

DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO DE CONTROL PARA REALIZAR
EL PROCESO DE RECUBRIMIENTO POR INMERSIÓN

LINA MARCELA RUIZ MARIN

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERIAS
INGENIERÍA ELECTRÓNICA
MEDELLIN
2017

DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO DE CONTROL PARA REALIZAR
EL PROCESO DE RECUBRIMIENTO POR INMERSIÓN

LINA MARCELA RUIZ MARIN

Trabajo de grado para optar al título de Ingeniero Electrónico

Director

FREDDY RAFAEL PÉREZ

Doctor en Física

Asesor

MARYAM DEL MAR CORREA

Ingeniera Electrónica

UNIVERSIDAD PONTIFICIA BOLIVARIANA

ESCUELA DE INGENIERIAS

INGENIERÍA ELECTRÓNICA

MEDELLIN

2017

Notas de aceptación

Firma:

Nombre:

Presidente del Jurado

Firma:

Nombre:

Jurado:

Medellín, Mayo 5 de 2017

26 de Abril de 2016

Yo, Lina Marcela Ruiz Marín

“Declaro que este trabajo de grado no ha sido presentado para optar a un título, ya sea en igual forma o con variaciones, en esta o cualquier otra universidad” Art 82 Régimen Discente de Formación Avanzada.

Firma

A la memoria de Liliana Ruiz Marín

AGRADECIMIENTOS

Agradezco a Dios por darme la sabiduría, el entendimiento, la salud y la vida para terminar este trabajo y poner en mi camino todas las personas que contribuyeron para poder finalizarlo de la mejor manera.

A mis padres por su inagotable apoyo que me permitió superar cada uno de los obstáculos presentados en este proceso de formación que aun continua.

A mis hijos María Paulina y Julián Ruiz Ruiz por ser mi fortaleza y apoyo para continuar mi camino.

A mis hermanas y muy especialmente a Liliana Ruiz (Q.D.E.P) por su apoyo formación y orientación.

A Maryam del Mar Correa, la asesora y codirectora del presente trabajo, porque sin sus enseñanzas, correcciones y aportes este trabajo no hubiera sido posible llevarlo a cabo.

A los profesores de la universidad muy especialmente a Sergio Cock, Diego Cuartas, Alonso Sepúlveda y Hernán Valencia por ayudarme a adquirir una serie de conocimientos y formación durante sus clases.

Al grupo de Investigación de Óptica y Espectroscopía y a su director Freddy Pérez por sus valiosos comentarios y aportes en mi formación.

CONTENIDO

LISTA DE TABLAS	1
LISTA DE FIGURAS	2
LISTA DE ANEXOS	3
GLOSARIO	4
RESUMEN	7
INTRODUCCIÓN	16
1. MODELAMIENTO HIDRODINÁMICO DEL PROCESO DE RECUBRIMIENTO POR INMERSIÓN	18
2. DISEÑO MECÁNICO DEL EQUIPO	20
2.1. Selección del tornillo sin fin	20
2.2. Selección del motor de pasos	20
3. DISEÑO ELECTRÓNICO DEL EQUIPO	25
3.1. Diseño del circuito de interfaz entre el microcontrolador y el motor	25
3.2. Circuito de potencia para el motor de pasos	27
3.3. Configuración del sistema embebido	29
3.3.1. Microcontrolador	29
3.3.2 Periféricos utilizados	29
3.3.3. Ajuste de la frecuencia de operación del PWM	30
3.3.4. Algoritmo del control de velocidad	31
3.3.5. Relación entre la distancia y los resultados del sensor ultrasónico	33
CONCLUSIONES	37
RECOMENDACIONES	38
BIBLIOGRAFÍA	39
ANEXO 1 - DIARIO DE CAMPO	43

ANEXO 2 – ANTEPROYECTO.....	51
ANEXO 3 - MANUAL DE FUNCIONAMIENTO DEL EQUIPO.....	103
ANEXO 4 - DISEÑO DE LA MAQUETA.....	118
ANEXO 5 – PROGRAMAS REALIZADOS.....	121
5.1 Código control de velocidad equipo dipcoating.....	121
5.2. Programa comentado interfaz gráfica en Java.....	146
5.2.1. Código clase Com_micro.....	146
5.2.2 Código de clase Ventana_control.....	149
5.2.3. Código de Clase LectorCom.....	154

LISTA DE TABLAS

Tabla 1. Parámetros físicos para el cálculo del torque del motor.	21
Tabla 2. Datos eléctricos del motor de pasos.	26
Tabla 3. Datos eléctricos de los puertos del microcontrolador.	26
Tabla 4. Datos conocidos del circuito divisor.	29
Tabla 5. Correspondencia entre fases y colores de los cables.	29
Tabla 6. Valores de distancia en función de cuentas promedio.	35
Tabla 7. Valores de pendiente e intercepto.	36
Tabla 8. Coeficientes de función no lineal.	37
Tabla 9. Valores permitidos por la interfaz Java.	117
Tabla 10. Especificaciones eléctricas del equipo.	118
Tabla 11. Especificaciones eléctricas de la fuente.	118
Tabla 12. Datos eléctricos del motor de pasos.	118

LISTA DE FIGURAS

Figura 1. Diagrama de fuerzas en el equipo.	22
Figura 2. Esquemático de configuración del optoacoplador.	27
Figura 3. Conexiones del optoacoplador.	27
Figura 4. Esquemático del circuito de potencia para el motor de pasos Sanyo Denki.	28
Figura 5. Diagrama de flujo del control de velocidad.	33
Figura 6. Interfaz gráfica del equipo.	34
Figura 7. Señales de salida del sensor ultrasónico.	34
Figura 8. Relación lineal entre cuentas y distancia.	36
Figura 9. Relación no lineal entre cuentas y distancia.	37

LISTA DE ANEXOS

Anexo 1. Diario de campo.	44
Anexo 2. Anteproyecto.	53
Anexo 3. Manual de funcionamiento del equipo.	105
Anexo 4. Diseño de la Maqueta.	119
Anexo 5. Programas Realizados.	123

GLOSARIO

AUTODIFUSIÓN: Proceso en el cual átomos del mismo tipo intercambian posiciones dentro de un mismo material produciendo un consumo de energía para realizar el proceso es decir a temperatura ambiente las moléculas de H_2O en el agua pura están en movimiento continuo, desplazándose a través del mismo líquido. (Shackelford, 1992, págs. 156,157).

ÁNGULO DE CONTACTO O DE HUMECTANCIA: Ángulo que forma la superficie de un líquido al entrar en contacto con un sólido. El valor del ángulo de contacto depende principalmente de la relación que existe entre las fuerzas adhesivas y cohesivas del líquido. Cuando las fuerzas adhesivas con la superficie del sólido son muy grandes en relación a las fuerzas cohesivas, el ángulo de humectancia es menor de 90 grados sexagesimales, llegando a humedecer la superficie del sólido. Si por el contrario este ángulo es mayor de 90 no se produce el mojado de la pieza. (Velarde, M.G ; Starov, Victor, 2009, págs. 61-74)

CAPILARIDAD: Capacidad de un líquido de subir o bajar por un tubo capilar muy estrecho y de pequeña sección circular, competencia entre las fuerzas de atracción de las partículas de las moléculas del líquido y las fuerzas de tracción entre el líquido y el tubo que lo contiene. (S. Philips, 2000, p. 443)

CIRCUITO DE CONTROL: Conjunto de dispositivos encargados de administrar, dirigir o regular el correcto funcionamiento de otro sistema, con el fin de obtener el comportamiento deseado.

CIRCUITO DE POTENCIA: Dispositivos electrónicos utilizados para satisfacer las necesidades de corriente y voltaje de la carga, los cuales son superiores a los brindados por el microcontrolador.

DIODO EMISOR DE LUZ: Es un dispositivo electrónico que permite que la corriente fluya en un solo sentido, emitiendo luz a su paso.

DIP COATING O RECUBRIMIENTO POR INMERSIÓN: Proceso por el cual se sumerge un sustrato en un líquido de manera controlada, con el fin de crear un recubrimiento superficial con características específicas obteniendo un determinado comportamiento de la superficie. (Osorio, 2007)

LENGUAJE C: Lenguaje de propósito general utilizado para la implementación de sistemas operativos y aplicaciones, debido a su eficiencia de código y versatilidad. Permite al usuario programar las instrucciones del microcontrolador en alto nivel, mientras que los compiladores las traducen a lenguaje de máquina.

JAVA: Lenguaje de programación orientado a objetos, el cual contiene una estructura modular, es derivada del lenguaje C y permite ser utilizada indiscriminadamente en cualquier tipo de sistema operativo. (Anónimo, Conceptodefinición.de, 2016)

MEMORIA FLASH: Es un dispositivo en forma de tarjeta, que almacena datos en un espacio reducido, permitiendo la lectura y escritura de múltiples posiciones de memoria en la misma operación, a una mayor velocidad. (Anónimo, Culturación, s.f.)

MEMORIA SRAM: “Es aquella que tiene la capacidad de almacenar datos de programas o aplicaciones que estén en ejecución, de manera que el microprocesador pueda acceder a ella para utilizar estos datos y proceder a realizar las acciones que desee el usuario”. (Anónimo, Culturación, s.f.)

MODULO PWM: Parte del microcontrolador encargada de generar una señal modulada por ancho de pulso (PWM) y enviarla a un puerto específico. La señal es periódica y se le puede controlar el ancho del pulso y el tiempo en que la señal se encuentra en un nivel alto, mediante la comparación con una señal de reloj.

MISCIBILIDAD: Cuando se quieren mezclar dos líquidos se producirá que uno de ellos se difunda en el otro con mayor velocidad, es decir líquidos con propiedades similares (polares o no polares) serán miscibles, mientras que líquidos de diferente polaridad no podrán mezclarse tal es el caso del agua y el aceite. Habilidad de dos o más sustancias líquidas para mezclarse entre sí y formar una o más fases o mezclas. (More, 1981, p. 273)

MOTOR DE PASOS: Es un dispositivo electromecánico al que se le envía una señal de pulsos desde el microcontrolador, para realizar desplazamientos angulares en el motor siendo estos precisos y estables. (Anónimo, Oriental Motor, 2012)

NUMERO DE CAPILARIDAD: Valor adimensional utilizado en el análisis del flujo de fluidos que caracteriza la relación de las fuerzas viscosas con las fuerzas de tensión superficiales actuando a través de una interface entre un líquido y un gas. Se denota usualmente con NC en el campo del petróleo y con Ca en la ingeniería química. (Universidad de Valencia, 2009-2010, p. 17)

MENISCO (QUÍMICA): Curva que forma la superficie de un líquido dentro de un tubo angosto. Los meniscos son producidos por las fuerzas de atracción entre el líquido y el tubo que lo contiene provocando que el nivel del líquido cerca a las paredes del tubo sea mayor que en el centro, dándole una superficie cóncava o convexa al fluido. (S. Philips, 2000, pp. 443)

OPTOACOPLADOR: Es un circuito integrado compuesto generalmente por un diodo LED y un fototransistor, conectados en forma tal, que cuando una señal eléctrica circula a través del LED, éste se enciende generando una corriente en la base del fototransistor. (Anónimo, Educachip, 2016)

SPIN COATING: Este método consiste en el depósito de un líquido en el centro del sustrato plano, generalmente circular, que se hace girar a un determinado número de revoluciones por segundo (rpm), con el fin de lograr un recubrimiento delgado y uniforme. Es también llamado depósito químico por proceso de centrifugado.

SUSTRATO: Sustancia u objeto sobre la que actúa, reacciona o se deposita otra. 5(S. Philips, 2000, p. 676).

TENSIÓN SUPERFICIAL: Trabajo que se debe realizar para llevar las moléculas en un número suficiente, desde el interior del líquido hasta la superficie. Cuanto mayor sea la tensión superficial de un líquido, mayor será la resistencia que opone al desplazamiento molecular. Esto ocurre debido a que las fuerzas que operan sobre una molécula son diferentes en el interior del líquido y en la superficie. (S. Philips, 2000, p. 442)

TIMER: Señal de reloj generada por el microcontrolador, para determinar por cuánto tiempo estarán activas algunas funciones requeridas por el usuario. En un microcontrolador, un módulo de temporización permite generar interrupciones y realizar conteos ascendentes y descendentes.

TORNILLO SIN FIN: Mecanismo de transmisión circular compuesto por dos elementos, el tornillo (sinfín) que actúa como elemento de entrada (o motriz) y la rueda dentada, que actúa como elemento de salida (o conducido) llamada también corona. La rosca del tornillo engrana con los dientes de la rueda de modo que los ejes de transmisión de ambos son perpendiculares entre sí. Por lo tanto, por cada vuelta que da el tornillo, el engranaje gira un solo diente, o lo que es lo mismo, es necesario que el tornillo gire tantas veces como dientes tiene el engranaje para que la rueda de una vuelta completa. Se puede deducir que este sistema posee una relación de transmisión muy baja, comportándose como un reductor de velocidad y aportando una mayor ganancia mecánica al sistema. (Universidad Pontificia Bolivariana, Facultad de Ingeniería Mecánica., 2008)

TRANSISTOR BJT O BIPOLAR: Dispositivo amplificador de corriente que tiene tres terminales llamadas base, colector y emisor ; el cual permite que al ingresar una corriente por su base, entregue en el emisor un nivel de corriente superior, siendo este cambio llamado el factor de amplificación.(Anónimo, Electrónica Unicrom, 2016)

TRANSISTOR DARLINGTON: Es un tipo especial de transistor que tiene una muy alta ganancia de corriente debido a que está compuesto internamente por dos transistores bipolares comunes que se conectan en cascada. (Anónimo, Electrónica Unicrom, 2016)

TRANSISTOR MOSFET: Son las siglas de *Metal Oxide Semiconductor Field Effect Transistor*. Es un dispositivo que contiene un sustrato de material semiconductor dopado, en el que mediante técnicas de difusión de dopantes se crean dos islas de tipo opuesto separadas por un área. Estas islas son llamadas Drenador y Surtidor. Sobre las cuales se hace crecer una capa de dieléctrico culminada por una capa de conductor. (García, 2012).

UART: Dispositivo programable en el que pueden establecerse las condiciones de comunicación, que se utilizarán para la transmisión de información entre dos equipos como son la velocidad, paridad, longitud y bits de parada etc. Por sus siglas en inglés *Universal Asynchronous Receiver-Transmitter*. (Anónimo, Tecnología del PC, s.f.)

VISCOSIDAD: “Oposición del fluido a las deformaciones tangenciales debido a las fuerzas de cohesión molecular”. Es una medida de la resistencia de los líquidos a fluir, cuanto más viscoso es un líquido más lento es su flujo. La viscosidad de un líquido tiende a disminuir con el aumento de la temperatura (Chang Reimod ,College Williams, 2002).

RESUMEN

Se da un informe detallado de cómo se realizó el diseño y la implementación de un equipo para realizar procesos de recubrimiento por inmersión. Esta técnica consiste en sumergir y extraer el material a recubrir en un solvente a una velocidad controlada, para su posterior solidificación logrando un grosor y textura determinada. Esta técnica se eligió debido a que su implementación es más sencilla, económica y eficiente; permite una mayor flexibilidad de precursores químicos y la menor generación de residuos tóxicos para el ambiente.

El equipo se desarrolló utilizando un motor paso a paso acoplado a un tornillo de bolas de precisión, usando el microcontrolador Texas Instruments Piccolo F28027F para controlar la velocidad de giro, y un sensor ultrasónico para determinar su posicionamiento a lo largo del tornillo.

En este trabajo se esboza el modelo hidrodinámico del proceso químico de recubrimiento por inmersión para su posterior uso, los criterios analizados para realizar el diseño mecánico del equipo, así como la selección de las piezas, los algoritmos para el control de desplazamiento vertical del objeto a recubrir mediante el uso del controlador, y el diseño electrónico de los circuitos de potencia y control implementados.

El presente documento contiene, además, el código realizado en Java para la interfaz de usuario que facilita el control del equipo y el ingreso de los datos necesarios en el proceso.

PALABRAS CLAVE: *DIPCOATING*, RECUBRIMIENTO POR INMERSION, MOTOR DE PASOS, MICROCONTROLADOR, Java

INTRODUCCIÓN

El presente trabajo fue propuesto por el Grupo de Investigación de Óptica y Espectroscopía de la Universidad Pontificia Bolivariana, debido a que se necesitaba contar con un equipo duradero y de costo razonable que permitiera realizar recubrimientos sobre diferentes tipos de materiales. Se encontró que un equipo de recubrimiento por inmersión (*dip coating*, en inglés) podía cumplir con los requerimientos del grupo de investigación y que su construcción era viable.

Los equipos de recubrimiento por inmersión actuales (Oliveria R M Alfredo, Zarbin C. J Aldo, 2005, p. 1), requieren de un control de velocidad preciso para el motor, debido a que esta se relaciona con el grosor del recubrimiento. Dicho control se realiza desde un computador o microcontrolador, el cual se encarga de regular la velocidad en un motor de corriente directa o en un motor paso a paso. Para el motor de directa, se controla el nivel de voltaje que recibe. Para el motor paso a paso, se controla el número de pulsos y el periodo de cada uno de ellos. En sus inicios, los controladores se programaron en lenguaje Qbasic (Oliveria R M Alfredo, Zarbin C. J Aldo, 2005). Sin embargo, a medida que la tecnología avanzó se han venido programando controladores de velocidad en otros lenguajes más amigables para el diseñador como C, además de utilizar microcontroladores con mayor capacidad de procesamiento y almacenamiento. Actualmente, los equipos pueden ser conectados a través de un puerto serial, paralelo o USB (Oliveria R M Alfredo, Zarbin C. J Aldo, 2005, p. 2) (Anónimo, Nadetech Innovations, s.f., p.p 1-3) siendo el último el más utilizado actualmente.

En cuanto al diseño mecánico del equipo, generalmente se utilizan engranajes y reductores de velocidad y tornillos (Oliveria R M Alfredo, Zarbin C. J Aldo, 2005, p.1), siendo éstos utilizados hasta el momento sin mayores modificaciones. Las variables típicas que ingresa el operario son las siguientes: el número de rotaciones del *beaker* con la mezcla, el tiempo estipulado para el secado de las placas, la velocidad de retiro y el tiempo de inmersión de la placa dentro de la solución (Guillen Baca Yessy B. et al, S.f, pp. 4-5). Otros equipos también permiten ingresar la cantidad de ciclos de inmersión de la placa (Anónimo, Alibaba.com, s.f.). Adicionalmente, cuentan algunos con puertos de comunicación USB, pantalla táctil o panel LCD (Anónimo, Nadetech Innovations, s.f., p.p. 1-3), además de un teclado y una memoria adicional para el almacenamiento de información con el detalle de los procesos realizados para su posterior procesamiento (Anónimo, Alibaba.com, s.f.).

Para este trabajo, se optó por usar un motor paso a paso, debido a que éste requiere menos piezas mecánicas para lograr velocidades muy bajas. Como consecuencia, se buscó un microcontrolador que pudiera generar una señal digital de frecuencia variable y diferida a cuatro fases. También se procuró que la comunicación con el microcontrolador fuera mediante el puerto USB.

En este trabajo, se presentan las ecuaciones hidrodinámicas del menisco, las cuales permiten realizar y predecir el grosor del recubrimiento entre la lámina y el líquido, relacionando las variables implicadas en el proceso, como son la tensión superficial, la viscosidad y la capilaridad, partiendo de la suposición de que el drenaje gravitacional es despreciable.

Luego, se tomarán las consideraciones necesarias al realizar el diseño mecánico para evitar al máximo la vibración del sistema y la selección adecuada del torque del motor paso a paso, de acuerdo con el peso de todo el mecanismo, así como las consideraciones del diseño electrónico para el circuito de potencia y control

1. MODELAMIENTO HIDRODINÁMICO DEL PROCESO DE RECUBRIMIENTO POR INMERSIÓN

Si el número de capilaridad, Ca , satisface la condición:

$$Ca = \frac{\mu v}{\sigma} = \frac{\text{Fuerzas Viscosas}}{\text{Fuerzas capilares}} \ll 1 \quad (1)$$

donde μ representa la viscosidad, v la velocidad del sistema y σ la tensión superficial, entonces se puede construir un modelo basado en las ecuaciones de Landau y Levich para la dinámica de fluidos Newtonianos (Brasjen Bj et, 2010, p.p 1-2). De la ecuación (1) se puede ver que el número de capilaridad es directamente proporcional a la viscosidad del fluido y a la velocidad relativa entre este y la pieza a recubrir, mientras que es inversamente proporcional a la tensión superficial del fluido.

Suponiendo que el drenaje gravitacional es despreciable (Hsu Wei et al, 2008, p. 1), el espesor del recubrimiento satisface la relación:

$$h_0 \propto v^x \quad (2)$$

donde h_0 representa el espesor o ancho del recubrimiento, v es la velocidad del sistema y x se encuentra definida en un rango entre 0.53 y 0.64 para un caso en particular utilizando una solución de polímeros. Es importante aclarar que el valor de esta constante puede variar dependiendo de los materiales a utilizar en el experimento.

La relación definida en la ecuación (2) tiene la siguiente aproximación (Brasjen Bj- A.W Van Cuigk-A.A, 2010, p. 1):

$$h_0 = 0.946 I_a C_a^{2/3} \quad (3)$$

Con la ecuación (3) se define la longitud capilar I_a la cual está en función de la tensión superficial (σ), densidad (ρ) y aceleración de la gravedad (g), pudiéndose encontrar en un rango de 1-3 mm respectivamente. Se expresa de la siguiente manera:

$$I_a = \sqrt{\frac{\sigma}{\rho g}} \quad (4)$$

En una extensión del análisis realizado por Landau y Levich (Brasjen Bj et al, 2010, pág. 1) se tomó valores más grandes de Ca , modificando la ecuación (3) y expresando el espesor del recubrimiento de la siguiente forma:

$$h_0 = I_a(0.946 C_a^{\frac{2}{3}} - 0.107C_a) \quad (5)$$

Dado un espesor deseado h_0 , se usa la ecuación (3) o (5) para hallar el valor numérico de la capilaridad C_a , y posteriormente hallar la velocidad de retiro despejándola de la ecuación (1) de igual manera se realizan los cálculos si se tiene la velocidad y se desea conocer el grosor del recubrimiento utilizando las ecuaciones (1) y (3).

2. DISEÑO MECÁNICO DEL EQUIPO

2.1. Selección del tornillo sin fin

Para la aplicación deseada, se busca conseguir una velocidad mínima de inmersión o retirada de 50 μm por paso. En aras de alcanzar esta velocidad, se prestó atención especial en la selección del tornillo sin fin y del motor paso a paso, de acuerdo con las capacidades del submódulo PWM del microcontrolador Texas Piccolo F28027. Del submódulo mencionado, se pueden generar desde dos pulsos adelante por cada fase generada en cuadratura, así que, se generan dos pasos en adelante. Teniendo en cuenta esta consideración, se seleccionó un motor de 0.9° por paso, que efectivamente resulta en 1.8° por cada movimiento mínimo. Los siguientes cálculos están en la documentación disponible de Oriental Motor (Anónimo, Oriental Motor general Catalog, 2016) para la selección de tornillos sin fin y detalles de operación del motor paso a paso.

