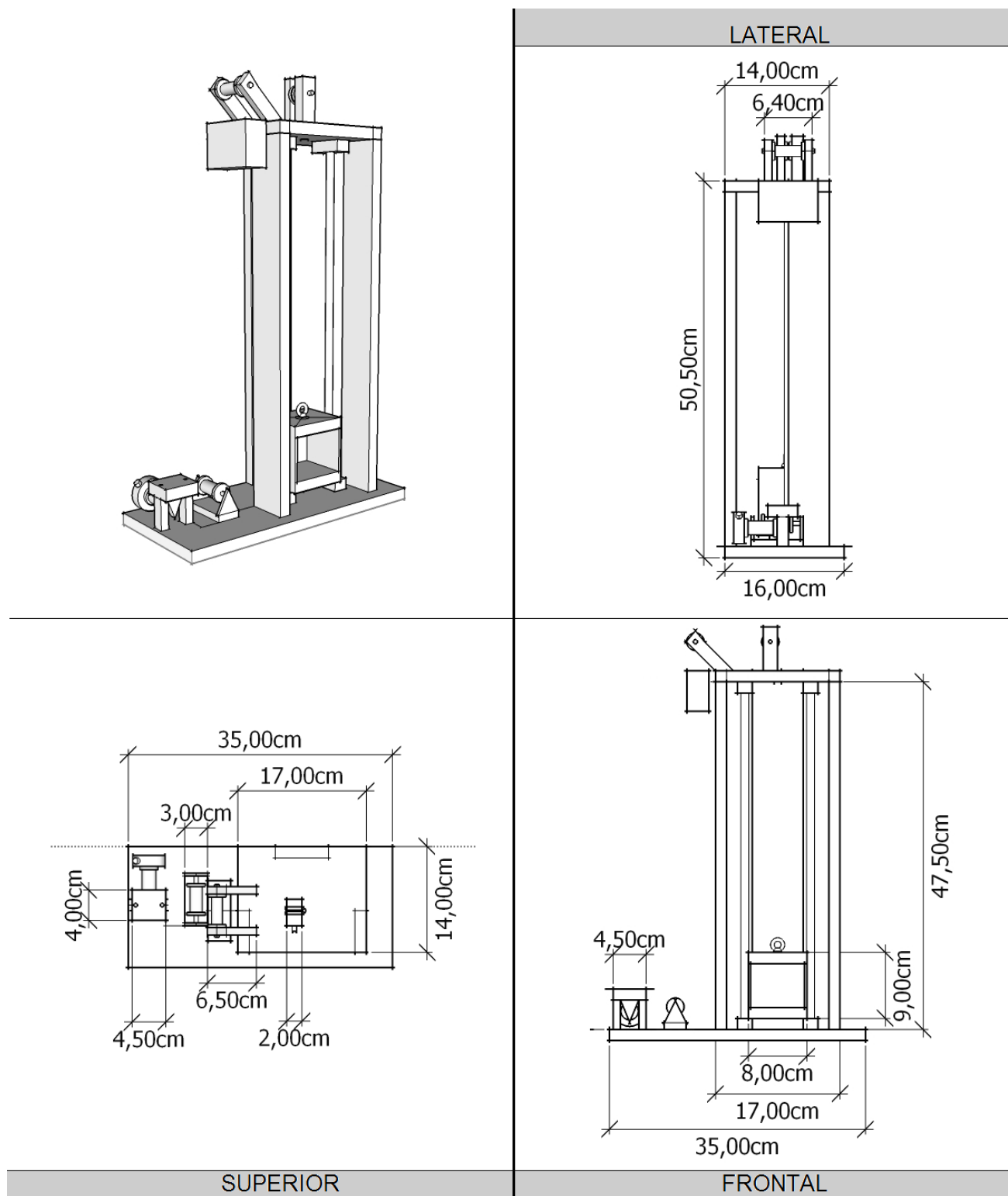
Diagrama interno circuito integrado DRV8412.



ANEXO F: ESQUEMA DISEÑO GABINETE



ANEXO G: ESQUEMA DISEÑO MAQUETA ELEVADOR



ANEXO H: ESQUEMA DISEÑO CINTA TRANSPORTADORA