De acuerdo con los lineamientos mencionados, se puede hallar el paso P_B del tornillo sin fin (en inglés, *lead*):

$$P_B = \frac{360^\circ \Delta l}{\Delta \theta} \quad (6)$$

donde Δl es la distancia mínima 50 μm y $\Delta \theta$ es el paso mínimo efectivo o distancia mínima recorrida por el motor al enviarle un pulso. Entonces se halla el paso del tornillo, P_B , es 10 mm.

Existen dos grados de alta precisión para los tornillos sin fin, C7 y C10. Se seleccionó la precisión C7 debido a que no afecta el desplazamiento mínimo que se busca, y también, es más económica. El diámetro y la longitud escogidos para el mismo fueron de 15 mm y 60 cm, respectivamente. Se escoge la referencia de BSST1510-600 de Misumi. (Rolled Ball Screws Standard Nut 15: lead 5,10,20, 2016).

Luego, se puede hallar el número A de pulsos a introducir, en función de la distancia deseada d :

$$A = \frac{360d}{\Delta \theta P_B} \quad (7)$$

Por ejemplo, para una distancia de 10 cm (100 mm), el número de pulsos es 2000.

2.2. Selección del motor de pasos

Con base en el tornillo de bolas seleccionado, se establece el torque mínimo que debe tener el motor al hallar la inercia que tendrá el esquema mecánico. El torque a hallar T_M es la suma del torque de carga T_L y el torque de aceleración T_A , multiplicada por un factor de seguridad a criterio del diseñador S_F : (Anónimo, Orientalmotor, 2016)

$$T_M = (T_L + T_A) S_F \quad (8)$$

Para los cálculos siguientes, se consignan en la siguiente tabla los parámetros asumidos y conocidos. Se usan las unidades imperiales y el coeficiente de fricción entre la guía y la madera se tomó desde la literatura (Anónimo, Engineeringtoolbox, s.f.).

Masa del tornillo de bolas	1126 g
Masa de la nuez del tornillo	300 g
Masa de la tabla	500 g
Masa de la pinza y de la lámina	100 g
Masa total	2026 g (4.466 lb)
Eficiencia del tornillo de bolas (η)	0.9
Coeficiente de fricción de la nuez (μ_o)	0.3
Coeficiente de fricción entre la guía y la madera (μ)	0.6
Diámetro del tornillo de bolas (D_B)	1.5 cm (0.59 in)
Longitud total del tornillo de bolas (L_B)	60 cm (23.622 in)
Paso del tornillo de bolas (P_B)	1 cm (0.393 in)
Densidad del material del tornillo de bolas (ρ)	Para el hierro, 4.64 oz/in ³
Resolución deseada (Δl)	50 μ m (0.001968 in)
Ángulo de inclinación del tornillo sin fin (α)	90°
Frecuencia de operación del submódulo PWM (f)	25- 175 Hz (175 Hz)
Tiempo mínimo de movimiento continuo (Δt)	0.1 s
Aceleración gravitacional (g)	386.1 in/s ²

Tabla 1. Parámetros físicos para el cálculo del torque del motor.

Primero, se halla la fuerza F , que ejerce la carga en la dirección de movimiento del tornillo de bolas:

$$F = F_A + m [\text{sen}(\alpha) + \mu \text{cos}(\alpha)] \quad (9)$$

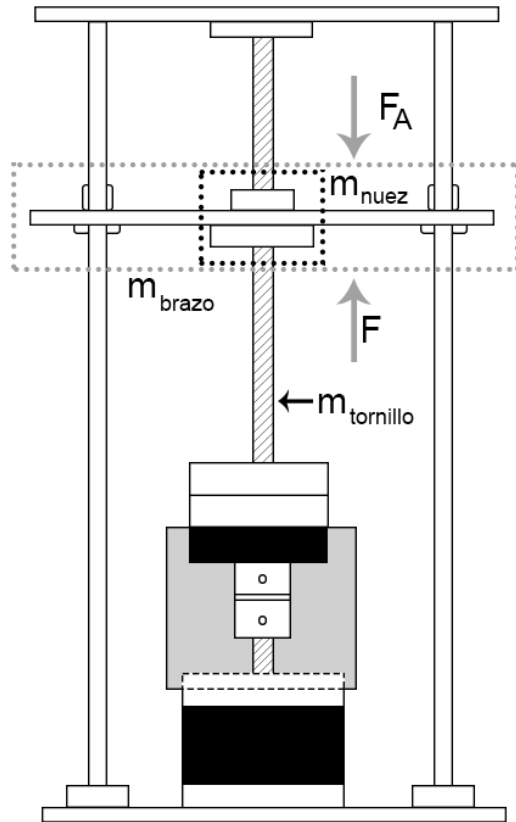


Figura 1. Diagrama de fuerzas en el equipo.

donde F_A es una fuerza externa y m es la masa total del sistema. Si se considera que F_A es nula, entonces:

$$F = 4.466 [\text{sen}(90) + 0.6 \text{cos}(90)] = 4.466 \text{ lb} \quad (10)$$

También se considera que la nuez impone una fuerza adicional F_o :

$$F_o = \frac{F}{3} \quad (11)$$

Entonces, dicha fuerza es de 1.488 lb.

Segundo, se halla el torque de carga siendo la eficiencia del tornillo de bolas η :

$$T_L = \frac{FP_B}{2\pi\eta} + \frac{\mu_o F_o P_B}{2\pi} \quad (12)$$

Teniendo en cuenta los parámetros de la tabla:

$$T_L = \frac{4.466 (0.393)}{2\pi(0.9)} + \frac{0.3 (1.488) 0.393}{2\pi} = 0.3383 \text{ lb} - \text{in} \quad (13)$$

y el resultado anterior también equivale a 5.4128 oz-in.

Si el acople entre el tornillo sin fin y el motor no tiene un mecanismo reductor, entonces el torque hallado no se modifica.

Tercero, se halla la inercia total J_L , que es la suma de la inercia J_B del tornillo de bolas y la inercia J_N de los demás elementos alrededor del tornillo:

$$J_L = J_B + J_N \quad (14)$$

Donde la inercia del tornillo es:

$$J_B = \frac{\pi}{32} \rho L_B D_B^4 \quad (15)$$

Y reemplazando las cantidades, se halla que J_B es 1.3039 lb in².

Luego, se halla la inercia de los demás elementos que aportan masa m , como la nuez, la tabla, la pinza y la lámina:

$$J_N = m \left(\frac{P_B}{2\pi} \right)^2 \quad (16)$$

Teniendo en cuenta los parámetros de la tabla:

$$J_N = 1.984 \left(\frac{0.59}{2\pi} \right)^2 = 0.0175 \text{ lb in}^2 \quad (17)$$

Entonces:

$$J_L = 1.3039 \text{ oz in}^2 + 0.0175 \text{ oz in}^2 = 1.3214 \text{ lb in}^2 \quad (18)$$

Cuarto, se calcula el torque de aceleración:

$$T_A = \frac{J_o + J_L}{g} \frac{\pi \Delta \theta}{180} \frac{f}{\Delta t} \quad (19)$$

donde J_o es la inercia del rotor. Se dejará como incógnita. Sólo se tomará el término independiente, teniendo en cuenta los parámetros de la tabla:

$$T_A = \frac{J_o + 1.3214}{386.1} \frac{\pi(1.8) 175}{180 \cdot 0.1} = 0.1434 J_o + 0.1882 \text{ lb in} \quad (20)$$

donde el término independiente equivale a 3.0112 oz-in.

Hallados tanto el torque de carga como el torque de aceleración, se halla el torque del motor mediante (8), con un factor de seguridad de 2 teniendo en cuenta que el tiempo de operación del equipo es inferior a 24 horas:

$$T_M = (5.4128 + 3.0112)2 = 16.8480 \text{ oz in} \quad (21)$$

Finalmente, se llega a que se necesita un motor que tenga un torque de 16.8480 oz oz-in. Si en algún momento se llega a usar láminas más pesadas, el torque mínimo podría aumentarse. Teniendo en cuenta esta consideración, se buscó un motor que tuviera cerca de 10 veces el torque hallado. La referencia Sanyo-Denki SH1603-0440 cumple con esta especificación, al tener un torque de 240.74 oz-in.

Sin embargo, se desea conocer cuál será la carga máxima (masa) que puede soportar el motor. Para esto, se asume que el torque del motor será de 120.37 oz-in, la mitad del valor máximo.

Si $T_L = 5.4128 \text{ lb} - \text{in}$ y $T_A = 3.0112 \text{ lb} - \text{in}$, entonces la relación entre T_A y T_L es de 1.7975.

Reemplazando en (8) el valor de T_L

$$T_M = 120.37 = (1.7975 T_A + T_A) \cdot 2 \quad (22)$$

Despejando T_A , se llega a que es 21.5139 oz in (1.3446 lb in). Entonces, T_L es 38.6711 oz in (2.4169 lb in).

Reemplazando (11) en (12), se tiene que:

$$T_L = \frac{FP_B}{2\pi\eta} + \frac{\mu_o FP_B}{6\pi} \quad (23)$$

y se halla F. Su valor es de 31.8852 lb. De la ecuación (9), se halla m . No hay fuerza externa, así que $F_A = 0$. Por consiguiente, su valor es 3.8852 lb (1762.32 g).

La carga máxima teórica que puede soportar el equipo es la resta entre la masa máxima hallada y las cantidades de masa mencionadas en la tabla 1. Así, el equipo podría soportar una masa de hasta 962.32 g. No obstante, es recomendable no cargar al máximo el equipo debido a que no se tuvo en cuenta la resistencia a la deformación de los materiales adicionales que componen el equipo como la madera, tornillos de ensamble, pinza, entre otros.

3. DISEÑO ELECTRÓNICO DEL EQUIPO

Partiendo de la finalidad del diseño electrónico, que es encontrar un circuito eficiente, sencillo y económico, evitando inconvenientes que afecten significativamente variables como la corriente o la potencia consumida, se determinó seguir la siguiente metodología realizando las respectivas validaciones en cada etapa. Se parte del análisis de las especificaciones técnicas y las necesidades de corriente, voltaje y potencia del motor paso a paso, para posteriormente continuar con los cálculos para la configuración del submódulo PWM del microcontrolador y el diseño del circuito de potencia, validando su correcto funcionamiento antes de empezar con la implementación del mismo (Carlos J. Jiménez Fernández, Antonio López Ojeda Carlos, León Mora, 2010).

Para el desarrollo y verificación del correcto funcionamiento del equipo se desarrolló una maqueta, que permitió conocer el funcionamiento mecánico y electrónico del sistema, con el fin de ir mejorando el diseño definitivo. Dicha información pertinente al diseño preliminar se encuentra en los anexos quedando registrado en este texto la información del diseño definitivo.

3.1. Diseño del circuito de interfaz entre el microcontrolador y el motor

El diseño del circuito de potencia parte del análisis de la diferencia de voltajes y corrientes que dan los pines del microcontrolador y que recibe cada fase del motor paso a paso. Es así como se determinó usar un optoacoplador que permite aislar cada uno, con el fin de no afectar los pines del microcontrolador por un posible retorno de corrientes parásitas o cortocircuitos.

Numero de fases	Fuente de voltaje (V)	Corriente por fase (A)
4	24	2

Tabla 2. Datos eléctricos del motor de pasos.

Voltaje máximo de salida (V)	Corriente máxima de salida (A)
3	2×10^{-3}

Tabla 3. Datos eléctricos de los pines del microcontrolador.

La característica aisladora del optoacoplador es producida por el uso de un diodo emisor de luz, el cual será activado por la salida de PWM del microcontrolador, y un fototransistor, que es altamente sensible a la luz. Como se observa en la figura 3, al realizar el cierre del interruptor S1 y garantizar el valor de voltaje mínimo para activar el fotodiodo, este emite un rayo de luz el cual genera en el fototransistor un voltaje de base mayor a 0.7 V, permitiendo que el transistor se sature y genere una corriente por el colector. Si por el contrario, se genera un retorno de corrientes parásitas a través de la salida del optoacoplador, éstas serán enviadas a tierra a través del emisor del transistor Q1.

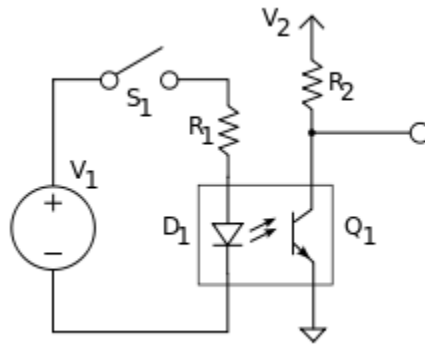


Figura 2. Esquemático de configuración del optoacoplador.

Fuente: Anónimo. (2007, 01 9). Wikipedia. Tomado de http://enhancedwiki.altervista.org/es.php?title=Archivo:Optocouple_circuit.svg

Se seleccionó el optoacoplador 4N25, teniendo en cuenta que el motor necesita 12 V por cada fase. De acuerdo con la hoja de datos, se necesita una corriente de colector de 1 mA para un voltaje entre colector y emisor de hasta 30 V, y una corriente de entrada de hasta 60 mA (Anónimo, Alldatasheet.com, s.f.). Así mismo, si se escoge una corriente baja, la hoja de datos sugiere que habrá una variación mínima frente a cambios de temperatura.

La hoja de datos recomienda que el voltaje de alimentación del optoacoplador sea de 5 V, permitiendo hallar la resistencia de colector de la siguiente forma y omitiendo el voltaje de polarización del diodo emisor de luz:

$$R = \frac{V}{I} = \frac{5V}{1mA} = 5 K\Omega \quad (24)$$

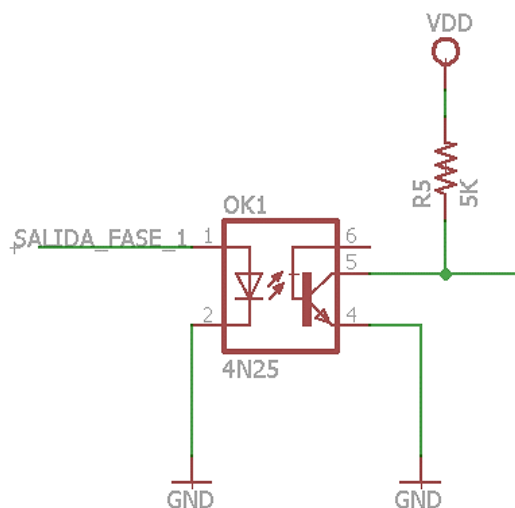


Figura 3. Conexiones del optoacoplador.

3.2. Circuito de potencia para el motor de pasos

Al inicio del proyecto, se dio a conocer que la velocidad requerida está relacionada con la rapidez con que sea retirado el sustrato, además de la vibración del mismo y otras variables químicas como la tensión superficial y la capilaridad, llegando a determinar una velocidad mínima de operación de $0.5 \mu\text{m}/\text{min}$ como un requisito funcional de diseño. Es por esta causa que se seleccionó un motor de pasos que permitiera movimientos más precisos y lentos, utilizando como generador de pulsos el módulo PWM del microcontrolador. Se parte de las especificaciones eléctricas del motor de pasos Sanyo Denki modelo SH1603-0410 para diseñar el circuito de potencia.

Teniendo en cuenta que por cada puerto de salida del microcontrolador sale una corriente de 2 mA y que cada fase del motor necesita 2 A, se concluye que se debe utilizar uno o varios transistores para amplificar la corriente, para lo cual se seleccionaron dos transistores NPN, el 2N2222 y Darlington TIP122.

Se añadió el transistor 2N2222, debido a que experimentalmente ocurrió que el optoacoplador entregaba al transistor Darlington una corriente de 0.4 mA en lugar de 1 mA, como se había supuesto por parámetros de diseño. Esto no permitió que se generara la corriente de base necesaria para saturar el transistor, y que éste a su vez entregara la corriente requerida para el movimiento del motor.

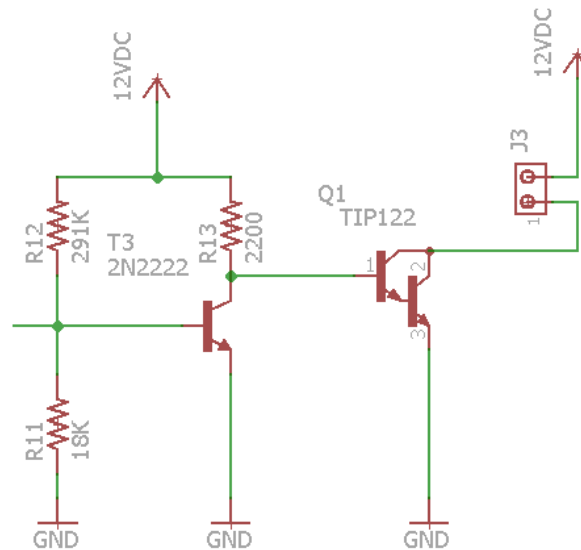


Figura 4. Esquemático del circuito de potencia para el motor de pasos.

Como puede observarse en el esquemático, se utilizó el transistor 2N2222 en configuración de emisor común. Este circuito contiene un divisor de tensión formado por las resistencias R_{11} y R_{12} respectivamente, las cuales están conectadas a la base y una resistencia de colector R_{13} . Se calcularon los valores de las resistencias mediante las siguientes ecuaciones, con base en los parámetros consignados en la tabla 4.

I_b (mA)	I_C (mA)	V_P Optoacoplador. experimental (mV)	V_{BE} (mV)
0.4	10	704	0.7

Tabla 4. Datos conocidos del circuito divisor.

Las resistencias R_{11} y R_{12} se hallan mediante las ecuaciones de resistencia equivalente de Thevenin, R_{eq} , y de divisor de voltaje:

$$R_{eq} = \frac{R_{12}R_{11}}{R_{12} + R_{11}} \quad (25)$$

$$R_{eq} = \frac{R_{12}R_{11}}{R_{12} + R_{11}} \quad (26)$$

$$V_{R_{11}} = V_P + V_{BE} \quad (27)$$

$$V_{R_{11}} = 1.404 = \frac{24R_{11}}{R_{12} + R_{11}} \quad (28)$$

Las ecuaciones (7), (8) y (9) forman un sistema de dos incógnitas así se hallan los valores de R_{11} y R_{12} . Al final, R_{12} es de 292 K Ω y R_{11} es de 18 K Ω , y R_{12} se aproxima a un valor comercial de 270 K Ω .

Se identificaron las dos bobinas del motor midiendo continuidad entre los seis cables, quedando identificados los tres cables pertenecientes a cada bobina. Para identificar el cable que es el común entre ellas, se mide resistencia entre los tres cables. El valor de resistencia mayor corresponde a los extremos de la bobina. El valor de resistencia que da la mitad de resistencia mayor será el común, el cual será conectado a 12 V. A continuación, se muestra una tabla con la correspondencia entre los colores de cada cable y la fase:

Color	Fase	Color	Fase
Piel	0°	Amarillo	270°
Azul	90°	Negro	Común
Rojo	180°	Blanco	Común

Tabla 5. Correspondencia entre fases y colores de los cables.

3.3. Configuración del sistema embebido

3.3.1. Microcontrolador

Para este proyecto, se seleccionó el microcontrolador Piccolo F28027F de Texas Instruments. Este microcontrolador permite un ajuste fino de la modulación por ancho de pulsos, es decir, permite ajustar la frecuencia de operación y el *duty cycle* del módulo PWM de una forma más flexible, siendo esta particularidad indispensable para el desarrollo del control de velocidad. Además, posee una configuración modular ortogonal, es decir, cada módulo es configurado independientemente para descartar fallas y problemas en el sistema más rápidamente. También, es un procesador de 32 bits el cual puede ser programado en un lenguaje de alto nivel como C o C++. Así mismo, es eficiente en el procesamiento de la información realizando cálculos rápidos mediante un DSP interno el cual trabaja con tecnología RISC, que en un sólo ciclo de reloj puede leer y escribir datos en forma simultánea o paralela. El *software* utilizado para la programación y ejecución del programa es de uso libre y no conlleva algún sobre costo adicional; y contiene las librerías para el manejo de todos los submódulos.

Otro dato a tomar en cuenta fue el espacio de memoria disponible del equipo para la programación y procesamiento de la información, la cual está distribuida de la siguiente manera: memoria flash de 32 KB, memoria ROM de 18 KB y la memoria SRAM repartida en dos módulos de memoria, cada uno con capacidad de 16 KB.

Algunas de las características técnicas más relevantes del microcontrolador son las siguientes:

- Frecuencias de operación de 40, 50 y 60 MHz
- Modulación por ancho de pulso (PWM) con ocho canales disponibles
- Modulación PWM de alta resolución (HRPWM) con cuatro canales cuya resolución mínima 310 ps
- Comunicación con otros periféricos por IIC y SPI
- Programación, simulación y depuración por un puerto USB-JTAG
- Comunicación Serial (UART) vía USB
- Flexibilidad de configuración del módulo temporizador (*TIMER*)
- Contador digital de alta resolución para estimar la duración de los pulsos (ECAP)

3.3.2 Periféricos utilizados

3.3.2.1. ECAP

Este periférico es un contador digital que utiliza una base de tiempo de 32 bits y registra hasta cuatro eventos, permitiendo ser configurado de forma tal que se registre un solo evento o eventos continuos. Este periférico permite además ser utilizado como un módulo PWM auxiliar. En esta aplicación, este periférico fue configurado para detectar y medir el ancho de un pulso de eco enviado por el sensor ultrasónico y generar las interrupciones necesarias para cuando ocurra el evento de flanco de bajada del pulso mencionado.

3.3.2.2. PWM

Este módulo realiza el proceso de modulación por ancho de pulso (PWM). Es decir, a la salida del puerto utilizado, se obtiene una señal periódica a la cual se le puede modificar el tiempo en que la señal se encuentra en alto, determinando así un ancho de pulso. Este proceso lo realiza internamente el microcontrolador comparando la señal del PWM con una señal de reloj interno del dispositivo mediante contadores y comparadores dedicados.

Este controlador tiene dos salidas llamadas PWMxA Y PWMxB, cada una con dos canales, los cuales pueden ser configurados de dos formas, como ePWM o HRPWM. La selección de uno de los dos modos está determinado por la frecuencia de operación del sistema, siendo utilizado el modo ePWM generalmente para frecuencias inferiores a 250 KHz y el modo HRPWM para las frecuencias superiores; teniendo en cuenta que para dicha selección se considera el ancho del pulso o resolución requerida por el sistema. El modo de alta resolución del PWM (HRPWM) permite un tamaño de pulso más fino y tiene una resolución mínima de 310 ps. Para el presente control de velocidad, se programó el microcontrolador en el modo ePWM, debido a que el óptimo desempeño del motor sin vibraciones se realiza trabajando a frecuencias entre 25 y 175 Hz.

El control de velocidad se realizó en lenguaje C, entendiendo que se desea producir un movimiento circular en el tornillo de bolas mediante el motor de pasos, y que se deben generar cuatro señales con fases de 0° , 90° , 180° y 270° , respectivamente. Para dicho proceso se utilizó el módulo PWM1A y PWM1B del microcontrolador para generar las señales de 0° y 180° y el módulo PWM2A y PWM2B para generar las señales de 90° y 270° , siendo este último configurado como esclavo del módulo PWM1 con el fin de mantener las cuatro fases sincronizadas.

3.3.2.3. UART

Este puerto es configurado para la comunicación serial a una velocidad de 115200 baudios, sin interrupciones y de 8 bits.

3.3.2.4. TIMER

El microcontrolador posee tres módulos *timer* cada uno de 32 bits, funcionando a una frecuencia de 60 MHz. El *timer* utiliza un registro de conteo descendente, el cual genera una interrupción cuando dicho contador llega a cero y su valor máximo está determinado por la frecuencia del controlador y el tiempo ingresado por el usuario desde la interfaz. Este módulo es el encargado de indicar por cuánto tiempo estará activo el PWM y el tiempo de pausa entre cada grupo de pasos. El *timer* 0 fue utilizado en esta aplicación, dado que es de propósito general y está conectado al bloque PIE que genera todas las interrupciones en el microcontrolador.

3.3.3. Ajuste de la frecuencia de operación del PWM

Para todas las señales del PWM, se escogió un ciclo de trabajo del 50%, generando una señal periódica de igual ancho de pulso en el ciclo alto y en el ciclo bajo de la señal. Además, se tomó una frecuencia de operación del microcontrolador (PLL) de 60 MHz.

Se observó que el motor sin carga trabaja a 500 Hz, siendo ésta la frecuencia asumida inicialmente para el PWM. Sin embargo, se encontró experimentalmente que el motor tiene una mejor respuesta sin vibraciones en un rango de frecuencias de 25 Hz a 175 Hz, siendo éste el intervalo de frecuencias seleccionadas para el control de velocidad.

Del manual del microcontrolador, se tiene que el periodo del PWM (T_{PWM}) está definido por el valor del registro $TBPRD$ y el periodo de la señal de reloj T_{TBCLK} :

$$T_{PWM} = 2(TBPRD) T_{TBCLK} \quad (29)$$

Por definición del periodo, se tiene que:

$$TBPRD = \frac{f_{Micro}}{2 f_{PWM}} \quad (30)$$

De la ecuación anterior se despeja $TBPRD$, asumiendo una frecuencia de 150 Hz para el PWM. El valor de este registro es de 200000. Sin embargo, este registro tiene un tamaño de 16 bits. Por consiguiente, caben datos del orden de $2^{16} - 1$, es decir, caben números de 0 a 65535. Dado que el valor hallado excede este límite, se emplea un divisor de frecuencia (en inglés, *prescaler*) por un factor de 128. Este factor se incorpora en la ecuación anterior.

$$TBPRD = \frac{f_{Micro}}{Prescaler(2 f_{PWM})} \quad (31)$$

Para los cálculos anteriores, se halla un valor de $TBPRD$ de 1562 para la frecuencia de 150 Hz. Se realizan cálculos similares para las frecuencias adicionales, con el ánimo de obtener los valores del registro $TBPRD$ correspondientes.

3.3.4. Algoritmo del control de velocidad

Inicialmente, el algoritmo de control configura todos los módulos necesarios para su funcionamiento. Luego, define los puertos y los inicializa. Posteriormente, lee las variables ingresadas por el usuario a la interfaz de realizada en Java y halla la posición en la que se encuentra el brazo mediante el uso del sensor y las cuentas brindadas por el mismo. Luego convierte las distancias ingresadas a un número de cuentas para el módulo ECAP mediante una ecuación lineal y empieza a comparar si este número es mayor o menor a las cuentas de posicionamiento brindadas por el sensor, tomando la decisión de activar el motor para que realice el movimiento ascendente o descendente del brazo, así como desactivar el motor cuando llegue a la distancia deseada.

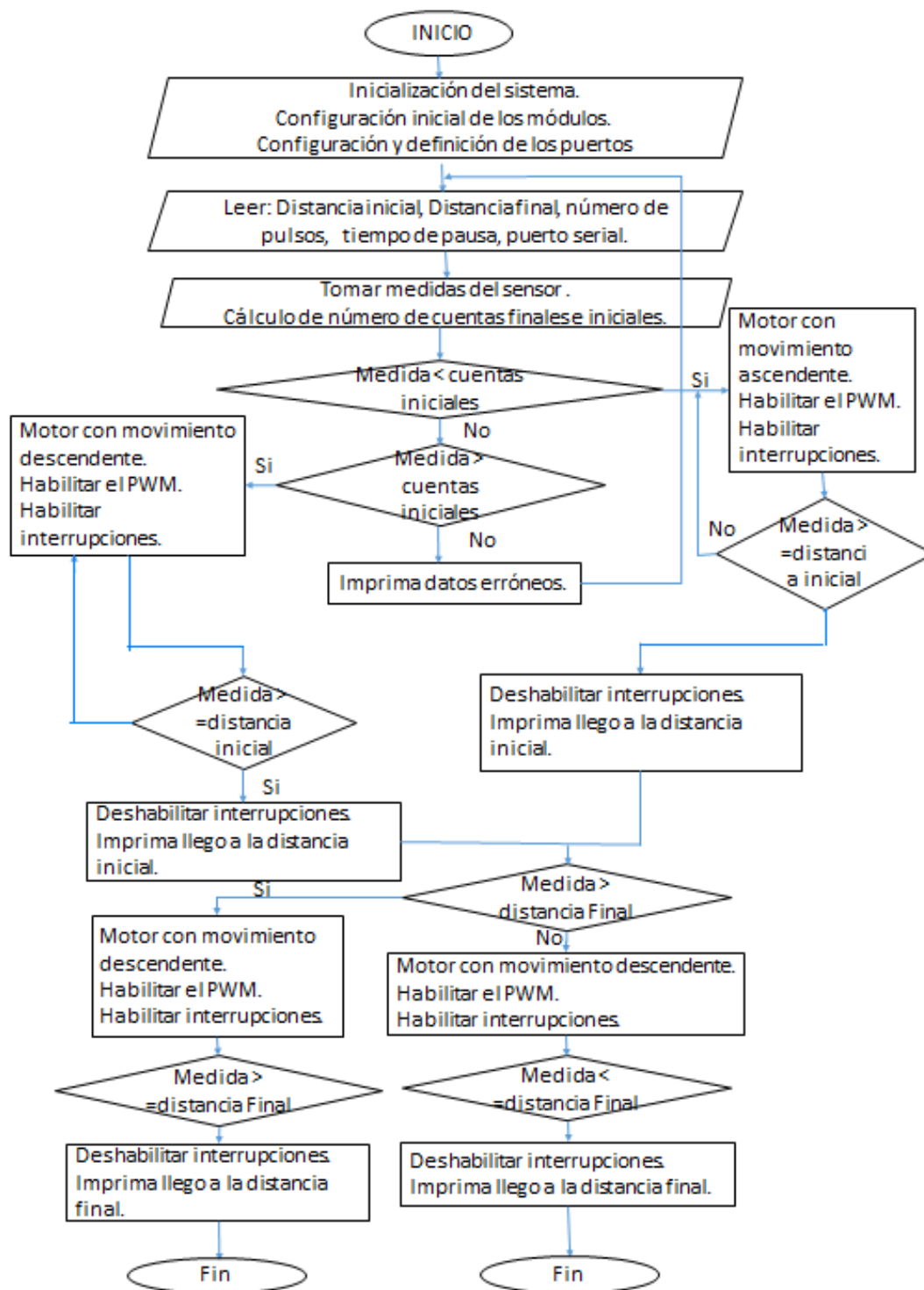


Figura 5. Diagrama de flujo del control de velocidad.

La interfaz realiza permite al usuario el ingreso o selección de los siguientes datos:

Control de Velocidad

Distancia inicial (cm)

Distancia final (cm)

Numero de Pulsos

Tiempo de Pausa (s)

Puerto Serial

Velocidad (mm/s)

Figura 6. Interfaz gráfica del equipo.

3.3.5. Relación entre la distancia y los resultados del sensor ultrasónico

En el equipo se utilizó el sensor HY-SRF05, el cual se alimenta con 5V y se le debe suministrar un pulso de 10 μ s para activar el circuito a través del pin *Trigger*. El módulo lanzará una ráfaga de 8 pulsos ultrasónicos de 40 KHz, pasando a nivel alto hasta que el módulo recibe la señal ultrasónica de retorno o eco, momento en el cual volverá a pasar a un nivel bajo. La salida del pin *Echo* es un pulso cuyo ancho será proporcional a la distancia respecto a un objeto. Si no se detecta un objeto, la salida *Echo* pasará a nivel bajo después de 30ms.

El módulo debe activarse cada 50 ms como mínimo. De esta manera, se asegura que la ráfaga ultrasónica haya desaparecido completamente y no provocará un eco falso en la siguiente medición de distancia.

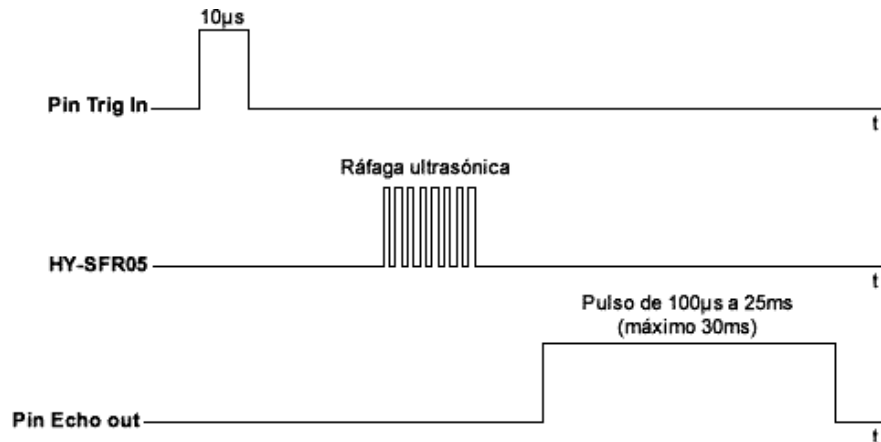


Figura 7. Señales de salida del sensor ultrasónico

Fuente: Anónimo.(s.f) Patagoniatec. Tomado de <http://saber.patagoniatec.com/tutorial-modulo-ultrasonico-arduino-argentina-ptec-sr04-srf05-us020/>

El módulo ECAP del microcontrolador, encargado de convertir la señal de eco a número de cuentas, realiza este proceso a través de la comparación de la señal de reloj del microcontrolador, que tiene una frecuencia de 60 MHz y la señal de salida del pin de eco. El número de cuentas corresponderá a la cantidad de ciclos de reloj que la señal de eco estuvo en alto, siendo este número proporcional a la distancia a la que se encuentra el brazo del equipo.

Para lograr el posicionamiento adecuado del brazo mediante una relación matemática entre la distancia y las cuentas, se tomaron los promedios de 16 tomas del módulo ECAP para diferentes valores de distancia. Los datos se hallan en la siguiente tabla:

Distancia (cm)	Cuentas promedio
3	66892
11	83555
19	91408
27	94097
35	98217
46	117269

Tabla 6. Valores de distancia en función de cuentas promedio.

Luego, se realizó un ajuste lineal con los datos obtenidos por el sensor. Esto, debido a que el microcontrolador presentó restricciones de espacio en memoria y no permitió el cálculo de expresiones de mayor orden polinomial.

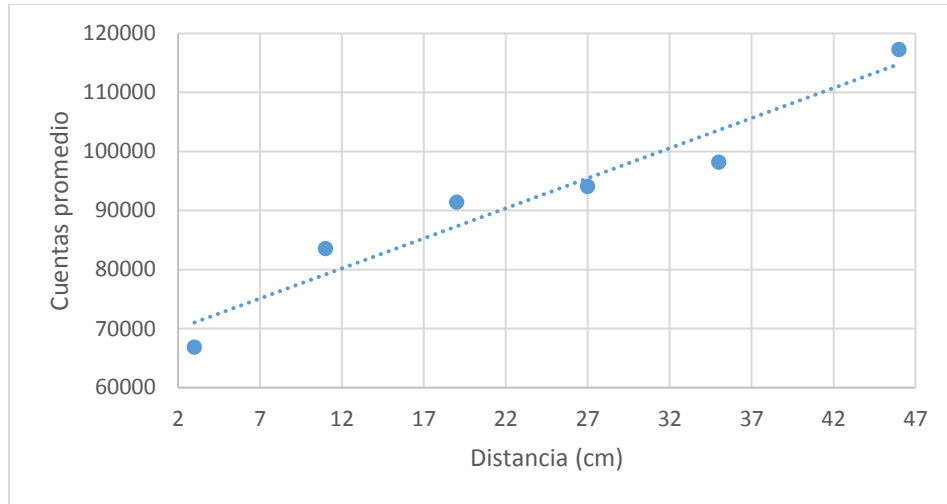


Figura 8. Relación lineal entre cuentas y distancia.

Los parámetros obtenidos del ajuste lineal se muestran a continuación:

	Valor
Pendiente	1018.2989
Intercepto	67976.3068

Tabla 7. Valores de pendiente e intercepto.

Con los parámetros anteriores, se logra un posicionamiento lo suficientemente preciso para el brazo del equipo.

Para un posicionamiento más preciso, se propone definir la función de relación no lineal entre distancia y cuentas, de la forma:

$$f(x) = p_1x^3 + p_2x^2 + p_3x + p_4 \quad (32)$$

para la que se muestran tanto la gráfica como los valores de los coeficientes hallados mediante la herramienta “cftool” de MATLAB.

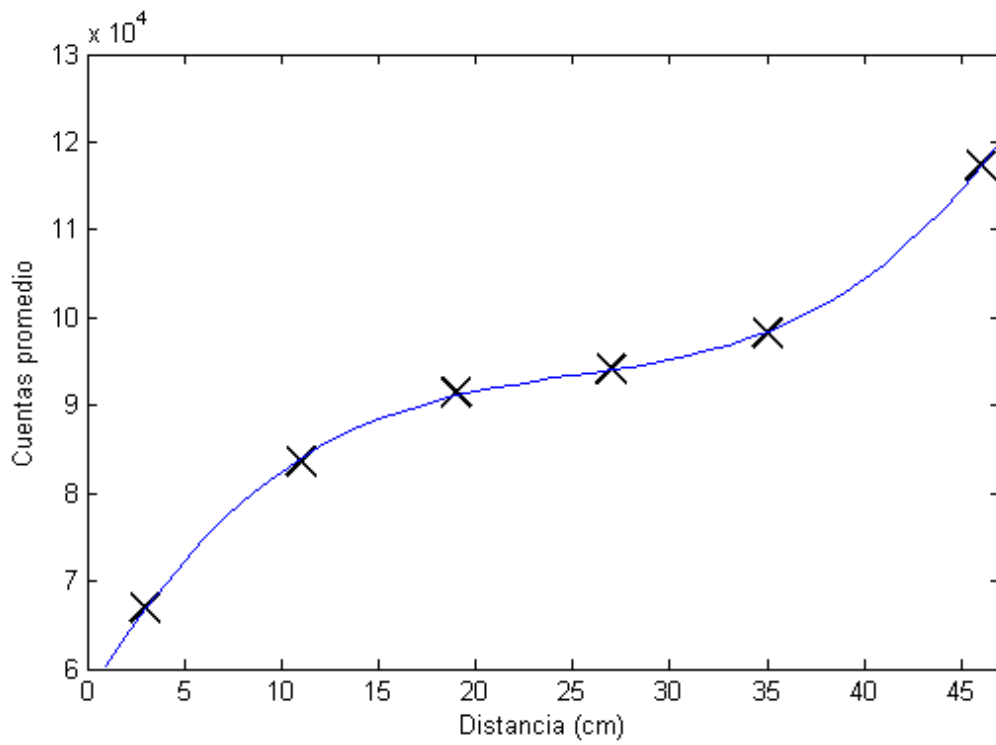


Figura 9. Relación no lineal entre cuentas y distancia.

	Valor
p_1	1.853
p_2	-138.8
p_3	3783
p_4	5.663×10^4

Tabla 8. Coeficientes de función no lineal.

CONCLUSIONES

Se obtuvo un equipo para realizar los procesos de recubrimiento por inmersión, el cual funciona a las velocidades de 2.5 mm/s, 5.0 mm/s, 7.5 mm/s, 10 mm/s, 12.5 mm/s, 13.75 mm/s, 15 mm/s, 16.25 mm/s y 17.5 mm/s. Mediante un sensor ultrasónico, se le dio la capacidad de posicionamiento de la pieza a recubrir. Con un ajuste lineal de los datos obtenidos por el sensor, se logró una aproximación matemática de la ubicación del brazo, pudiendo desplazar el mismo a la distancia requerida por el usuario. Aunque en este trabajo no se hicieron ensayos de inmersión de láminas, en futuros trabajos se espera generar diferentes espesores de recubrimiento de acuerdo con la velocidad de retirada del sustrato.

En cuanto al diseño mecánico implementado, se realizaron los cálculos teóricos del torque mínimo del motor con el fin de estimar la carga máxima que podría soportar el equipo. Así mismo, cumplió los requerimientos básicos para su correcto funcionamiento. También, se redujo al mínimo las vibraciones, y se controló la velocidad y el desplazamiento del brazo. Para futuros trabajos, se puede mejorar cambiando los materiales utilizados para disminuir su peso y hacerlo más versátil.

Se pudo disminuir el consumo de corriente a través del uso del módulo *timer*, el cual está encargado de activar y desactivar las señales enviadas a los puertos del PWM cuando estos no están en uso, logrando la puesta en bajo de todas las salidas del PWM y evitando así un sobrecalentamiento de la fuente de alimentación.

Se logró realizar una interfaz para el equipo en Java, la cual permite su funcionamiento y manipulación por parte del usuario de una manera amigable y eficaz, para el ingreso de los datos necesarios, como son las distancias requeridas, la velocidad, el tiempo de pausa, el número de pulsos y el puerto serial del microcontrolador.

El programa realizado para el microcontrolador consume muchos recursos, razón por la que se hace necesario revisar que el programa pueda almacenarse tanto en la memoria RAM como la memoria *flash* disponible del dispositivo, debido a que esta última puede limitar la programación y uso de la cantidad de módulos utilizados para cada actuador. Esto, para permitir que el programa pueda hacer uso de otros sensores, como el de temperatura y el de humedad.

RECOMENDACIONES

Al seleccionar un microcontrolador, se hace necesario el tratar de estimar cuánto espacio de memoria RAM y de memoria flash el programa consumirá con el fin de evitar los inconvenientes de no poder agregar más sensores. Así mismo, se debe validar si el fabricante del microcontrolador brinda información completa y eficaz sobre su utilización y funcionamiento, debido a que se tuvo que buscar en Internet información adicional de la configuración de la memoria flash y del módulo PWM.

Se recomienda el ingreso de las ecuaciones del modelamiento hidrodinámico del proceso de recubrimiento por inmersión, así como el ingreso de los valores de densidad y tensión superficial para los diferentes polímeros a utilizar en el programa en Java, permitiendo así al equipo trabajar autónomamente calculando la velocidad requerida para lograr un recubrimiento de determinado grosor y viceversa mediante el cálculo de la capilaridad.

Un sensor ultrasónico de posicionamiento más preciso ayudará a eliminar la desviación de los cinco centímetros observados en el brazo luego de que se ingresan los datos a la interfaz en Java.

Se sugiere adicionar entre el tornillo sin fin y el motor, un acople con engranajes reductores de velocidad, que ya se encuentran disponibles en el mercado, con el fin de lograr velocidades inferiores a los 2.5 mm/s.

BIBLIOGRAFÍA

- Chang Reimod ,College Williams . (2002). *Química*. Mexico D.F: Mc Graw Hill.
- Anónimo. (2007, 01 9). *Wikipedia*. Retrieved from http://enhancedwiki.altervista.org/es.php?title=Archivo:Optocouple_circuit.svg
- Anónimo. (2016). *Oriental Motor general Catalog*. Retrieved from www.cnc.info.pl/files/tecmtrsiz_155.pdf
- Anónimo. (2016). *Orientalmotor*. Retrieved from <http://www.orientalmotor.com/technology/articles/motor-sizing-calculations.html>
- Anónimo. (n.d.). *Alibaba.com*. Retrieved from http://www.alibaba.com/product-detail/Precision-Dip-Coater_114025707.html.
- Anónimo. (n.d.). *Alldatasheet.com*. Retrieved from <http://pdf1.alldatasheet.com/datasheet-pdf/view/2846/MOTOROLA/4N25.html>
- Anónimo. (n.d.). *Engineeringtoolbox*. Retrieved from http://www.engineeringtoolbox.com/friction-coefficients-d_778.html
- Anónimo. (n.d.). *holmarc Opto-Mechatronics Pvt.* Retrieved from http://www.holmarc.com/dip_coating_unit_IR.php
- Anónimo. (n.d.). *Nadotech Innovations*. Retrieved from <http://nadotech.com/documents/Datasheet%20ND-DC%20ES.pdf>
- Anónimo. (s.f). *Patagoniatec*. Retrieved from <http://saber.patagoniatec.com/tutorial-modulo-ultrasonico-arduino-argentina-ptec-sr04-srf05-us020/>
- Anónimo. (s.f). *Qualtech Products Industry*. Retrieved from <http://www.qualtechproductsindustry.com/products/cpplication-series-instruments/dip-coater/>
- Anónimo. (s.f.). *Sanyo Denki*. Retrieved from <http://www.tme.eu/es/Document/f522c7d6e80a758642abd6fb7fc95586/sh16.pdf>
- Anónimo. (Sf). *Ministerio de Educación y cultura*. Retrieved from <http://roble.pntic.mec.es/jlop0164/archivos/polarizacion-transistor.pdf>
- Arenas Gaviria, Bernardo . (2012). *Dinámica de un cuerpo libre* . Retrieved from http://aprendeonline.udea.edu.co/lms/moodle/file.php/801/textcuerporigido_2012_.pdf

- Bolívar Osorio, F. J. (2007). *Universidad Complutense de Madrid*. Retrieved from <http://eprints.ucm.es/7551/1/T29978.pdf>
- Brasjen Bj et al. (2010). Dip coating chemically patterned surfaces. *Elsevier*, 1-3.
- Brow, Leman Bursten, Murphy. (2009). *Química La Ciencia Central*. Mexico: Pearson Prentice Hall.
- Brown Lemay, B. (2009). *Química: La ciencia central*. Mexico D.F: Pearson Prentice Hall.
- Carlos J. Jiménez Fernández, Antonio López Ojeda Carlos, León Mora. (2010, Julio Diciembre). Metodología de Diseño Electrónico dentro de Practicas Obligatorias de Laboratorio. *Revista Medios y Educación*, 19-27. Retrieved from <http://www.sav.us.es/pixelbit/pixelbit/articulos/n37/2.pdf>
- Castro, G. (2009). *Departamento de ingeniería Mecánica F.I.U.B.A.* Retrieved from http://campus.fi.uba.ar/file.php/295/Material_Complementario/Aceros.pdf
- E.P.A. United States Environmental Protection Agency. (1996, Abril). *E.P.A. United States Environmental Protection Agency*. Retrieved from <http://www.clu-in.org/download/remed/spanthde.pdf>
- Escuela Americana El Salvador . (N.R). *Concentración de soluciones*. Retrieved from <http://www.amschool.edu.sv/paes/science/concentracion.htm>
- Fernandez, A. C. (2007, Oct). *Propiedades físicas de grafeno en presencia de desorden topológico*. Retrieved from Uniersidad Carlos III de Madrid: http://e-archivo.uc3m.es/bitstream/handle/10016/2370/Tesis_AlbertoCortijo.pdf?sequence=6
- Gonzalez, M. (2011). *Universidad Carlos III de Madrid*. Retrieved from <http://ocw.uc3m.es/cursos-cero/principios-basicos-de-Química/unidad-2/unidad-2>
- Guillen Baca Yessy B, Montoya Portugal Alberto, Palo Tejada Ernesto, Pastor Rodrigues Ronald . (S.f). *Peru Solar org*. Retrieved from <http://www.perusolar.org/include/simposios/archivos16/ponencias16/dip%20coating%20DISEÑO%20Y%20CONSTRUCCION%20DE%20UN%20EQUIPO%20PARA%20DEPOSICION%20DE%20PELICULAS%20DELGADAS.pdf>
- Hole, N. Z. (2009). *Introducción a la mecánica*. N.r: Universidad de Chile. Retrieved from http://www.profesorenlinea.cl/fisica/Fuerzas_Torque_momento.html
- Hsu Wei et al. (2008, 04 16). Dip coating assisted polylactic acid deposition on steel surface: film tickness afeccted by drag force and gravity. *El Sevier*, 1-3. doi:10.1016/j.matlet.2008.04.046
- Jin Tang Wu, Hsin Chun Lai, Sen Yeu Yang, Tzu Chien Huang, Sheng Hsin Wu. (2010). *Elsevier*. Retrieved from Doi:10.1016/j.mee.2010.01.003

- L.A.Pavez F, J.E.Jimenez C,E Ramos M. (2009, 08 2). *Física 3 Texto para el estudiante*. Chile: McGraw Hill. Retrieved from <http://es.wikipedia.org/wiki/Frecuencia>
- More, F. H. (1981). *Fundamentos de Química*. Prentice Hall Hispanoamerica S.a.
- Moreno, V. M. (2007). *Universidad Carlos III de Madrid*. Retrieved from http://e-archivo.uc3m.es/bitstream/handle/10016/1161/pfc_lecho_fluidizado.pdf?sequence=1
- Nisbett, R. G. (2008). *Diseño en ingeniería mecánica* (Octava ed.). Mexico : McGraw Hill. Retrieved from <https://www.dropbox.com/s/w3py0kxed7b13n4/Dise%C3%B1o%20en%20Ingenier%C3%ADa%20Mec%C3%A1nica%20de%20Shigley%20-%208%20Edici%C3%B3n%20-%20Budynas.pdf?dl=0>
- Norton, R. L. (2011). *Diseño de Máquinas un Enfoque Integrado*. Mexico: Pearson Prentice Hall.
- Oliveria R M Alfredo, Zarbin C. J Aldo. (2005). Um procedimento simples e barato para construação de um equipamento "Dip coating" para deposição del filmes em laboratorio. *Quim Nova*, 141-144.
- Osinergmin. Organismo supervisor de la inversión en energía y minería. (2008, Octubre). *Temario 3. Libro de física general*. Retrieved from http://www.osinergmin.gob.pe/newweb/pages/Publico/LV_files/Manual_Fisica_General.pdf
- Osorio, F. J. (2007). *Universidad Complutense de Madrid*. Retrieved from <http://eprints.ucm.es/7551/1/T29978.pdf>
- Raimont A. Serway, Jhon W. Jewett. (2008, abril 21). *Física para ciencias e ingeniería*. Mexico D.F: Cengage learning editores S.a. Retrieved from <http://es.wikipedia.org/wiki/Velocidad>
- Raymond Serway A, Jewett John W. (2008). *Física para ciencias e ingeniería*. Mexico DF: Cengage learning editores. Retrieved from http://es.wikipedia.org/wiki/Velocidad_angular
- Riveiro, L. C. (2008). *Reología de productos alimentarios*. Santiago de Compostela: Universidad de Santiago de Compostela. Retrieved from http://www.slideshare.net/FHEDRA/reologia-de-los-productos-alimenticios-de-unadm-p-tarea?qid=1d95bc56-6994-4159-a11c-6dfa96780d29&v=1&b=&from_search=4
- Roldan Aranda, A. (N.R, N.R). *Departamento de electrónica y tecnología de computadores.Crecimiento de semiconductores*. Retrieved from <http://electronica.ugr.es/~amroldan/deyte/cap09.htm>

- Rolled Ball Screws Standard Nut 15: lead 5,10,20.* (2016). Retrieved from http://us.misumi-ec.com/pdf/fa/2012/p1_0629.pdf
- S. Philips, J. (2000). *Química: Conceptos y aplicaciones*. Mexico D.F: McGraw-Hill.
- Sanchez Echeverria Judith ,García Becerril Maria , Balderas Solano Yolanda. (2008). *Química 1*. Nr: Nr. Retrieved from *Átomo*: <http://definicion.de/atomo/>
- Shackelford, F. (1992). *Introducción a la ciencia de materiales para ingenieros*. Prentice Hall. Retrieved from <http://materias.fi.uba.ar/6713/difusion-repaso.pdf>.
- Shackelford. James, f. (1992, 08 10). *Introducción a la ciencia de los materiales para ingeniería*. 6ta. Retrieved from <http://es.wikipedia.org/wiki/Mol%C3%A9cula>
- TLV Compañía especialista en vapor . (2015). *TLV*. Retrieved from <http://www.tlv.com/global/LA/steam-theory/types-of-steam.html>
- Universidad de Valencia. (2009-2010). *Química Física Avanzada cuarto curso*. Retrieved from http://ocw.uv.es/ciencias/3-2/tema_6_tens_sup.pdf
- Universidad de Valencia. (2009-2010). *Universidad de Valencia Open Course Ware* . Retrieved from http://ocw.uv.es/ciencias/3-2/tema_6_tens_sup.pdf
- Universidad Pontificia Bolivariana . (2008). *Curso de mecanismos 1*. medellin: Universidad Pontificia Bolivariana.
- Universidad Pontificia Bolivariana, Facultad de Ingeniería Mecánica. (2008). *Definiciones básicas curso mecanismos 1*. Medellin Colombia.
- Urien Pinedo, A. (2013, septiembre). *Obtención de biocarbones y biocombustibles mediante pirólisis de biomasa residual*. Retrieved from http://digital.csic.es/bitstream/10261/80225/1/BIOCARBONES_CENIM_CSIC.pdf
- Velarde, M.G ; Starov, Victor. (2009). Wetting concepts and basic features. *Espanola de Física*, 23, 61-74.
- Witascheck, Catrin Strobel Anke, Kadow-Romacker Tom. (2011). Evaluation of process parameter of an automated Dip coating. *El Sevier*, 1-2. Retrieved from doi 10.1016/j.matlet.2011.07.102

ANEXO 1 - DIARIO DE CAMPO

Mayo.2015		
Semana	Actividad	Horas
Semana 1	Búsqueda bibliográfica (Bibliotecas, Internet)	2
Semana 2	Lectura documentos asignados en idioma Ingles	10
Semana 3	Elaboración del anteproyecto	2
Semana 4	Asesoría	7
Apuntes		
	Subtotal	21
	Acumulado	19
Junio.2015		
Semana	Actividad	Horas
Semana 1	Búsqueda bibliográfica (Bibliotecas, Internet)	10
Semana 2	Lectura documentos asignados en idioma Ingles	10
Semana 3	Elaboración del anteproyecto	2
Semana 4	Asesoría	7
Apuntes		
	Subtotal	29

	Acumulado	48
	Julio.2015	
Semana	Actividad	Horas
Semana 1	Búsqueda bibliográfica (Bibliotecas, Internet)	10
Semana 2	Lectura documentos asignados en idioma Ingles	6
Semana 3	Elaboración del anteproyecto	2
Semana 4	Asesoría	8
Apuntes		
	Subtotal	26
	Acumulado	74
	Agosto.2015	
Semana	Actividad	Horas
Semana 1	Diseño mecánico preliminar	10
Semana 2	Elaboración de la Maqueta	1
Semana 3	Elaboración del anteproyecto	8
Semana 4	Asesoría	8
Apuntes		
	Subtotal	27

	Acumulado	101
	Septiembre.2015	
Semana	Actividad	Horas
Semana 1	Diseño mecánico preliminar	10
Semana 2	Elaboración de la Maqueta	1
Semana 3	Elaboración del anteproyecto	8
Semana 4	Asesoría	8
Apuntes		
	Subtotal	27
	Acumulado	128
	Octubre.2015	
Semana	Actividad	Horas
Semana 1	Diseño mecánico preliminar	2
Semana 2	Elaboración de la Maqueta	6
Semana 3	Elaboración del anteproyecto	8
Semana 4	Asesoría	8
Apuntes		
	Subtotal	24

	Acumulado	152
	Noviembre.2015	
Semana	Actividad	Horas
Semana 1	Diseño mecánico preliminar	6
Semana 2	Elaboración de la Maqueta	4
Semana 3	Elaboración del anteproyecto	15
Semana 4	Asesoría	8
Apuntes		
	Subtotal	33
	Acumulado	304
	Diciembre.2015	
Semana	Actividad	Horas
Semana 1	Diseño mecánico preliminar	1
Semana 2	Elaboración de la Maqueta	4
Semana 3	Elaboración del anteproyecto	4
Semana 4	Asesoría	4
Apuntes		
	Subtotal	13

	Acumulado	317
	Enero.2016	
Semana	Actividad	Horas
Semana 1	Elaboración de algoritmos de control electrónico del sistema	8
Semana 2	Validación de funcionamiento en la maqueta	2
Semana 3	Elaboración del documento final	2
Semana 4	Asesoría	7
Apuntes		
	Subtotal	19
	Acumulado	336
	Febrero.2016	
Semana	Actividad	Horas
Semana 1	Elaboración de algoritmos de control electrónico del sistema	20
Semana 2	Validación de funcionamiento en la maqueta	2
Semana 3	Elaboración del documento final	2
Semana 4	Asesoría	8
Apuntes		

	Subtotal	32
	Acumulado	368
	Marzo.2016	Horas
Semana	Actividad	Horas
Semana 1	Elaboración de algoritmos de control electrónico del sistema	20
Semana 2	Validación de funcionamiento en la maqueta	2
Semana 3	Elaboración del documento final	7
Semana 4	Asesoría	2
Apuntes		
	Subtotal	31
	Acumulado	399
	Abril	
Semana	Actividad	Horas
Semana 1	Elaboración de algoritmos de control electrónico del sistema	20
Semana 2	Diseño, manufactura y ensamble de circuitos impresos	4
Semana 3	Elaboración del documento final	2
Semana 4	Asesoría	8

Apuntes		
	Subtotal	34
	Acumulado	433
	Mayo	
Semana	Actividad	Horas
Semana 1	Elaboración de algoritmos de control electrónico del sistema	15
Semana 2	Diseño, manufactura y ensamble de circuitos impresos	10
Semana 3	Implementación de diseño mecánico definitivo	2
Semana 4	Asesoría	8
Apuntes		
	Subtotal	35
	Acumulado	468
	Junio	
Semana	Actividad	Horas
Semana 1	Implementación de diseño mecánico definitivo	15
Semana 2	Ensayo de inmersión y retiro de laminas	4
Semana 3	Optimización del equipo	2

Semana 4	Elaboración del documento final	8
	Asesoría	8
Apuntes		
	Subtotal	37
	Acumulado	505
	Julio	
Semana	Actividad	Horas
Semana 1	Ensayo de inmersión y retiro de laminas	15
Semana 2	Optimización del equipo	20
Semana 3	Elaboración de manual de funcionamiento del equipo	2
Semana 4	Elaboración del documento final	10
	Asesoría	8
Apuntes		
	Subtotal	55
	Acumulado	560

ANEXO 2 – ANTEPROYECTO

Diseño e implementación de un dispositivo de control para realizar el proceso de recubrimiento por inmersión.

Estudiante: Lina Marcela Ruiz M

Cédula: 42691346

Teléfono: 3003739837

ID: 000012665

E-mail: liruib233@gmail.com

Programa: Ingeniería Electrónica

Director: Freddy Rafael Pérez

Cédula: 15.043.436 de Sahagún

Teléfono: 3007809856

ID: 000000149

E-mail: fredy.perez@upb.edu.co

Institución: Universidad Pontificia Bolivariana

Grupo de investigación: Grupo de Óptica y Espectroscopía

Teléfono: 3544522 ext 14003.

Institución: Universidad Pontificia Bolivariana

E-mail: fredy.perez@upb.edu.co

Codirector: Maryam del Mar Correa

Teléfono: 3192463351

Universidad Pontificia Bolivariana

E-mail: mariam.delmarc@gmail.com

Asesor: Andrés Felipe Betancur

Teléfono: 3007886854

Universidad Pontificia Bolivariana

E-mail: andresfebelo@gmail.com

Facultad de ingeniería Electrónica

Universidad Pontificia Bolivariana

Agosto de 2014

Medellín

**DISEÑO E IMPLEMENTACION DE UN DISPOSITIVO DE CONTROL PARA
REALIZAR EL PROCESO DE RECUBRIMIENTO POR INMERSION.**

Primera revisión

Recibió: _____

Fecha: _____

Lectura y asignación de evaluador en comité #: _____

Fecha: _____

Recomendación: _____

Reprobado:

Aplazado:

Aprobado:

Comité #:

Firma responsable:

Fecha:

Comentarios:

Título del proyecto

Código

**DISEÑO E IMPLEMENTACION DE UN DISPOSITIVO DE CONTROL PARA
REALIZAR EL PROCESO DE RECUBRIMIENTO POR INMERSION.**

Segunda revisión

Recibió: _____

Fecha: _____

Lectura y asignación de evaluador en comité #: _____

Fecha: _____

Recomendación: _____

Reprobado:

Aplazado:

Aprobado:

Comité #:

Firma responsable:

Fecha:

Comentarios:

<hr/> <hr/> <hr/> <p>Título del proyecto</p> <p>Código</p>
--

Primera revisión

Fecha de Radicación (Secretaría Académica)	
Fecha estudio del comité:	
Evaluación asignada a:	
Recomendación:	<input type="checkbox"/> Aceptado <input type="checkbox"/> Rechazado
Firma y Fecha	
ACCIÓN: (Secretaría Académica)	<input type="checkbox"/> Carta

Segunda revisión

Fecha de Radicación (Secretaría Académica)	
Fecha estudio del comité:	
Evaluación asignada a:	
Recomendación:	<input type="checkbox"/> Aceptado <input type="checkbox"/> Rechazado
Firma y Fecha	
ACCIÓN: (Secretaría Académica)	<input type="checkbox"/> Carta

TABLA DE CONTENIDO

1. Involucrados.
 - Datos del estudiante
 - Datos del director.
 - Datos del grupo de investigación
2. Glosario preliminar
3. Modalidad
4. Tema del proyecto
5. Antecedentes
 - 5.1. Origen de la idea

5.2. Estado del arte.

6. Marco teórico

6.1. Recubrimientos por difusión

6.2. Recubrimientos por sobrecapa

6.3. Recubrimientos por barreras térmicas

6.4. Recubrimiento por depósito en fase vapor

6.4.1. Depósito físico en fase vapor (PVD).

6.4.1.1. Técnica de evaporación

6.4.1.2. Técnica de pulverización catódica (sputtering)

6.4.2. Depósito químico en fase vapor (CVD)

6.4.2.1. Depósito químico por proceso de inmersión (dip coating)

6.4.2.2. Depósito químico por proceso de centrifugado (spin coating)

6.4.2.3. Depósito químico por aerosol (*spray coating*).

6.5. Comparación de los métodos de recubrimiento por depósito.

6.6. Modelamiento matemático hidrodinámico del proceso del recubrimiento

por inmersión.

7. . Objetivos

7.1. General

7.2. Específicos

8. Justificación y beneficios.

9. Alcances.

10. Metodología

11. Contenido preliminar

12. Productos esperados

13. Presupuesto y recursos necesarios

14. Financiación.

15. Cronograma de actividades.

16. Referencias preliminares

17. Propiedad intelectual y destinación del proyecto

18. Cartas de aval del anteproyecto

1. INVOLUCRADOS

Estudiante: Lina Marcela Ruiz M

Cédula: 42691346

Teléfono: 3003739837

ID: 000012665

E-mail: lirui233@gmail.com

Programa: Ingeniería Electrónica

Director: Freddy Rafael Pérez

Cédula: 15.043.436 de Sahagún

Teléfono: 3007809856

ID: 000000149

E-mail: fredy.perez@upb.edu.co

Institución: Universidad Pontificia Bolivariana

Grupo de investigación: Grupo de Óptica y Espectroscopía

Teléfono:

Institución: Universidad Pontificia Bolivariana.

E-mail: fredy.perez@upb.edu.co

Cédula: 15.043.436 de Sahagún

Codirector: Maryam del Mar Correa

Teléfono: 3192463351

Universidad Pontificia Bolivariana

E-mail: mariam.delmarc@gmail.com

Asesor: Andrés Felipe Betancur

Teléfono: 3007886854

Universidad Pontificia Bolivariana

E-mail: andresfebelo@gmail.com

2. GLOSARIO PRELIMINAR

Autodifusión: Átomos del mismo tipo que intercambian posiciones dentro de un mismo material produciendo un consumo de energía para realizar el proceso es decir a temperatura ambiente las moléculas de H_2O en el agua pura están en movimiento continuo, desplazándose a través del mismo líquido. (Shackelford, 1992, págs. 156,157)

Ángulo de contacto o de humectancia: Ángulo que forma la superficie de un líquido al entrar en contacto con un sólido. El valor del ángulo de contacto depende principalmente de la relación que existe entre las fuerzas adhesivas y cohesivas del líquido. Cuando las fuerzas adhesivas con la superficie del sólido son muy grandes en relación a las fuerzas cohesivas, el ángulo de humectancia es menor de 90 grados sexagesimales, llegando a humedecer la superficie del sólido. Si por el contrario este ángulo es mayor de 90 no se produce el mojado de la pieza. (Velarde, M.G ; Starov, Victor, 2009, págs. 61-74)

Capilaridad: Capacidad de un líquido de subir o bajar por un tubo capilar muy estrecho y de pequeña sección circular, competencia entre las fuerzas de atracción de las partículas de las moléculas del líquido y las fuerzas de tracción entre el líquido y el tubo que lo contiene. (S. Philips, 2000, pág. 443)

Crecimiento Epitaxial: Proceso utilizado en la fabricación de integrados, a partir de una cara de un cristal de material semiconductor o sustrato. Se hace crecer una capa uniforme y de poco espesor con la misma estructura cristalina que éste. Mediante esta técnica se puede controlar de forma muy precisa el nivel de impurezas en el semiconductor, que es lo que define su carácter de donador o receptor. Este proceso se realiza calentando el semiconductor hasta casi su punto de fusión y poniéndolo en contacto con el material de base, para que al enfriarse se cristalice con la estructura adecuada. (Roldan Aranda, N.R, pág. 1)

Dip coating o recubrimiento por inmersión: Proceso por el cual se sumerge un sustrato en un líquido de manera controlada con el fin de crear un recubrimiento que lo proteja. (Osorio, 2007)

Grafeno: Es un cristal con estructura bidimensional de átomos de carbono fuertemente cohesionados en una superficie uniforme ligeramente plana, (1nm de espesor) y que conforman una malla semejante a un panel de abejas dado la configuración hexagonal que presenta. Este material viene descrito por la ecuación de Dirac sin masa en (2+1) dimensiones, siendo esta la principal ventaja de sus propiedades electrónicas y de transporte, las cuales no se encuentran presentes en semiconductores de baja dimensionalidad, en comparación con el grafito es más ligero en cuanto al peso se refiere (Fernandez, 2007)

Nucleación: Cambios que ocurren dentro del núcleo del átomo de un elemento o un isótopo, durante una reacción nuclear diferente del elemento original. Cambio de estado en una región pequeña pero estable. El cambio de estado puede ser la formación de gas o cristal a partir de un líquido. (More, 1981, págs. 542,545,544)

Numero de capilaridad: Valor adimensional utilizado en el análisis del flujo de fluidos que caracteriza la relación de las fuerzas viscosas con las fuerzas de tensión superficiales actuando a través de una interface entre un líquido y un gas. Se denota usualmente con NC en el campo del petróleo y con Ca en la ingeniería química. (Universidad de Valencia, 2009-2010, pág. 17)

Menisco (química): Curva que forma la superficie de un líquido dentro de un tubo angosto. Los meniscos son producidos por las fuerzas de atracción entre el líquido y el tubo que lo contiene provocando que el nivel del líquido cerca a las paredes del tubo sea mayor que en el centro, dándole una superficie cóncava o convexa al fluido. (S. Philips, 2000, pág. 443)

Miscibilidad: Cuando se quieren mezclar dos líquidos se producirá que uno de ellos se difunda en el otro con mayor velocidad, es decir líquidos con propiedades similares (polares o no polares) serán miscibles, mientras que líquidos de diferente polaridad no podrán mezclarse tal es el caso del agua y el aceite. Habilidad de dos o más sustancias líquidas para mezclarse entre sí y formar una o más fases o mezclas. (More, 1981, pág. 273)

Spin coating: Este método consiste en el depósito de un líquido en el centro del sustrato plano generalmente circular que se hace girar a un determinado número de revoluciones por segundo (rpm) con el fin de lograr un recubrimiento delgado y uniforme es también llamado depositación química por proceso de centrifugado.

Spray coating: Es llamada también Depositación química por aerosol técnica utilizada para hacer recubrimientos delgados y multicapa, la cual consiste en recubrir la pieza o el sustrato utilizando gases comprimidos, los cuales se deben aplicar a una determinada temperatura, presión, distancia y flujo controlados.

Sputtering o Técnica de pulverización catódica: En este método, los átomos que realizan el recubrimiento se obtienen mediante el bombardeo de unos blancos metálicos o cerámicos, con iones de un gas inerte (argón) a baja energía (500-1000 eV). Estos átomos que se desprenden del blanco, viajan hasta la superficie de la muestra y se depositan sobre la misma en proporciones similares a la del blanco. Este proceso también permite el depósito simultáneo de dos o más materiales (Bolívar Osorio, 2007, p. 38).

Sustrato: Sustancia u objeto sobre la que actúa o reacciona otra. (S. Philips, Química: conceptos y aplicaciones, 2000, pág. 676).

Tensión superficial: El trabajo que debe realizarse para llevar las moléculas en un número suficiente desde el interior del líquido hasta la superficie. Fuerza necesaria para vencer las atracciones interno moleculares y abrirse paso por la superficie de un líquido o distribuirlo, cuanto mayor sea la tensión superficial de un líquido, mayor será la resistencia que se

oponga en su superficie. Esto ocurre debido a que las fuerzas que operan sobre una molécula son diferentes en el interior del líquido y en la superficie. (S. Philips, Química: conceptos y aplicaciones, 2000, pág. 442)

Tornillo sin fin: Mecanismo de transmisión circular compuesto por dos elementos, el **tornillo (sinfín)** que actúa como elemento de entrada (o motriz) y la **rueda dentada**, que actúa como elemento de salida (o conducido) llamada también **corona**. La rosca del tornillo engrana con los dientes de la rueda de modo que los ejes de transmisión de ambos son perpendiculares entre sí. Por lo tanto, por cada vuelta que da el tornillo, el engranaje gira un solo diente, o lo que es lo mismo, es necesario que el tornillo gire tantas veces como dientes tiene el engranaje para que la rueda de una vuelta completa. Se puede deducir que este sistema posee una relación de transmisión muy baja, comportándose como un reductor de velocidad y aportando una mayor ganancia mecánica al sistema. (Universidad Pontificia Bolivariana, Facultad de Ingeniería Mecánica., 2008)

Viscosidad: “Oposición del fluido a las deformaciones tangenciales debido a las fuerzas de cohesión molecular”. Es una medida de la resistencia de los líquidos a fluir, cuanto más viscoso es un líquido más lento es su flujo. La viscosidad de un líquido tiende a disminuir con el aumento de la temperatura (Chang Reimod ,College Williams, 2002)

3. MODALIDAD

Apoyo a la investigación: Este trabajo está dirigido por el Dr. Freddy Pérez y asesorado por el ingeniero químico Andrés Felipe Betancur Lopera y la ingeniera electrónica Maryam del Mar Correa pertenecientes al grupo de investigación de Óptica y Espectroscopía de la Universidad Pontificia Bolivariana.

La tesis doctoral del ingeniero Andrés Betancur propone investigar inicialmente los resultados obtenidos con la optimización del depósito del grafeno sobre superficies metálicas de carácter comercial, para lo cual se hace necesario diseñar y construir un equipo que permita realizar el proceso de recubrimiento por inmersión o *dip coating* por su designación en inglés, el cual se describirá en el presente trabajo.

El presente trabajo de grado hace parte de uno de los productos adscritos al proyecto interno “Establecimiento de condiciones óptimas para el depósito de grafeno sobre superficies metálicas”, con el código de radicado 102B-02/13-27 en el CIDI.

La modalidad de trabajo es la siguiente:

		Tipo	%
Teórico	Búsqueda		20
	Desarrollo		40
Experimental			
Aplicado	Prototipo		30
	De campo		
Gestión			10

Total	100
-------	-----

4. TEMA DEL PROYECTO

La presente propuesta de trabajo de grado trata sobre el diseño y construcción de un equipo para depositar recubrimientos sobre un sustrato mediante la técnica de recubrimiento por inmersión, con su respectivo control de tiempo de inmersión del sustrato y velocidad de su retiro. Para este fin, se implementará un sistema automático de control en un microcontrolador Piccolo F28027F de Texas Instruments; presentando además los siguientes documentos de soporte, tales como manuales de operación y cuidado, criterios de selección de piezas mecánicas y los archivos de diseño electrónico realizados.

Área	Porcentaje (%)
Microcontroladores	20
Circuitos electrónicos	20
Programación aplicada	60
Total	100

5. ANTECEDENTES

5.1 Origen de la idea

El depósito por inmersión es una técnica clásica y simple de recubrir piezas o sustratos con una película líquida y uniforme para su posterior solidificación como recubrimiento. Esta técnica consiste en sumergir y extraer el material a recubrir en un solvente adecuado de manera controlada con el fin de lograr un grosor y calidad determinado manteniendo así el flujo del líquido constante.

Este trabajo de grado nace de la necesidad del Grupo de investigación de Óptica y Espectroscopía de contar con un equipo robusto, duradero y de costo razonable que permita analizar, investigar y mejorar los procesos realizados a través del recubrimiento por inmersión. Dicha técnica se implementará por ofrecer el menor costo, la implementación más sencilla, la mayor flexibilidad de precursores químicos y la menor generación de residuos, para el recubrimiento de placas metálicas con nanoestructuras de carbono disueltas en soluciones poliméricas.

5.2. Estado del arte

En la actualidad se pueden encontrar diferentes publicaciones científicas respecto a los equipos desarrollados, los cuales son de utilidad para los proyectos de investigación relacionados con recubrimientos delgados y uniformes de sustratos. Algunos de estos artículos fueron publicados por (Guillen Baca Yessy B, Montoya Portugal Alberto, Palo Tejada Ernesto, Pastor Rodrigues Ronald , S.f, pág. 1) (Witascheck, Catrin Strobel Anke, Kadow-Romacker Tom, 2011)) y (Oliveria R M Alfredo, Zarbin C. J Aldo, 2005, pág. 2)

Los equipos fueron diseñados para ser aplicados en catalizadores UV para la descontaminación del agua y aire, nanomateriales en sus diversas aplicaciones en ingeniería, además de su utilización en la industria farmacéutica, fotolitografía y fabricación de implantes ortopédicos. En general, estos equipos según (Oliveria R M Alfredo, Zarbin C. J Aldo, 2005, pág. 1) cuentan con un control de velocidad del motor muy preciso, el cual es realizado a través de un computador o controlador. Cabe anotar, que los autores mencionados hacen referencia a que el software de control fue diseñado inicialmente en el lenguaje de programación Qbasic, permitiendo que, por cada pulso enviado, el motor se mueva un paso de la circunferencia de rotación, siendo necesario 200 pasos para que el eje del motor rote una vuelta completa, pudiendo de esta manera controlar eficientemente la velocidad del dispositivo y garantizar el desplazamiento correcto de inmersión de la placa.

En términos generales, los equipos citados deben ser conectados a través de uno de los siguientes puertos: serial, paralelo o USB (Oliveria R M Alfredo, Zarbin C. J Aldo, 2005,

pág. 2) (Anonimo, s.f., págs. 1-3) con el objetivo de lograr una interfaz que permita la comunicación y control del circuito de potencia del motor de pasos y la extracción de los datos experimentales realizados.

En cuanto al diseño mecánico del equipo, generalmente se utilizan engranajes y reductores de velocidad, además de un par de guías que evitan la generación de momentos angulares en el plano XY o perturbaciones de algún otro tipo. (Oliveria R M Alfredo, Zarbin C. J Aldo, 2005, pág. 1).

Respecto a la medición del espesor del recubrimiento resultante, el cual se debe hacer sin tocar la placa, se observa que pocos dispositivos cuentan con esta función, teniendo que recurrir a un equipo externo que rote en el plano X e Y además de un espectroscopio ((NanoCalc-2000-VIS, Mikropack, Germany) para tomar medidas en diferentes áreas de la placa obteniendo una diferencia de espesor menor al 1% según lo documentado por (Jin Tang Wu, Hsin Chun Lai, Sen Yeu Yang, Tzu Chien Huang, Sheng Hsin Wu, 2010, pág. 2), o utilizar un micrómetro digital para realizar la medición. (Witascheck, Catrin Strobel Anke, Kadow-Romacker Tom, 2011, pág. 2).

Hasta el momento, las variables que ingresa el operario para el control del proceso químico son el número de rotaciones del beaker con la mezcla, tiempo estipulado para el secado de las placas, velocidad de retiro y tiempo de retención de la placa dentro de la solución, lo anterior según (Guillen Baca Yessy B, Montoya Portugal Alberto, Palo Tejada Ernesto, Pastor Rodriguez Ronald , S.f, págs. 4-5).

En cuanto a los equipos existentes en el mercado, se hallan modelos comerciales que permiten la configuración del tiempo de inmersión del sustrato, (Anonimo, Qualtech Products Industry, s.f, pág. 1) y número de ciclos de inmersión de la placa (Anonimo, Alibaba.com, s.f., pág. 1) Algunos equipos poseen puertos de comunicación USB, pantalla táctil o panel LCD (Anonimo, Nadetech Innovations, s.f., págs. 1-3), (Anonimo, Alibaba.com, s.f.), teclado y memoria adicional para almacenar los datos de los procesos realizados para su posterior procesamiento.

Otras funcionalidades halladas en los equipos comerciales incluyen el uso de un secador de recubrimientos en una cámara, función realizada por medio de una resistencia con control de temperatura. (Anonimo, holmarc Opto-Mechatronics Pvt , s.f., pág. 1). Así mismo, estas máquinas cuentan con carruseles de rotación que permiten realizar recubrimientos con múltiples capas. Sin embargo, el sistema de control de variables se encuentra en periodo de evaluación y diseño electrónico según lo validado con el autor (Guillen Baca Yessy B, Montoya Portugal Alberto, Palo Tejada Ernesto, Pastor Rodrigues Ronald , S.f, págs. 4-5).

Hasta el momento no se ha encontrado en la literatura revisada y reportada para esta propuesta el primer equipo que evite al operario los cálculos de velocidad de retiro para un espesor determinado, partiendo de otras variables más familiares y obtenidas de otros instrumentos como la viscosidad, la tensión superficial y la densidad del solvente. El presente trabajo se enfocará en la descripción e implementación de algoritmos para el cálculo automático de velocidad de retiro.

6. MARCO TEÓRICO

El proceso de recubrimiento por inmersión de piezas consiste en depositar sobre la superficie de un objeto llamado sustrato, una película de un espesor determinado que mejore las propiedades o cualidades de la superficie; como son el aspecto, adhesión, características de mojado, resistencia a la corrosión, desgaste o ralladuras, entre muchas otras. Por ejemplo, en procesos de impresión y fabricación de dispositivos semiconductores en los cuales el sustrato es un disco de material semiconductor, el recubrimiento es una parte esencial para la funcionalidad del producto terminado siendo esta otra utilidad de este procedimiento. De acuerdo a los diferentes tipos de condiciones ambientales que debe resistir el sustrato, según (Bolivar Osorio, 2007, pág. 30), se pueden clasificar de la siguiente manera: oxidación a altas temperaturas, corrosión catastrófica tipo I y corrosión catastrófica tipo II. La agresión producida por estos tipos de oxidación ocasiona la pérdida de algunas propiedades en el material que van desde el desprendimiento del mismo y rotura de las capas de óxido, hasta la pérdida de algunas propiedades mecánicas ocasionadas por los fenómenos de interdifusión.

Los tipos de recubrimiento utilizados en la actualidad se pueden clasificar en cuatro categorías principales (Bolivar Osorio, 2007, pág. 31) que se explicarán en los siguientes apartados.

- Recubrimientos de difusión.
- Recubrimientos sobre-capas (en inglés, *overlay coating*).
- Recubrimiento por barreras térmicas (TBC).
- Recubrimientos por depósito. Se hará énfasis en este tipo. Siendo el recubrimiento por inmersión una de estas técnicas utilizadas.

Los tipos de técnicas utilizadas para depositar películas delgadas se encuentran clasificadas de acuerdo con la fase del medio con el soluto. Mostrándolas en la siguiente figura:

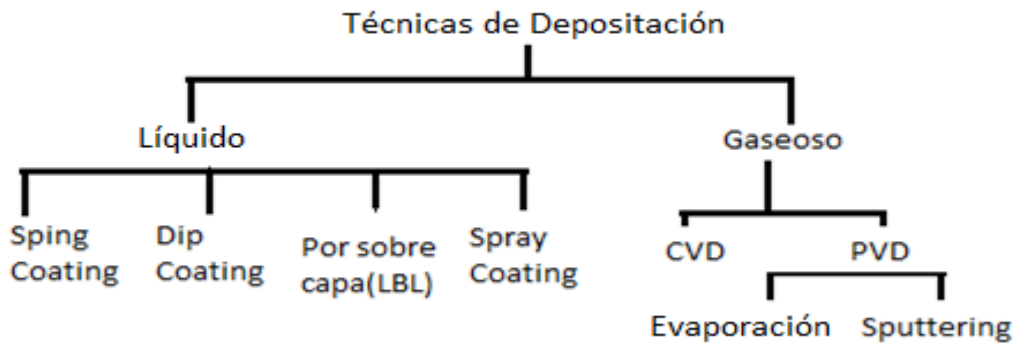


Figura 1. Síntesis esquemática de los métodos de recubrimiento a comparar. (Oliveria R M Alfredo, Zarbin C. J Aldo, 2005)

Los recubrimientos por difusión sobre-capas y barreras térmicas son utilizados generalmente para procesos industriales donde se recubren placas metálicas en serie. Los demás son métodos utilizados en experimentos de prototipo en laboratorios, donde se necesita recubrir placas pequeñas y ligeras, además de permitir mayor variedad de sustratos y precursores químicos.

6.1. Recubrimiento por difusión

Los recubrimientos por difusión se pueden formar cuando se produce la interdifusión, ya sea del material del recubrimiento hacia el interior del sustrato, o por la difusión desde el material base hacia el recubrimiento (Bolívar Osorio, 2007, pág. 31). Siempre que haya una mayor actividad particular del sustrato hacia el material se producirá el fenómeno de interdifusión. Si ocurre al contrario, se presentará el fenómeno de difusión “mezcla” del material hacia el sustrato. Por ejemplo en el documento (Bolívar Osorio, 2007, pág. 31), en el caso del depósito de aluminio sobre níquel, el aluminio se propaga más rápidamente hacia el níquel o metal base (interdifusión), siempre que haya una alta actividad particular del aluminio. Estos recubrimientos son depositados mediante el calentamiento de las piezas a tratar, las cuales se encuentran en contacto con el material del recubrimiento que puede estar en forma de polvo, o en una atmósfera constituida por un compuesto del material del recubrimiento.

6.2 Recubrimiento Sobre-Capa “Overlay”

“Estas capas, generalmente, se les conoce como recubrimientos MCrAl o MCrAlY, en donde “M” puede corresponder a hierro, níquel, cobalto o una combinación de níquel y cobalto. Estos tipos de recubrimientos tuvieron su origen directamente de los procesos de depósito con vapor y se diferencian de los recubrimientos por difusión en que no es necesaria una interdifusión entre el material del recubrimiento y el sustrato. (Bolívar Osorio, 2007, pág. 33)”. Con la implementación de estas técnicas de depósito se pueden obtener capas con composiciones totalmente independientes del sustrato.

Los equipos requeridos para depositar recubrimientos por sobre capa son aplicados, principalmente, por dos métodos: spray Plasma (*PS*) y depósito en fase vapor por haz de electrones (*EB-PVD*). También se utilizan otros métodos como: la técnica de llama a alta velocidad (*HVO*), depósito físico en fase de vapor (*PVD*) y la pulverización (*SD*). (Bolívar Osorio, 2007, pág. 33)

6.3. Recubrimiento por barreras térmicas

Los procesos realizados a través de la técnica TBC (*Thermal barrier coating*, por su designación en inglés) se caracterizan por tener capas muy delgadas, una baja conductividad térmica, una elevada resistencia al choque térmico y una alta concentración de poros internos que reducen la conductividad térmica a valores por debajo de los del material (Bolívar, 2007, p. 34).

Las TBC más comunes generalmente consisten en un sistema dúplex como el mostrado en la figura 1. Este sistema está constituido por una capa exterior, formada por un recubrimiento cerámico que actúa como aislante térmico y cuya función principal es la de reducir la transferencia de calor al sustrato metálico, y una capa interna metálica que sirve de unión entre el cerámico y el metal, cuya función principal es proporcionar al

metal una buena resistencia frente a la corrosión. La súper aleación u otro material que sea usado como sustrato le dan el soporte estructural al sistema formado por la TBC y el material base (Bolivar Osorio, 2007, pág. 34).

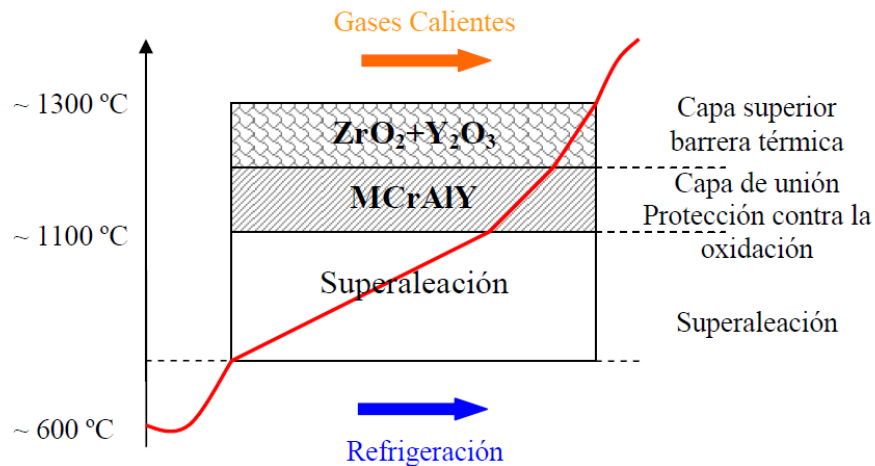


Figura 2. Esquema típico de una TBC vs. Gradiente de temperatura (Bolivar Osorio, 2007, pág. 34).

6.4. Recubrimiento por depósito en fase vapor: Estos se pueden dividir en dos grupos: Depósito física en fase vapor (PVD) y depositación químico en fase vapor (CVD).

6.4.1 Depósito físico en fase vapor (PVD)

Según lo documentado por el autor (Bolivar Osorio, 2007, pág. 37) Los procesos de PVD (en inglés. *physical vapor deposition*) son procesos de depósito atómica o molecular que implican por lo general la evaporación, pulverización o laceración del material depositado, además de su condensación sobre el sustrato llegando a obtener un recubrimiento de un rango de pocos nanómetros a cientos de nanómetros de espesor, siendo utilizado generalmente para realizar recubrimientos multicapa.

En la actualidad existen una gran variedad de procesos de PVD en los cuales generalmente se debe considerar lo siguiente según (Bolivar Osorio, 2007, pág. 37), el sistema debe contener bajos niveles de impurezas, una adecuada orientación y distancia de las placas al entrar el contacto con el recubrimiento, además del control y monitoreo del flujo de vapor,

presión de los gases y temperatura. Teniendo en cuenta todo lo anterior con el fin de que el proceso sea reproducible.

Las técnicas más usadas son:

6.4.1.1. Técnica de evaporación

“Esta técnica Se caracterizan por la evaporación del material de partida normalmente un material metálico, logrado por el calentamiento intenso de la pieza utilizando resistencias eléctricas o corrientes inducidas, láseres o por el bombardeo de un haz de electrones (EB-PVD, en inglés, *electron-beam*)”. (Bolívar Osorio, 2007, pág. 38).

6.4.1.2. Técnica de pulverización catódica (“Sputtering”)

En este método, los átomos que realizan el recubrimiento se obtienen mediante el bombardeo de unos blancos metálicos o cerámicos, con iones de un gas inerte (argón) a baja energía (500-1000 eV). Estos átomos que se desprenden del blanco, viajan hasta la superficie de la muestra y se depositan sobre la misma en proporciones similares a la del blanco. Este proceso también permite el depositación simultáneo de dos o más materiales (Bolívar Osorio,2007, p. 38).

Los procesos de pulverización catódica dicho por el autor (Bolívar Osorio, 2007, pág. 39) presentan el inconveniente de que son más lentos para su realización, que los procesos de evaporación además de presentar el calentamiento del sustrato, sin embargo, son más limpios, versátiles y permiten un mejor control de las variables que intervienen en el proceso como es el caso de la temperatura.

6.4.2. Depósito químico en fase de vapor (CVD)

La técnica CVD (en inglés, *Chemical Vapor Deposition*) según el autor (Gomez C, s.f, pág. 1) consiste en hacer reaccionar una mezcla de gases en el interior de una cámara de vacío (reactor), donde se encuentra la pieza a recubrir, produciendo la formación del revestimiento del material, dicha reacción que puede ser activada por diferentes vías (calor, luz, plasma, etc.) lo que conlleva a la formación de un producto sólido y estable. Dichas reacciones producidas dentro del reactor pueden ser de carácter homogéneo, en fase gaseosa, y/o heterogéneo, como puede observarse en la imagen la cámara cuenta con un

sistema de alta velocidad de bombeo para evacuar los subproductos de la reacción hacia el exterior.

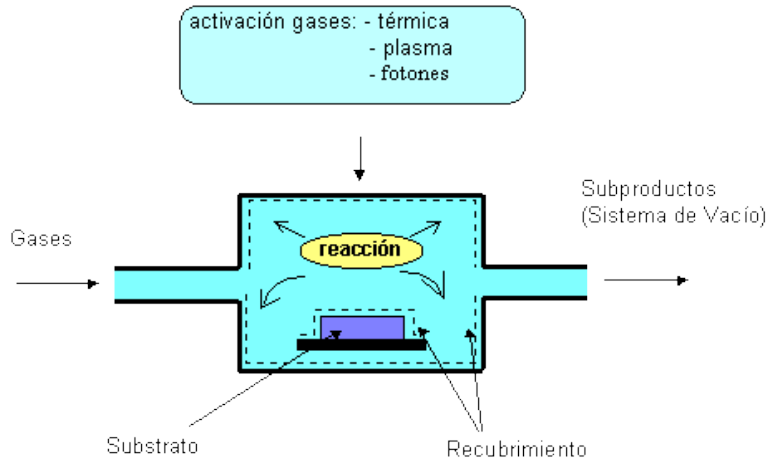


Figura 3. Deposición química en fase de vapor (CVD) (Gomez C, s.f, pág. 1)

Las técnicas más usadas son:

6.4.2.1. Depósito químico por proceso de inmersión (dip coating)

De acuerdo con L. E. Scriven (Scriven, 1988, p.121), es un método clásico y simple de depositar en un sustrato, especialmente losas pequeñas y cilindros, una película líquida uniforme y delgada para su solidificación en un recubrimiento. Esta técnica consiste en sumergir y extraer el material a recubrir en un solvente adecuado de manera controlada con el fin de mantener el flujo del líquido constante, efecto que se implica el control del grosor y calidad del recubrimiento, de acuerdo con los parámetros del fluido, como son la fuerza viscosa, tensión superficial (capilaridad) y gravedad. Este método es utilizado generalmente para realizar pruebas de laboratorio en áreas de investigación y desarrollo, debido a que permite la deposición de gran variedad de líquidos y sustratos siendo generalmente usado para recubrir piezas pequeñas en forma no serial.

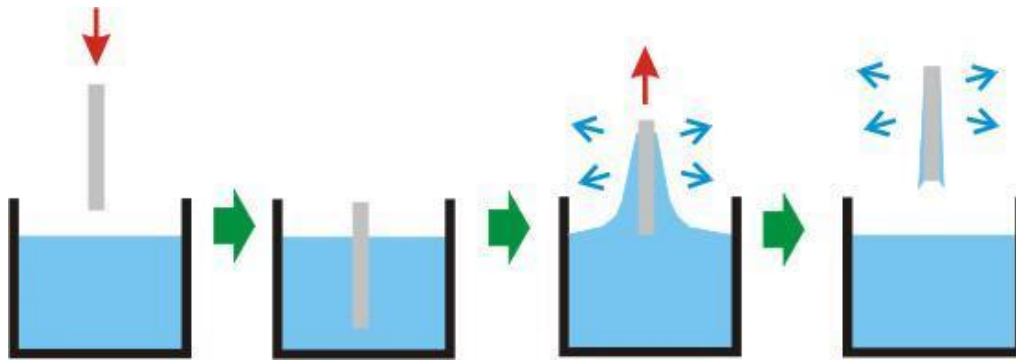


Figura 4. Fases del proceso de recubrimiento por inmersión. (Anónimo, Sf, pág. 1)

El proceso de depositación por inmersión se divide en cinco etapas según (Anónimo, Sf, pág. 2):

Inmersión: El sustrato se sumerge en una solución que lo revista a una velocidad constante y controlada.

Puesta en Marcha: El sustrato se mantiene dentro de la solución durante un determinado tiempo sin ser aun levantado.

Depósito: para realizar depósitos de capas delgadas sobre el material, se debe sacar el material del sustrato a una velocidad contante, sin vibraciones que alteren el proceso y lentamente, debido a que la velocidad de retirada determina el espesor del recubrimiento además del goteo de la misma.

Drenaje: Se retira el exceso de líquido de la superficie.

Evaporación: consiste en la evaporación del líquido en el sustrato para los disolventes

volátiles como alcoholes. La evaporación empieza durante las etapas de depositación y drenaje, de lo contrario el proceso continúa paso por paso.

6.4.2.2. Depositación química por proceso de centrifugado (spin coating)

Es una técnica utilizada recientemente para recubrir sustratos, generalmente pequeños discos planos, con la cual se logran películas finas y uniformes. Este método consiste en el depósito de un líquido en el centro del sustrato y hacer que éste gire a un determinado número de revoluciones por segundo (rpm).

Según L. E. Scriven (Scriven, 1988, p.1), esta técnica tiene el problema que en ocasiones el drenaje radial es inestable debido a las fuerzas que intervienen en el proceso, como son la tensión superficial y la fuerza centrífuga, presentando un recubrimiento más delgado en los extremos del disco que en su origen. Este inconveniente puede ser solucionado con la utilización de disolventes volátiles que permiten lograr capas delgadas y sólidas en menor tiempo, pudiendo predecir teóricamente el espesor y la uniformidad del material, siempre y cuando se tengan en cuenta las variables que intervienen en el sistema como son la velocidad, las condiciones del gas y la reología de la concentración utilizada para solidificar el líquido. Debido a las particularidades de esta técnica, este tipo de recubrimiento funciona mejor para recubrir materiales usados en microelectrónica.

6.4.2.3 Depositación química por aerosol (spray coating)

El recubrimiento por aerosol es una técnica utilizada para hacer recubrimientos delgados y multicapa, la cual consiste en recubrir la pieza o el sustrato utilizando gases comprimidos, los cuales se deben aplicar a una determinada temperatura, presión, distancia y flujo controlados. Dicho gas inicialmente se encuentra en fase líquida para poder realizar correctamente la mezcla entre el soluto y el solvente, siendo necesario pasarla más tarde a fase gaseosa para su posterior aplicación sobre la pieza teniendo que recurrir en ocasiones a adicionar otras sustancias más volátiles para el proceso.

6.5 Comparación de los métodos de recubrimiento por depósito

Característica	PVD	<i>Sputtering</i>	CVD	<i>Dip Coating</i>	<i>Spin Coating</i>	<i>Spray Coating</i>
Costo	Alto	Alto	Medio	Bajo	Bajo	Bajo
Dificultad en preparación del sustrato	Alto	Alto	Baja	Baja	Baja	Bajo
Tiempo de recubrimiento	Bajo	Alto	Bajo	Medio	bajo	Medio
Uniformidad	Media		Alta	Media	Baja	Media
Espesor del recubrimiento	Delgado	Delgado	Grueso	Ambos	Delgado	Delgado
Flexibilidad de precursores químicos	Algunos	Pocos	Algunos	Alto	Media	Pocos
Variedad de recubrimientos logrados	Pocos	Pocos	Pocos	Muy Alta	Alta	Bajo

Variedad de sustratos utilizados	Si	Pocos	Pocos	Alto	Medio	Bajo
Generación de residuos tóxicos	Si	Si	Si	Si	Si	Si
Hermeticidad del equipo	Si	Si	Sí	No	No	Si
Temperatura necesaria	Alta	Alta	Alta	Media	Media	Alta
Eficiencia del equipo	Media	Baja	Baja	Alto	Media	Bajo
Consumo de potencia	Alta	Alta	Muy alto	Bajo	Bajo	Medio
Permite realizar múltiples capas	Si	Si	Se dificulta	Si	Se dificulta	Si
Capacitación del operario	Muy alta	Muy alta	Muy alta	Baja	Alta	Media

6.6 Modelamiento hidrodinámico del proceso de recubrimiento por inmersión

El modelo matemático utilizado para definir el recubrimiento mediante la técnica de depositación por inmersión puede ser analizado a partir del artículo de los autores (Brasjen Bj- A.W Van Cuigk-A.A, 2010, págs. 1-2) donde se explica teóricamente el uso de las ecuaciones descubiertas por Landau y Levich en 1942, las cuales están basadas en las ecuaciones de hidrodinámica de fluidos Newtonianos siempre y cuando el número de capilaridad esté dado por:

$$Ca = \frac{\mu v}{\sigma} = \frac{\text{Fuerzas Viscosas}}{\text{Fuerzas capilares}} \ll 1 \quad (1)$$

De la ecuación anterior se puede analizar que el número de capilaridad está íntimamente relacionada con la μ viscosidad, v velocidad del sistema y la σ tensión superficial, siendo esta última inversamente proporcional a las dos anteriores.

Suponiendo que el drenaje gravitacional es despreciable, según (Hsu Wei Fang-Kuo Yen Li- Tai Lun Su-Thomas Chun Kuang Yang- Ji Sheng Chang- Po Liang Lin-, 2008, pág. 1), el espesor viene dado por la siguiente relación proporcional entre el mismo y la velocidad:

$$h_0 \propto v^x \quad (2)$$

De la cual se define que h_0 representa el espesor o ancho del recubrimiento, v es la velocidad del sistema y x se encuentra definida en un rango entre 0.53 y 0.64 para un caso en particular utilizando una solución de polímeros, según Landau y Levich. Es importante aclarar que el valor de esta constante puede variar dependiendo de los materiales a utilizar en el experimento.

La relación definida en (2) tiene la siguiente aproximación según (Brasjen Bj- A.W Van Cuigk-A.A, 2010, pág. 1):

$$h_0 = 0.946 I_a C_a^{2/3} \quad (3)$$

Con la ecuación (3) se define la longitud capilar I_a la cual está en función de la tensión superficial (σ), densidad (ρ) y gravedad (g), pudiéndose encontrar en un rango de 1-3 mm respectivamente. Se expresa de la siguiente manera:

$$I_a = \sqrt{\frac{\sigma}{\rho g}} \quad (4)$$

Una extensión del análisis realizado por Landau y Levich fue elaborada por los autores (Brasjen Bj- A.W Van Cuigk-A.A, 2010, pág. 1) donde se tomó valores más grandes de Ca ,

llegando a modificar la ecuación (3) determinando el espesor del recubrimiento la siguiente forma:

$$h_0 = I_a(0.946 C_a^{\frac{2}{3}} - 0.107C_a) \quad (5)$$

Dado un espesor deseado h_0 , se usa la ecuación (3) o (5) dependiendo del resultado de la C_a “capilaridad”, para posteriormente realizar el cálculo de la velocidad de retiro despejándola de la ecuación (1) o al contrario si fuese necesario.

7. OBJETIVOS

7.1 Objetivo general

Diseñar y construir un equipo para realizar el proceso de recubrimiento por inmersión.

7.2 Objetivos específicos

- Realizar un diseño mecánico del dispositivo teniendo en cuenta aspectos financieros y la fenomenología hidrodinámica involucrada en los procesos.
- Diseñar y construir los circuitos electrónicos de manejo del motor paso a paso y funcionamiento de los sensores de temperatura y humedad
- Desarrollar e implementar el sistema automático de control de velocidad y tiempo de contacto del material con la solución.
- Realizar pruebas de desempeño del sistema de inmersión para el recubrimiento de diversas láminas.
- Realizar el manual de funcionamiento del equipo con el fin de brindar una información completa al usuario sobre su funcionamiento y operación, así como la guía de solución de problemas.

8. JUSTIFICACIÓN Y BENEFICIOS

Este trabajo nace como apoyo a la tesis doctoral del ingeniero Andrés Felipe Betancur, la cual propone investigar sobre el depósito de grafeno sobre superficies metálicas de carácter comercial, para lo cual se hace necesario implementar y fabricar un equipo que desarrolle este proceso de la manera más óptima.

Este equipo será utilizado conjuntamente por el Grupo de investigación de Óptica y Espectroscopía y estará disponible para otros grupos de la universidad o la región que requieran de su utilización, para brindar un control adecuado del espesor y uniformidad a los recubrimientos, siendo importante optimizar los productos de materiales nanoestructurados de base carbono, tanto para la corrosión como para la conducción analizando a profundidad los siguientes dos casos. En corrosión, se evaluará la capacidad protectora de las nanoestructuras de carbono y su eficiencia en la protección de diferentes metales. En conducción, se realizarán recubrimientos conductores o aislantes con base en una solución de nanoestructuras de carbono y polianilina.

9. ALCANCES

Se propone la presentación de los resultados obtenidos mediante una ponencia de tipo oral o *poster*, en un evento académico de carácter local, regional o nacional.

10. METODOLOGÍA

En la formulación del proyecto, se efectuó la búsqueda bibliográfica en tres temas: el modelamiento hidrodinámico del proceso de *dip coating*, la selección de componentes mecánicos para el movimiento del sustrato y la selección de componentes electrónicos para el sistema embebido de control.

Entre los sistemas mecánicos que existen para subir y bajar objetos pequeños, se encuentran:

- Uso de tambor y poleas
- Uso de poleas y bandas transportadoras
- Uso de piñón y cremallera corrediza
- Uso de un tornillo sin fin y guías

El último mencionado ofrece el mecanismo de reducción de velocidad más efectivo, por lo que es el más visto en los prototipos y equipos comerciales. Éste es el que se usará. En el

mismo, el tornillo se rota mediante un motor paso a paso y está acoplado a una o más guías para evitar los momentos angulares y las vibraciones (ver Figura 1).

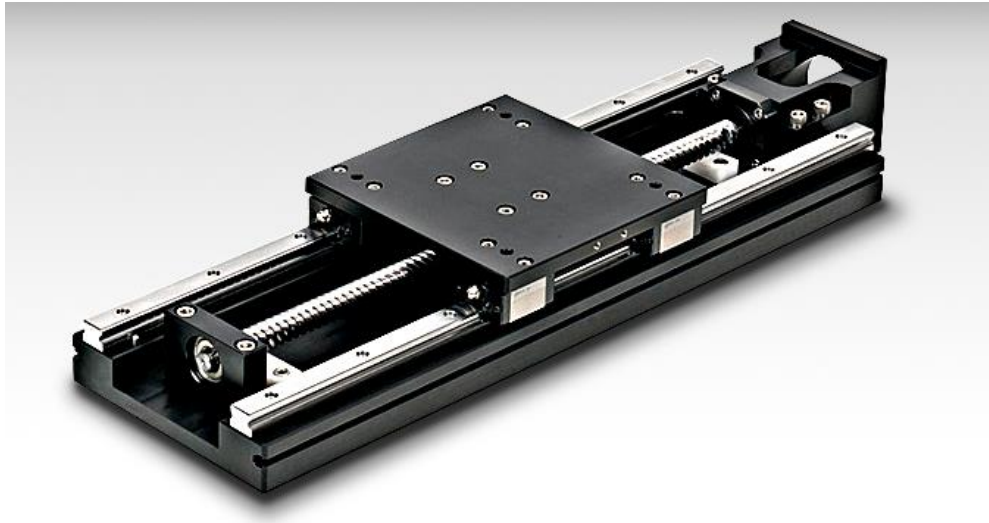


Figura 5. Detalle del deslizamiento de objetos mediante tornillo de bolas y motor paso a paso. Fuente: Misumi USA, Inc. [En línea] Disponible en: <http://blog.misumiusa.com/wp-content/uploads/2014/06/ActuatorBlogImage.2014.06.03.jpg>. [consulta: 07 Abr 2015]

Se validó el diseño mecánico inicial a través de asesorías de expertos en diseño de componentes mecánicos, y se refinará durante la ejecución del proyecto, la cual agregará la selección de guías y piezas al final del tornillo sin fin para hacer mínima la influencia de las vibraciones.

En cuanto al desarrollo del diseño mecánico, se realizaron diversos cálculos físicos necesarios para la selección del tornillo de bolas y del motor paso a paso, teniendo en cuenta los siguientes requerimientos funcionales:

Poner la referencia: Oriental Motor USA, Technical Reference 2012/2013. http://www.orientalmotor.com/products/pdfs/2012-2013/G/usa_tech_calculation.pdf

- Velocidad mínima de retiro: 50 μm por minuto
- Peso del sustrato y pinza de sujeción: 300 g

Se seleccionaron referencias concretas de las siguientes especificaciones:

- Tornillo de bolas Misumi BSST1510-600: Paso de 10 mm, diámetro de 15 mm y precisión de categoría C7 (desviación máxima de 50µm para 300 mm).
- Motor paso a paso Sanyo Denki SH1603-0440: Paso mínimo de 1.8° y torque superior a 0.4 oz-in.

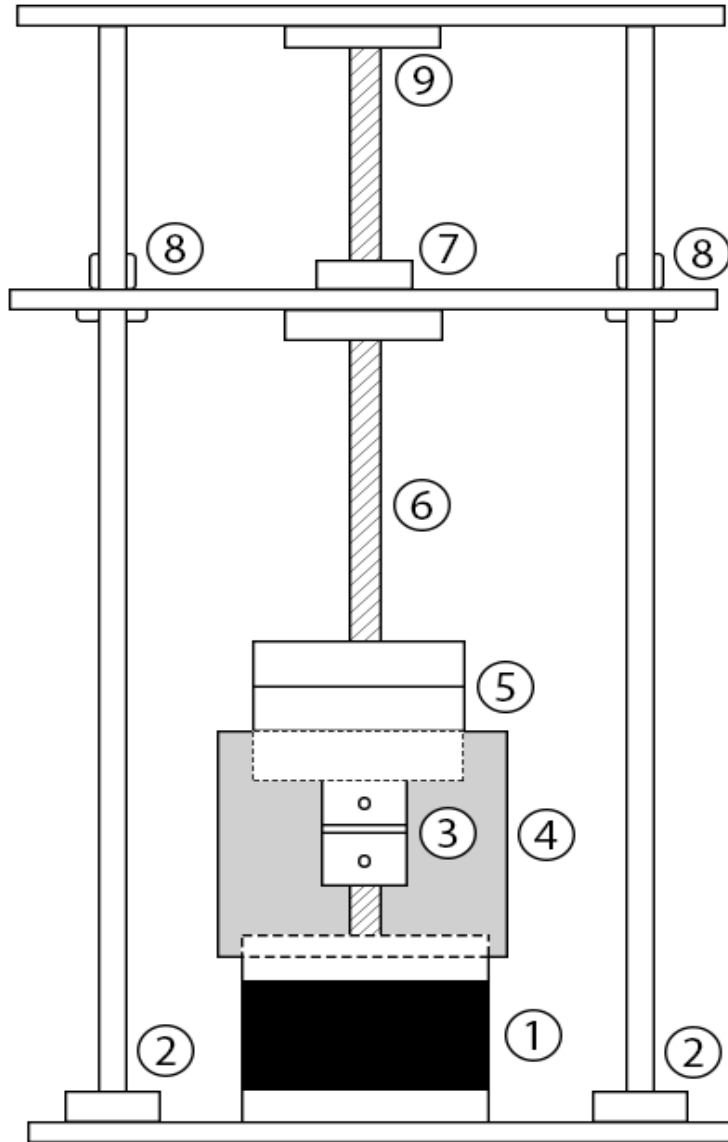


Figura 6. Boceto no técnico preliminar del diseño mecánico.

Fuente: Ilustración elaborada por M. D. Correa.

	Descripción	Parte
--	-------------	-------

1	Motor paso a paso	Sanyo Denki SH1603-0440
2	Guía con soporte	No aplica
3	Acople	Misumi MCGLRK25
4	Cabecera del motor	Misumi BJS12A
5	Soporte móvil del tornillo de bolas	Misumi BRW12
6	Tornillo de bolas	Misumi BSST1510-600
7	Nuez del tornillo de bolas	Incluida con el tornillo
8	Cojinete lineal	No aplica
9	Soporte fijo del tornillo de bolas	Misumi BRW12

Respecto a la implementación electrónica, se realizará una maqueta para subir y bajar una carga que permita plantear las rutinas que deberá llevar a cabo el microcontrolador Piccolo F28027F de Texas Instruments.

Este microcontrolador se destaca por su ajuste fino de la modulación por ancho de pulsos y capacidad de efectuar cálculos rápidos de potenciación y raíces mediante un DSP interno. También cuenta con las siguientes características de interés, de acuerdo con los objetivos de este trabajo:

- Modulación por ancho de pulso con una resolución de 300 ps y cuatro canales disponibles
- Conversor analógico a digital, con una resolución de 12 bits con una tasa de conversión de 4.6 millones de muestras por segundo
- Comparadores con conversores digitales a analógicos, con umbral ajustable a palabras de 10 bits
- Comunicación con otros periféricos por IIC y SPI
- Programación por puerto USB

- Simulación y depuración vía JTAG
- Comunicación UART vía USB
- Control de pantalla LCD clásica mediante cuatro pines

Así, se procederá a diseñar los circuitos para manejar el motor paso a paso y obtener la información de los sensores de temperatura (uno para el líquido y otro ambiental) y el de humedad. Luego, se elaborarán los algoritmos de interacción del microcontrolador con estos circuitos en el lenguaje C. Se ensayarán primero con la maqueta y luego con el equipo real. Posteriormente, se implementará una interfaz para el usuario en Java que permita el manejo del prototipo mediante el puerto USB-Uart. Finalmente, se seleccionará el tipo de control automático apropiado, el cual tendrá en cuenta las expresiones matemáticas (1), (3) y (5) descritas en el modelo hidrodinámico.

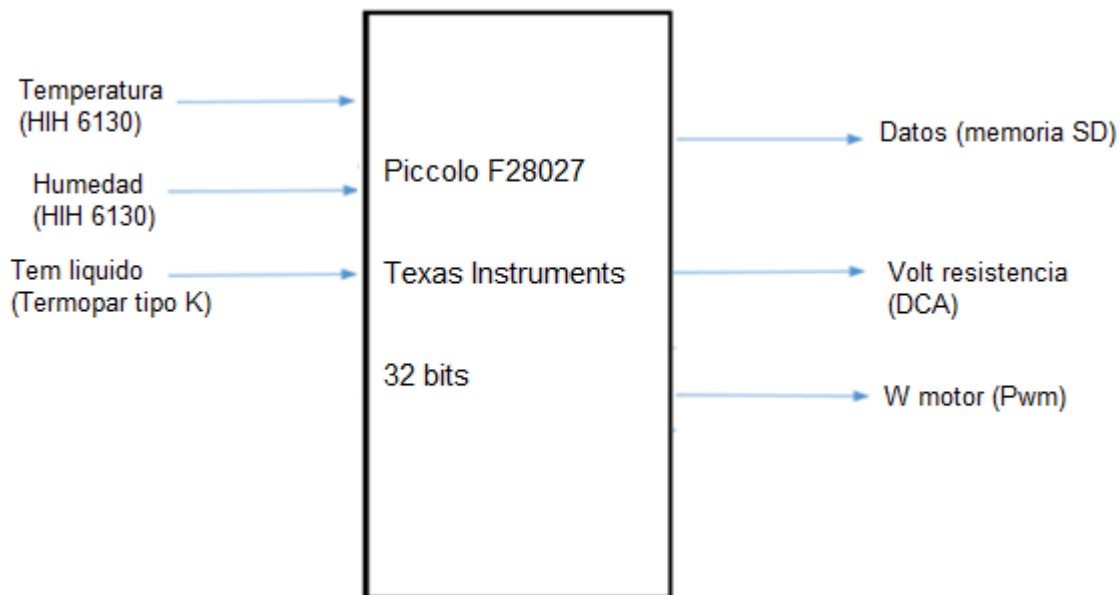


Figura 7. Disposición de entradas y salidas del sistema de control.

Al finalizar la implementación del prototipo, se realizarán ensayos de desempeño con el fin de corregir los inconvenientes que puedan surgir y garantizar la repetitividad en los recubrimientos.

11. CONTENIDO PRELIMINAR

Tabla de contenido

Lista de tablas

Lista de figuras

Lista de anexos

Glosario

Lista de símbolos y variables

Resumen

Introducción

Principio físico-matemático de la técnica de depositación por inmersión.

Diseño mecánico del equipo

Diseño electrónico de equipo

Resultados

Conclusiones

Recomendaciones

Bibliografía

Anexos

- Anexo 1 - Artículo publicable
- Anexo 2 - Diario de campo
- Anexo 3 - Anteproyecto
- Anexo 4 - Manual de funcionamiento del equipo

12. PRODUCTOS ESPERADOS.

- Diseño físico del equipo para realizar el proceso de recubrimiento por inmersión.
- Diagramas de esquemáticos y circuitos impreso del equipo
- Equipo funcional
- CD-ROM para entregar en la biblioteca
- Manuales de manejo y cuidados del dispositivo

- Informe de actividades realizadas durante el trabajo grado.

13. PRESUPUESTO Y RECURSOS NECESARIOS

Item	Participación		Desembolso	
	Estudiante	Servicios UPB	Si	No
Internet	40.000			X
Fotocopias	5.000			X
Transporte	50.000			X
Sala de Espectroscopía en el laboratorio GOE		4.200.000		X
Osciloscopio Tektronix TDS2014D		9.500.000		x
Computador de mesa		1.500.000		
Horas estudiante (480 horas x \$ 10.000)		4.800.000		X
Horas director (100 horas x \$ 50.000)		5.000.000		X

Horas codirector (200 horas x \$25.000)		5.000.000		X
Horas asesor (50 horas x 25.000)		1.250.000		X
Insumos de componentes electrónicos		800.000	X	
Fuente DC de bajo rizado		881.600	X	
Piezas mecánicas de precisión		5.000.000	X	
Circuitos impresos		500.000	X	
Subtotales	95.000	38.431.600		
Imprevistos (10%)	59.500	3.843.160		
Totales	154.500	42.274.760		

14. FINANCIACION

Todos los materiales, equipos e insumos a usar que implican desembolso están contemplados en la ejecución presupuestal del proyecto interno CIDI “Establecimiento de condiciones óptimas para el depósito de grafenos sobre superficies metálicas” con radicado 102B-02/13-27.

15. CRONOGRAMA DE ACTIVIDADES.

Actividad / Mes	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
Formulación de anteproyecto												
Implementación de maqueta												
Implementación de diseño mecánico definitivo												
Elaboración de circuitos electrónicos del sistema												
Elaboración de algoritmos control electrónico del sistema												
Ensayos de inmersión y retiro de láminas												
Optimización del equipo												
Redacción del libro												

16. REFERENCIAS PRELIMINARES

- Anónimo. (s.f.). *Alibaba.com*. Obtenido de http://www.alibaba.com/product-detail/Precision-Dip-Coater_114025707.html.
- Anónimo. (s.f.). *holmarc Opto-Mechatronics Pvt* . Obtenido de http://www.holmarc.com/dip_coating_unit_IR.php
- Anónimo. (s.f.). *Nadotech Innovations*. Obtenido de <http://nadotech.com/documents/Datasheet%20ND-DC%20ES.pdf>
- Anónimo. (Sf). *Nadotech innovations*. Obtenido de <http://www.nadotech.com/index.php/es/tecnologias>
- Anónimo. (s.f.). *Qualtech Products Industry*. Obtenido de <http://www.qualtechproductsindustry.com/products/cppllication-series-instruments/dip-coater/>
- Anónimo. (Sf). *Texas Instruments*. Obtenido de <http://www.ti.com/sitesearch/docs/universalsearch.tsp?searchTerm=f28027f#linkId=1>
- Arenas Gaviria, Bernardo . (2012). *Dinámica de un cuerpo libre* . Obtenido de http://aprendeenlinea.udea.edu.co/lms/moodle/file.php/801/textcuerporigido_2012_.pdf
- Bolívar Osorio, F. J. (2007). *Universidad Complutense de Madrid*. Obtenido de <http://eprints.ucm.es/7551/1/T29978.pdf>
- Brasjen Bj- A.W Van Cuigk-A.A. (2010). Dip coating chemically patterned surfaces. *Elsevier*, 1-3.
- Brow, Leman Bursten, Murphy. (2009). *Química La Ciencia Central*. México: Pearson Prentice Hall.
- bursten, B. L. (2009). *Química la ciencia central*.
- Cabrera Vásquez Adan, UDLAP bibliotecas. (2004). UDLAP. Obtenido de http://catarina.udlap.mx/u_dl_a/tales/documentos/lim/cabrera_v_a/capitulo3.pdf
- Castro, G. (2009). *Departamento de ingeniería Mecánica F.I.U.B.A.* Obtenido de http://campus.fi.uba.ar/file.php/295/Material_Complementario/Aceros.pdf
- Chang Reimod, College Williams . (2002). *Química*. México D.F: Mc Graw Hill.
- E.P.A. United States Environmental Protection Agency. (Abril de 1996). *E.P.A. United States Environmental Protection Agency*. Obtenido de <http://www.clu-in.org/download/remed/spanthde.pdf>
- Escuela Americana El Salvador. (N.R). *Concentración de soluciones*. Obtenido de <http://www.amschool.edu.sv/paes/science/concentracion.htm>

- Fernández, A. C. (Oct de 2007). *Propiedades físicas de grafeno en presencia de desorden topológico*. Obtenido de Universidad Carlos III de Madrid: http://e-archivo.uc3m.es/bitstream/handle/10016/2370/Tesis_AlbertoCortijo.pdf?sequence=6
- Gómez C, A. (s.f). *Técnicas de deposición física en fase vapor (CVD)*. Obtenido de <http://www.icmm.csic.es/fis/espa/cvd.html>
- González, M. (2011). *Universidad Carlos III de Madrid*. Obtenido de <http://ocw.uc3m.es/cursos-cero/principios-basicos-de-Química/unidad-2/unidad-2>
- Guillen Baca Yessy B, Montoya Portugal Alberto, Palo Tejada Ernesto, Pastor Rodríguez Ronald . (S.f). *Perú Solar org*. Obtenido de <http://www.perusolar.org/include/simposios/archivos16/ponencias16/dip%20coating%20DISEÑO%20Y%20CONSTRUCCIÓN%20DE%20UN%20EQUIPO%20PARA%20DEPOSICIÓN%20DE%20PELICULAS%20DELGADAS.pdf>
- Hole, N. Z. (2009). *Introducción a la mecánica*. N.r: Universidad de Chile. Obtenido de http://www.profesorenlinea.cl/fisica/Fuerzas_Torque_momento.html
- Hsu Wei Fang-Kuo Yen Li- Tai Lun Su-Thomas Chun Kuang Yang- Ji Sheng Chang- Po Liang Lin-. (16 de 04 de 2008). Dip coating assisted polylactic acid deposition on steel surface: film thickness affected by drag force and gravity. *El Sevier*, 1-3. doi:10.1016/j.matlet.2008.04.046
- Ibrahim, V. z. (4 de 12 de 2012). *Blog forma protección contra la corrosión*. Obtenido de <http://formaproteccioncontralacorrosion.blogspot.com/2012/12/v-behaviorurldefaultvmlo.html>
- Jin Tang Wu,Hsin Chun Lai, Sen Yeu Yang, Tzu Chien Huang, Sheng Hsin Wu. (2010). *Elsevier*. Obtenido de Doi:10.1016/j.mee.2010.01.003
- L.A. Pavés F, J.E. Jiménez C,E Ramos M. (2 de 08 de 2009). *Física 3 Texto para el estudiante*. . Chile: McGraw Hill.
- More, F. H. (1981). *Fundamentos de Química* . Prentice Hall Hispanoamérica S.a.
- Moreno, V. M. (2007). *Universidad Carlos III de Madrid*. Obtenido de http://e-archivo.uc3m.es/bitstream/handle/10016/1161/pfc_lecho_fluidizado.pdf?sequence=1
- Nisbett, R. G. (2008). *Diseño en ingeniería mecánica* (Octava ed.). México : McGraw Hill. Obtenido de <https://www.dropbox.com/s/w3py0kxed7b13n4/Dise%C3%B1o%20en%20Ingenier%C3%ADa%20Mec%C3%A1nica%20de%20Shigley%20-%208%20Edici%C3%B3n%20-%20Budynas.pdf?dl=0>

- Oliveria R M Alfredo, Zarbin C. J Aldo. (2005). Um procedimento simples e barato para construcao de um equipamento "Dip coating" para deposicao del filmes em laboratorio. *Quim Nova*, 141-144.
- Osinermin. Organismo supervisor de la inversión en energía y minería. (Octubre de 2008). *Temario 3. Libro de física general*. Obtenido de http://www.osinermin.gob.pe/newweb/pages/Publico/LV_files/Manual_Fisica_General.pdf
- Osorio, F. J. (2007). *Universidad Complutense de Madrid*. Obtenido de <http://eprints.ucm.es/7551/1/T29978.pdf>
- Philips. S, J. (2000). *Química conceptos y aplicaciones*. México D.F: McGraw-Hill.
- Raimont A. Serway, Jhon W. Jewett. (21 de abril de 2008). *Física para ciencias e ingeniería*. México D.F: Cengage learning editores S.a .
- Riveiro, L. C. (2008). *Reologia de productos alimentarios*. Santiago de Compostela: Universidad de Santiago de Compostela. Obtenido de http://www.slideshare.net/FHEDRA/reologia-de-los-productos-alimenticios-de-unadm-p-tarea?qid=1d95bc56-6994-4159-a11c-6dfa96780d29&v=qf1&b=&from_search=4
- Sánchez Echeverría Judith ,García Becerril María, Balderas Solano Yolanda. (2008). *Química 1*. Nr: Nr. Obtenido de Átomo: <http://definicion.de/atomo/>
- Scriven, L. E. Department of Chemical Engineering, & Materials Science and Center for Interfacial Engineering, University of Minnesota. (1988). Mrs online proceedings library. Obtenido de <http://dx.doi.org/10.1557/PROC-121-717>
- Shackelford, J.F. (1992). *Introducción a la ciencia de materiales para ingenieros*.6ta. Prentice Hall. Obtenido de <http://materias.fi.uba.ar/6713/difusion-repaso.pdf>.
- TLV Compañía especialista en vapor. (2015). *TLV*. Obtenido de <http://www.tlv.com/global/LA/steam-theory/types-of-steam.html>
- Universidad de Granada, Departamento de electrónica y tecnología de computadores. (N.R de N.R). *Departamento de electrónica y tecnología de computadores.Crecimiento de semiconductores* . Obtenido de <http://electronica.ugr.es/~amroldan/deyte/cap09.htm>
- Universidad de valencia. (2009-2010). *Química Física Avanzada cuarto curso* . Obtenido de http://ocw.uv.es/ciencias/3-2/tema_6_tens_sup.pdf
- Universidad Pontificia Bolivariana, Facultad de Ingeniería Mecánica. (2008). *Definiciones básicas curso mecanismos 1*. Medellín Colombia.

Urien Pinedo, A. (septiembre de 2013). *Obtencion de biocarbones y biocombustibles mediante pirolisis de biomasa residual*. Obtenido de http://digital.csic.es/bitstream/10261/80225/1/BIOCARBONES_CENIM_CSIC.pdf

Velarde, M.G ; Starov, Victor. (2009). Wetting concepts and basic features. *Espanola de Física*, 23, 61-74.

Witascheck, Catrin Strobel Anke, Kadow-Romacker Tom. (2011). Evaluation of process parameter of an automated Dip coating. *El Sevier*, 1-2. Obtenido de doi 10.1016/j.matlet.2011.07.102

Zavaleta. Bravo, H (11 de 2007). Instituto Politécnico Nacional. Obtenido de <http://itzamna.bnct.ipn.mx:8080/dspace/bitstream/123456789/7136/1/ZAVALETA.pdf>

17. PROPIEDAD INTELECTUAL Y DESTINACIÓN DEL PROYECTO

 <p>Universidad Pontificia Bolivariana</p>		<p>ACTA DE PROPIEDAD INTELECTUAL TRABAJOS DE GRADO</p>	<p>VICERRECTORÍA ACADÉMICA</p>
--	--	---	---

FECHA: Junio 17 de 2015

1. NOMBRE DEL PROYECTO: Diseño e implementación de un dispositivo de control para realizar el proceso de recubrimiento por inmersión.

2. PARTICIPANTES

2.1. ESTUDIANTE

Nombre: Lina Marcela Ruiz Marín

No. c.c.: 42.691.346

Horas dedicadas al proyecto: 500

Programa en el cual se encuentra matriculado: Ingeniería Electrónica

2.2. DIRECTOR

Nombre: Freddy Rafael Pérez

No. c.c.: 15.043.436 de Sahagún

Empresa donde labora actualmente: Universidad Pontificia Bolivariana

Cargo en la empresa: Jefe de Ciencia Básica

Horas dedicadas al proyecto: 50

2.3. CODIRECTOR

Nombre: Maryam del Mar Correa Gómez

No. c.c.: 1.017.152.137 de Medellín

Empresa donde labora actualmente: Universidad Pontificia Bolivariana

Cargo en la empresa: Joven Investigador

Horas dedicadas al proyecto: 200

3. FUENTES DE FINANCIACIÓN

Fuentes de financiación	Ítem	Aporte en efectivo	Aporte en especie	Total por fuente
--------------------------------	-------------	---------------------------	--------------------------	-------------------------

				de financiación
UPB	Sala de espectroscopía en el laboratorio GOE	4.200.000		
	Osciloscopio Tektronix TDS2014D	9.500.000		
	Computador de mesa	1.500.000		
	Horas director (100 horas x \$ 50.000)	5.000.000		
	Horas codirector (200 horas x \$25.000)	5.000.000		
	Horas asesor (50 horas x 25.000)	1.250.000		
	Insumos de componentes electrónicos	800.000		
	Fuente DC de bajo rizado	881.600		
	Piezas mecánicas de precisión	5.000.000		
	Circuitos impresos	500.000		
Estudiante	Internet	40.000		
	Fotocopias	5.000		
	Transporte	50.000		
	Horas estudiante (480 horas x \$ 10.000)	4.800.000		

Total del proyecto			38.526.600
---------------------------	--	--	-------------------

4. PRODUCTOS

Los productos a entregar son:

- 4.1 Diseño físico y mecánico del equipo para realizar el proceso de recubrimiento por inmersión.
- 4.2 Diagramas de esquema electrónico y circuito impreso del equipo.
- 4.3 Entrega del equipo funcionando.
- 4.4 Cd-Rom con el trabajo de grado en versión digital y software utilizado para su implementación
- 4.5 Manuales de uso y cuidados del equipo.
- 4.6 Informe de actividades realizadas durante el trabajo grado.
- 4.7 Artículo publicable.

5. EQUIPOS Y MATERIAL BIBLIOGRÁFICO

La estudiante, director y asesores, se comprometen a devolver a la Universidad Pontificia Bolivariana todo el equipo y material bibliográfico, obtenido con recursos proporcionados por las dependencias académicas y administrativas de la Universidad, una vez termine el proyecto.

6. DURACIÓN DEL PROYECTO

El proyecto tendrá una duración de 12 Meses, contados a partir del 17 de Mayo de 2015.

7. PROPIEDAD INTELECTUAL

7.1. DERECHOS MORALES

Los derechos morales de autor corresponden a los estudiantes, al director y a toda persona que a criterio de éstos, haga aportes originales intelectuales en los avances y en el resultado final del proyecto.

En cualquier tipo de divulgación se dará crédito a los autores y la Universidad Pontificia Bolivariana.

7.2. DERECHOS PATRIMONIALES

Los derechos sobre los resultados derivados del presente trabajo de grado se rigen por el Estatuto de Propiedad Intelectual de la Universidad.

8. CONFIDENCIALIDAD

Los participantes se comprometen a guardar absoluta reserva sobre la información confidencial que conozcan o a la que tengan acceso en virtud de su participación en el presente proyecto. Igualmente se comprometen a guardar reserva sobre los resultados innovadores con potencial de aprovechamiento comercial del proyecto.

Igualmente, están en la obligación de no copiar, enviar por fax, reproducir, divulgar, o distribuir el o los documentos, ni en su totalidad o parte, sin el expreso consentimiento por escrito de la UPB.

Acepto que la violación de la reserva a lo aquí convenido dará lugar a la aplicación de las sanciones previstas en el Código Penal, artículo 308 de la Ley 599 del 2000, referente a la Violación de reserva industrial o comercial, sin perjuicio de las que se imponen por competencia desleal y del cobro de las indemnizaciones a que haya lugar.

9. SOLUCIÓN DE CONFLICTOS

En caso de controversia sobre la interpretación o aplicación de la presente acta, se agotarán en primera instancia los conductos regulares de la Universidad, y en caso de no llegar a ningún acuerdo, se dirimirá el conflicto mediante dos (2) amigables componedores designados por cada parte. La decisión será en equidad y tendrá fuerza vinculante para los intervinientes.

10. CONSTANCIA

Todos los partícipes declaran conocer el Estatuto de Propiedad Intelectual de la Universidad Pontificia Bolivariana.

En caso que algún participante se retire antes del 80% de ejecución del cronograma del proyecto, perderá todos los derechos sobre los resultados de la misma. En caso de presentarse alguna circunstancia que altere los términos de la presente acta, deberá anexarse al presente documento la respectiva modificación aprobada por la respectiva facultad.

Para constancia se firma en Medellín, el _____.

DIRECTOR PROYECTO

Dr. Freddy Rafael Pérez

DIRECTOR DE FACULTAD

Ing. Hugo Cardona Restrepo

ESTUDIANTE

Lina Marcela Ruiz Marín

18. CARTAS DE AVAL DE DESARROLLO DEL PROYECTO

Señores

Consejo de Facultad

Facultad de Ingeniería Electrónica

Medellín

Asunto: Proyecto de Grado

Cordial Saludo

La presente tiene como objetivo presentar para su estudio por parte del Consejo de Facultad, el Proyecto de Grado titulado **Diseño e implementación de un dispositivo de control para realizar el proceso de recubrimiento por inmersión**. Manifiesto además que, conozco el manual de Proyecto de Grado de la Escuela de Ingenierías de la UPB y los deberes y derechos que como Estudiante esto implica. El desarrollo del proyecto se hará de conformidad con lo estipulado en dicho manual.

Atentamente,

Lina Marcela Ruiz Marín

CC. 42.691.346 de Copacabana

Estudiante

Facultad de Ingeniería Electrónica

Medellín, 17 de Junio de 2015

Señores

Consejo de Facultad

Facultad de Ingeniería Electrónica

Medellín

Asunto: Director de Trabajo de Grado

Cordial Saludo

La presente tiene como objetivo informar mi intención de participar como Director en el desarrollo del Proyecto de Grado **Diseño e implementación de un dispositivo de control para realizar el proceso de recubrimiento por inmersión** que será desarrollado por la estudiante Lina Marcela Ruiz Marin con ID 000012665. Así mismo, como director del **Grupo de Óptica y Espectroscopía** avalo la ejecución del Proyecto de Grado.

Manifiesto además que, conozco el manual de Proyecto de Grado de la Escuela de Ingenierías de la UPB y los deberes y derechos que como Director esto implica. El desarrollo del proyecto se hará de conformidad con lo estipulado en dicho manual.

Atentamente,

Dr. Freddy Rafael Pérez

CC. 15.043.436 de Sahagún

Doctor en Física

Investigador del proyecto

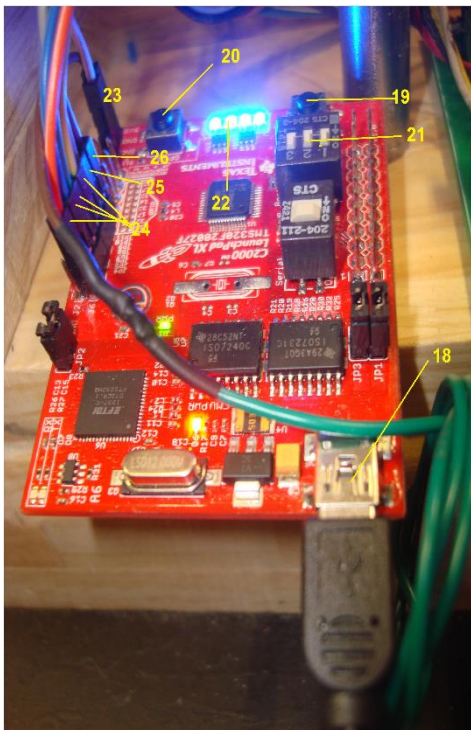
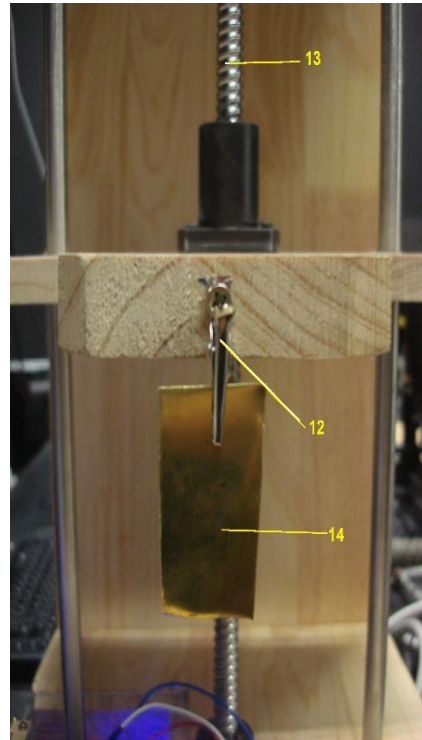
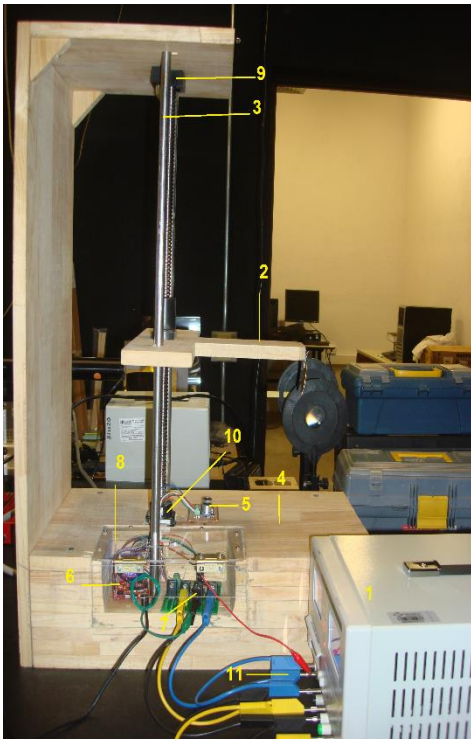
ANEXO 3 - MANUAL DE FUNCIONAMIENTO DEL EQUIPO

Introducción

El presente equipo fue desarrollado para ser utilizado en procesos de recubrimiento por inmersión de placas con diferentes tipos de resinas o soluciones, permitiendo de una manera eficaz controlar la calidad del recubrimiento con la velocidad de ingreso y retiro de las mismas.

Partes del equipo

- 1 Fuente de alimentación regulada Atten TPR3005T-3C
- 2 Soporte o brazo de madera
- 3 Dos guías iguales
- 4 Tapa superior de madera
- 5 Sensor ultrasónico de posicionamiento HY-SRF05
- 6 Tarjeta del microcontrolador Texas Instruments C2000-Launchpad
- 7 Tarjeta con circuitos de potencia y control
- 8 Tapa de acrílico
- 9 Tope superior Misumi BUND12
- 10 Tope inferior Misumi BRW12
- 11 Cableado de potencia
- 12 Pinza
- 13 Tornillo de bolas de precisión C7 Misumi BSST1510-600
- 14 Placa a recubrir
- 15 Cableado del sensor
- 16 Motor de pasos Sanyo Denki SH1603-0440
- 17 Acople entre tornillo de bolas y motor de pasos
- 18 Conector cable USB desde el computador.
- 19 Botón reset del micro
- 20 Botón de programación del micro.
- 21 Swich S1,S2,S3 del micro.
- 22 Led indicador de funcionamiento del micro.
- 23 Pines de conexión a tierra del micro.
- 24 Pines conexión de las fases del motor
- 25 Pines de conexión del *trigger*
- 26 Pin de conexión Ecap (sensor)
- 27 Cables del motor.
- 28 Conector para las fases del motor.



Requerimientos básicos

Un Computador con puerto USB disponible, capacidad de procesamiento adecuado y sistema operativo Windows 7, 8 o 10.

Software requerido:

Instalación de la interfaz de usuario instalada, la cual consta de validar que estén instalados los archivos ejecutar.bat y DipCoating.jar



DipCoating.jar

Instalación de los programas requeridos

Instalación de Java

Ingrese a las página <https://netbeans.org/downloads/> y seleccione la opción de descarga de *All* en la parte inferior para descargar la versión que tiene todos los archivos incluidos. Sin embargo, usted podrá seleccionar los archivos que considere necesarios.

NetBeans IDE 8.0 Download 7.4 | 8.0 | Development | Archive

Email address (optional):

Subscribe to newsletters: Monthly Weekly

NetBeans can contact me at this address

IDE Language: Platform:

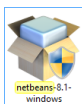
Note: Greyed out technologies are not supported for this platform.

NetBeans IDE Download Bundles

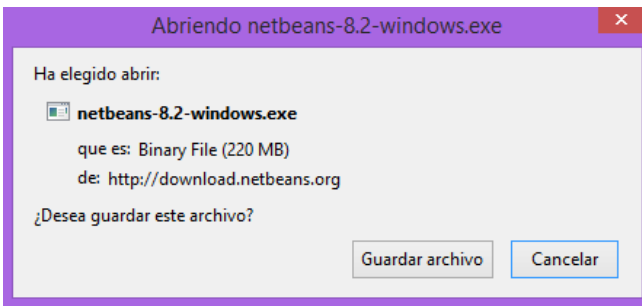
Supported technologies *	Java SE	Java EE	C/C++	HTML5 & PHP	All
<input type="checkbox"/> NetBeans Platform SDK	•	•			•
<input type="checkbox"/> Java SE	•	•			•
<input type="checkbox"/> Java FX	•	•			•
<input type="checkbox"/> Java EE		•			•
<input type="checkbox"/> Java ME					•
<input type="checkbox"/> HTML5				•	•
<input type="checkbox"/> Java Card™ 3 Connected					—
<input type="checkbox"/> C/C++			•		•
<input type="checkbox"/> Groovy					•
<input type="checkbox"/> PHP				•	•
Bundled servers					
<input type="checkbox"/> GlassFish Server Open Source Edition 4.0		•			•
<input type="checkbox"/> Apache Tomcat 8.0.3		•			•

Free, 89 MB
Free, 191 MB
Free, 62 MB
Free, 63 MB
Free, 203 MB

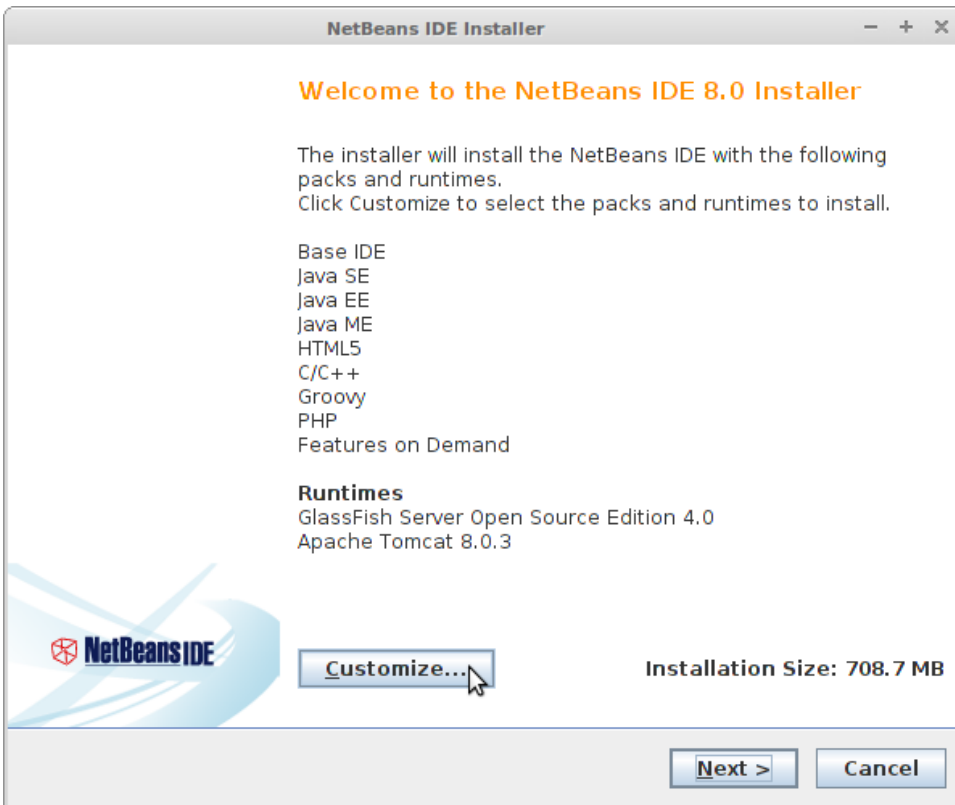
En la carpeta de descargas del computador le aparece el siguiente ícono, dar doble click:



Le aparecerá la siguiente ventana y debe seleccionar guardar archivo. Empezará automáticamente la descarga.

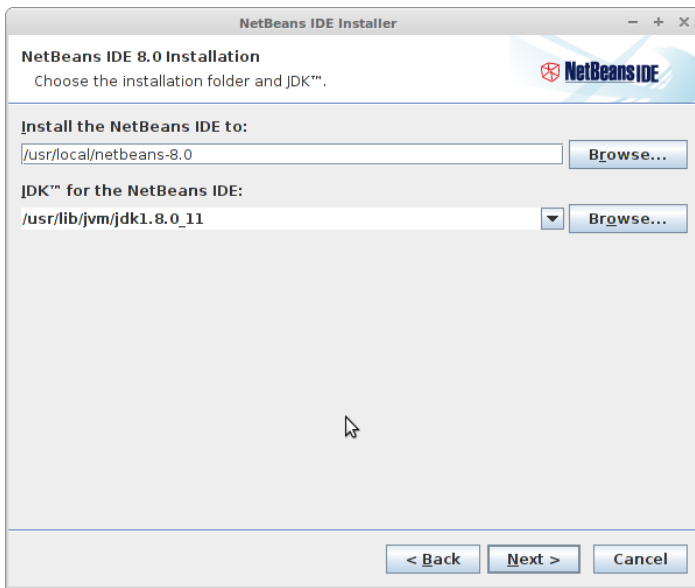


Entonces aparecerá la siguiente ventana.

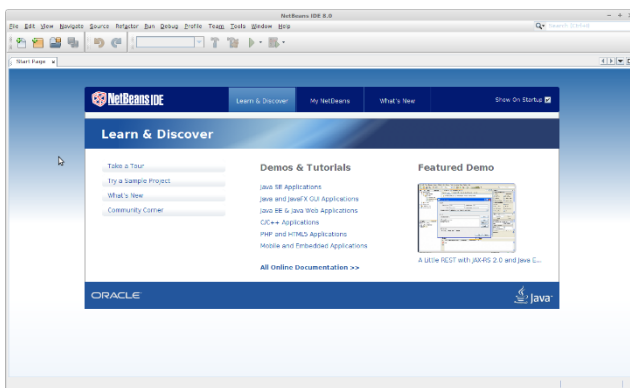


Luego, se procede a seleccionar siguiente, siguiente, siguiente, acepto los términos y condiciones y aceptar.

Si el JDK está correcto, debe de verse la siguiente pantalla:



Basta con dar click en siguiente, de lo contrario cambie la ruta predeterminada para la instalación si lo desea. Finalmente se instalará y podrán ejecutarlo con el comando “netbeans”. O bien, encontrarlo en su menú principal, el programa se abrirá de la siguiente manera.



Descarga del CCS para el uso del microcontrolador Texas Instruments f28027f

Usted debe crear una cuenta de usuario para que la empresa le permita la descarga del software. Luego, debe ingresar con su nombre de usuario y contraseña antes del proceso.

Ingrese a la página <http://www.ti.com/tool/launchxl-f28027?DCMP=c2x-launchpad&HQS=c2000-launchpad> Aquí podrá visualizar toda la información correspondiente al microcontrolador, el software de descarga y guías de usuario necesarias para su manipulación.

Paso 1: descargar controlSUITE.

Seleccione esta opción *controlSUITE software suite*.

The C2000 LaunchPad is based around the C2000 Piccolo TMS320F28027 MCU, which features a 60 MIPS processing core, 64 KB integrated flash, 8 PWM channels with high resolution capability, 12-bit 4.6 MSPS ADC, capture interface, serial connectivity, and more.

The board includes an isolated XDS100 JTAG emulator facilitating easy programming and debugging, 40 PCB pins providing easy access to the F28027 processor, convenient reset and programmable buttons, and more. Plus, with the BoosterPack ecosystem, designers can expand the functionality of the C2000 LaunchPad hardware to explore application use-cases.

Software Features

LaunchPad software example code, libraries, drivers, and more are provided for free through the C2000 MCU family's **controlSUITE™ software suite**. Likewise, users get access to an unrestricted version of Code Composer Studio integrated development environment (IDE) for software development with the C2000 Piccolo TMS320F28027 microcontroller.

Getting Started

Getting started with your C2000 LaunchPad is easy, thanks to a simple and fun out-of-the box project:

Get started today with My First C2000 LaunchPad Project: Internal Temperature Measurement!

Get to Know C2000 Microcontrollers
C2000 MCUs enable uncompromising performance for various real-time control applications like motor control, digital power supplies, and solar.
> more

Like the LaunchPad? Upgrade it with InstaSPIN!
The same LaunchPad you love
Now InstaSPIN-FOC enabled for motor applications
24V/10A Booster Pack
> more

Se abrirá una nueva ventana donde debe seleccionar la opción Get software.

controlSUITE™ Software Suite: Essential Software and Development Tools for C2000™ Microcontrollers

(ACTIVE) CONTROL SUITE

Description & Features
 Technical Documents
 Support & Training

Order Now

Part Number	Buy from Texas Instruments or Third Party	Alert Me	Status	Current Version	Version Date	Description
CONTROLSUITE-ZIP: Offline (ZIP) Installer	Free Get Software	Alert Me	ACTIVE	v3.4.3	07-NOV-2016	Drivers, libraries, BOMs, demos, schematics & code examples, etc. (powerSUITE & DesignDRIVE software)
CONTROLSUITE: Web (EXE) Installer	Free Get Software	Alert Me	ACTIVE	v3.4.3	07-NOV-2016	Drivers, libraries, BOMs, demos, schematics & code examples, etc. (powerSUITE & DesignDRIVE software)

Le aparecerá una ventana donde usted deberá seleccionar el uso que le dará al controlador, además de validar que su información personal es verídica. Seleccionar por ultimo *Submit*

En la siguiente ventana seleccionar *Download*

1 Home

TI Request

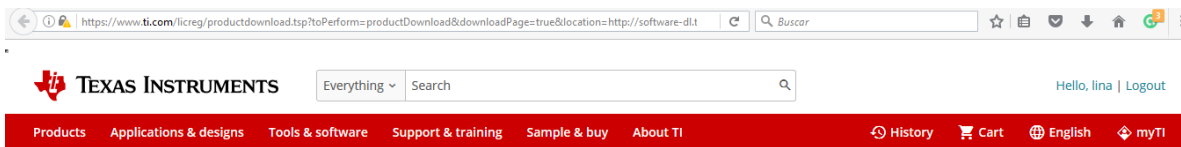
You have been approved to receive this file.
Click "Download" to proceed.

In a few moments, you will also receive an email with the link to this file.

[Download](#)

Having trouble downloading? Try www.ti.com/software-help

Dar click en *HERE* si no empieza inmediatamente la descarga.

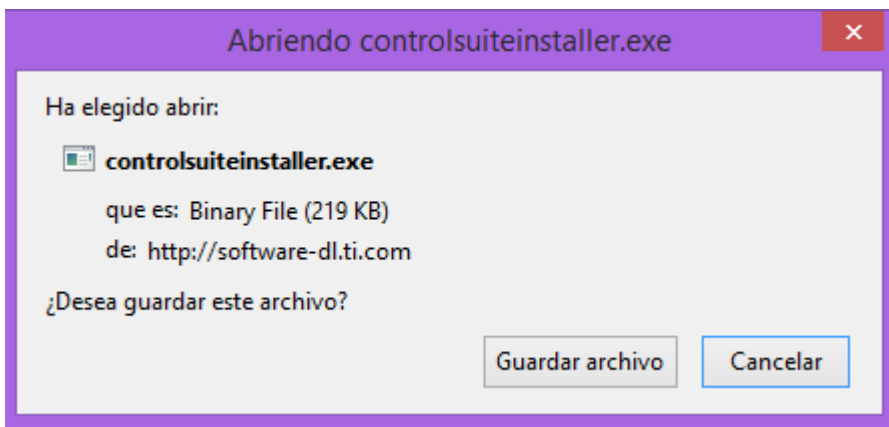


Download

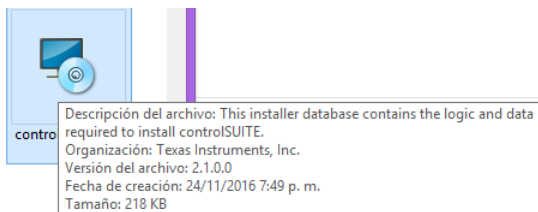
- Please monitor the download process to completion.
- Please close this window when download is complete.

Having trouble downloading? Click [HERE](#) to start download
Still having issues? Try: www.ti.com/software-help

Seleccionar en la ventana siguiente Guardar archivo empezara la descarga.



Buscar en la carpeta de descargas el siguiente icono



De doble click en el anterior icono y continúe con la instalación.

Paso 2: Realice un procedimiento similar para la descarga he instalación del archivo [Code Composer Studio](#)

Partiendo de la selección de esta opción en la pagina <http://www.ti.com/tool/launchxl-f28027?DCMP=c2x-launchpad&HQS=c2000-launchpad>

The C2000 LaunchPad is based around the C2000 Piccolo TMS320F28027 MCU, which features a 60 MIPS processing core, 64 KB integrated flash, 8 PWM channels with high resolution capability, 12-bit 4.6 MSPS ADC, capture interface, serial connectivity, and more.

The board includes an isolated XDS100 JTAG emulator facilitating easy programming and debugging, 40 PCB pins providing easy access to the F28027 processor, convenient reset and programmable buttons, and more. Plus, with the BoosterPack ecosystem, designers can expand the functionality of the C2000 LaunchPad hardware to explore application use-cases.

Software Features

LaunchPad software example code, libraries, drivers, and more are provided for free through the C2000 MCU family's [controlSUITE™ software suite](#). Likewise, users get access to an unrestricted version of [Code Composer Studio](#) integrated development environment (IDE) for software development with the C2000 Piccolo TMS320F28027 microcontroller.

Getting Started

Getting started with your C2000 LaunchPad is easy, thanks to a simple and fun out-of-the box project:

[Get started today with My First C2000 LaunchPad Project: Internal Temperature Measurement!](#)

Features



Get to Know C2000 Microcontrollers

C2000 MCUs enable uncompromising performance for various real-time control applications like motor control, digital power supplies, and solar.

[> more](#)



Like the LaunchPad? Upgrade it with InstaSPIN!

The same LaunchPad you love
Now InstaSPIN-FOC enabled for motor applications
> 24V/10A Booster Pack

[> more](#)

Seleccione el sistema operativo utilizado



Code Composer Studio™ - Integrated Development Environment for C2000 32-bit Real-time Control MCUs

- **CCS latest version** - Click below to download CCSv6 for the specified host platform.
- **Additional downloads** - For a complete list of downloads visit the [CCS download site](#).
- **Use CCS for free** - A free license will be generated that supports working with low cost XDS100 debug probes or boards with an onboard debug probe. A 90-day extension for the full featured evaluation license is also available.



Aparecerá nuevamente la ventana donde usted debe seleccionar el uso que le dará al software civil o militar, sus datos personales y la confirmación que toda la información es verídica. Seleccione nuevamente la opción Submit- Download-HERE-Guardar archivo y empezara la descarga. Continuar con la instalación del archivo de la manera anterior.


Paso 3: Conecte el controlador al portátil e instale los drivers desde el mismo para su reconocimiento posterior.

Descarga de Hercules

Hercules es una interfaz utilizada para la comunicación serial entre el microcontrolador y el computador.

Ingrese a la página http://www.hw-group.com/products/hercules/index_es.html y seleccione la siguiente opción

Downloads & Links

- **DOWNLOAD:**  [Download version 3.2.8](#)
- Utilidad Hercules está **relacionada con siguientes dispositivos:**
 - [Charon I](#) - incrustado módulo convertidor RS-232 (TTL) / Ethernet
 - [PortBox](#) - módulo convertidor de RS-232/485 / Ethernet
 - [PortStore](#), [PortStore2](#) - convertidor RS-232/485 / Ethernet con 2 048 kB de memoria buffer Flash para los datos seriales en línea.
 - [I/O Controller](#) - entradas y salidas binarias + convertidor RS-232/485 - Ethernet
 - [IP Relay](#) - 2x relés controlados a través de Ethernet + convertidor RS-232/485 / Ethernet
- [TEA](#) - codificación soportada & algoritmo autorizado ([TCP Client](#), [TCP Server](#), tabs de [Modo de prueba](#) ..)
- [NVT](#) - **Network Virtual Terminal (terminal virtual de red)** para los dispositivos de control remoto a través de TCP

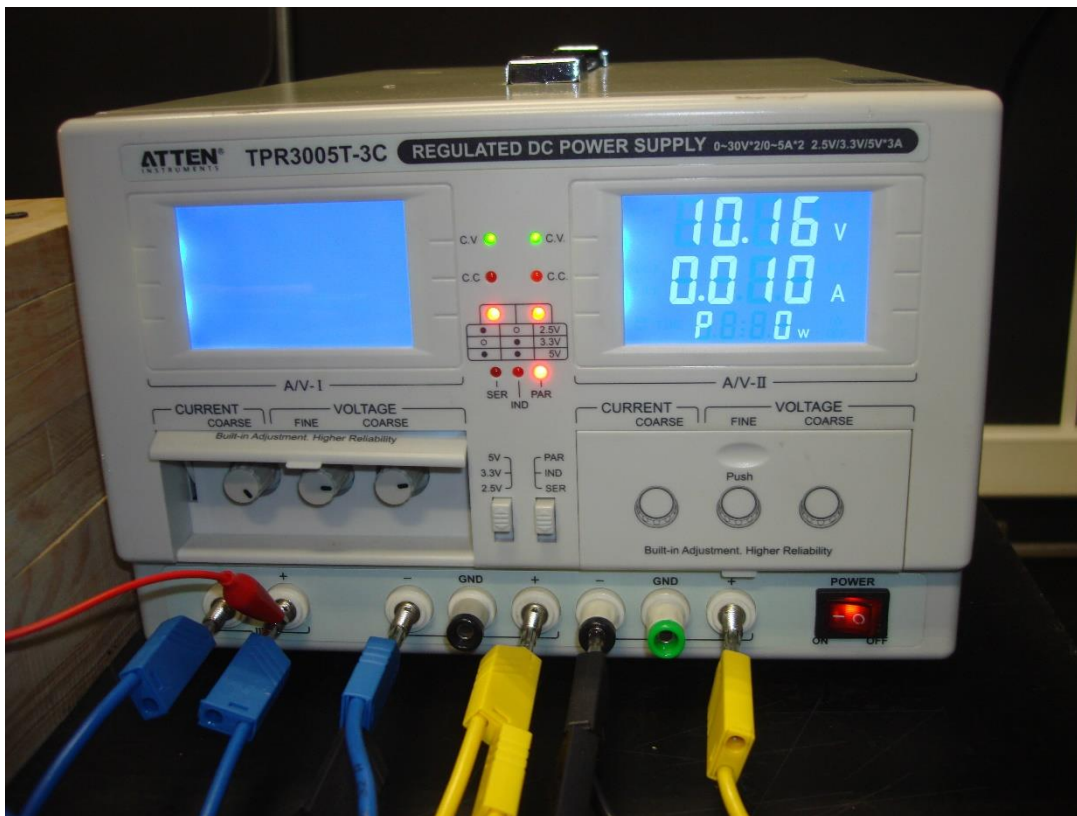
Seleccione la opción guardar archivo en la ventana emergente y automáticamente empezara la descarga.

En la carpeta de descargas seleccione el archivo dándole doble click, le aparecerá una ventana donde debe seleccionar permitir acceso y continuara con el proceso de instalación, apareciendo el icono en el escritorio y en la barra de inicio.

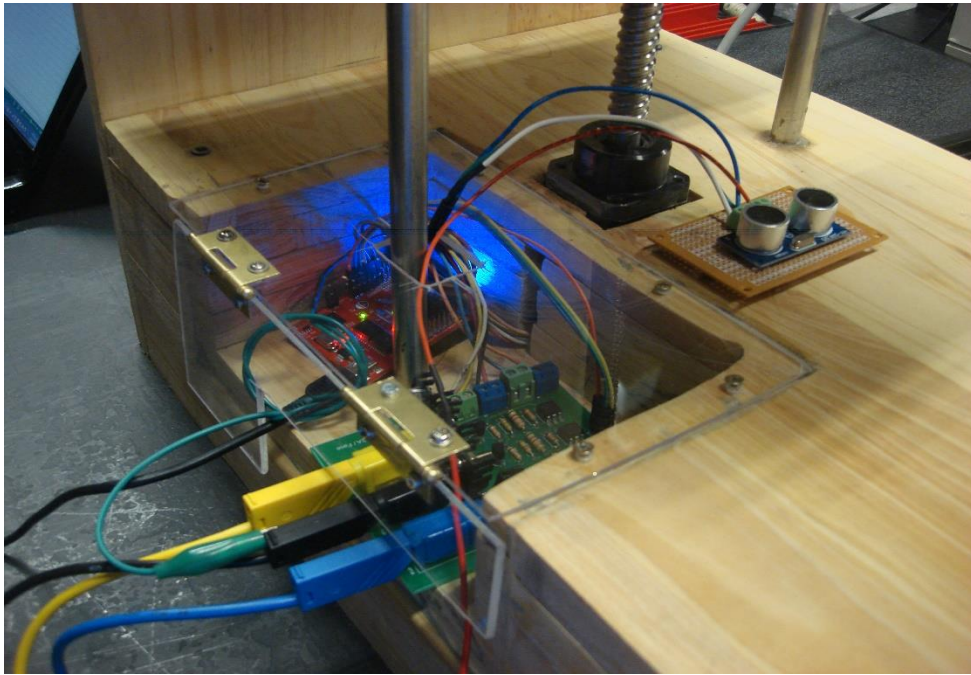
Conexión del equipo

Conexiones de la fuente y validaciones antes de comenzar.

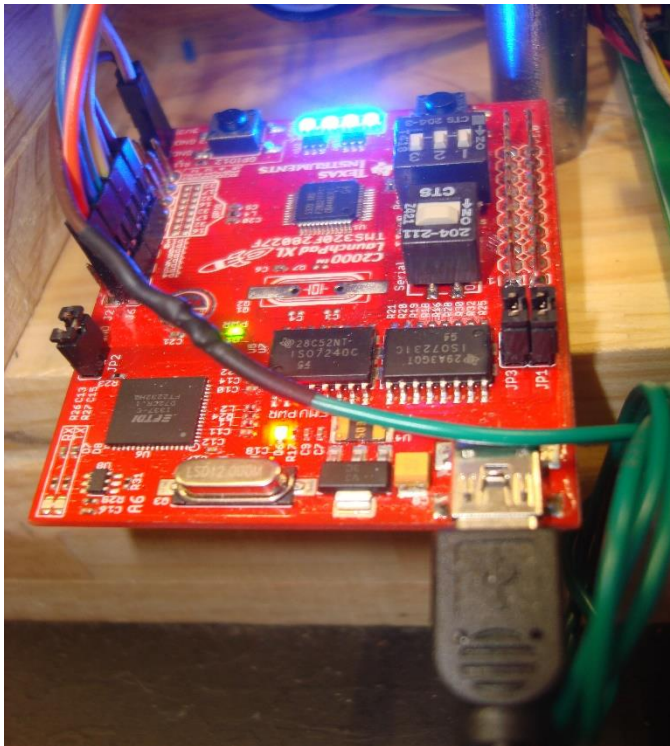
1. Conecte la fuente en paralelo y la fuente de 5 V desplazando los botones del panel frontal de la misma hacia arriba. Este proceso se valida mediante el encendido del led rojo que se encuentra encima de la palabra PAR.



2. Valide con el multímetro que la fuente este brindando a la salida del lado izquierdo un voltaje de 5 V y por la otra 12 V. Si esto no es correcto, levante las tapas de la fuente que dicen encima la palabra PUSH, y con las perillas que aparecen debajo de la palabra VOLTAGE, aumente o disminuya el mismo hasta que llegue al voltaje deseado.
3. Con uno de los cables azules conecte o realice un puente o conexión entre los dos negativos de la fuente de 5 V como se muestra en la foto anterior
4. Con el otro cable banana-banana azul del positivo de la fuente de 5 V al positivo de 5 V (conector azul) de la tarjeta de control verde.
5. Conecte el cable rojo de alimentación del sensor ultrasónico proveniente del equipo a la fuente de 5 V como se muestra en la figura con un cable caimán rojo.
6. Realice un puente o conecte los positivos de la fuente de 12 V entre sí, con el cable banana-banana amarillo.
7. Conecte el otro cable amarillo desde el positivo de la fuente de 12 V al positivo de 12 voltios (conector amarillo) de la tarjeta de control verde.
8. Conecte el cable banana-banana negro, desde el negativo de la fuente de 12 Voltios a la tierra (conector negro) de la tarjeta verde ubicada en el equipo.
9. Conecte el caimán verde proveniente de la tarjeta roja del microcontrolador al cable negro o tierra de la tarjeta de control verde como se visualiza en la fotografía.




11. Por último, conecte el cable USB desde el puerto del computador al puerto del microcontrolador, validando su correcto funcionamiento mediante el encendido de los cuatro LEDs de color azul de la tarjeta.

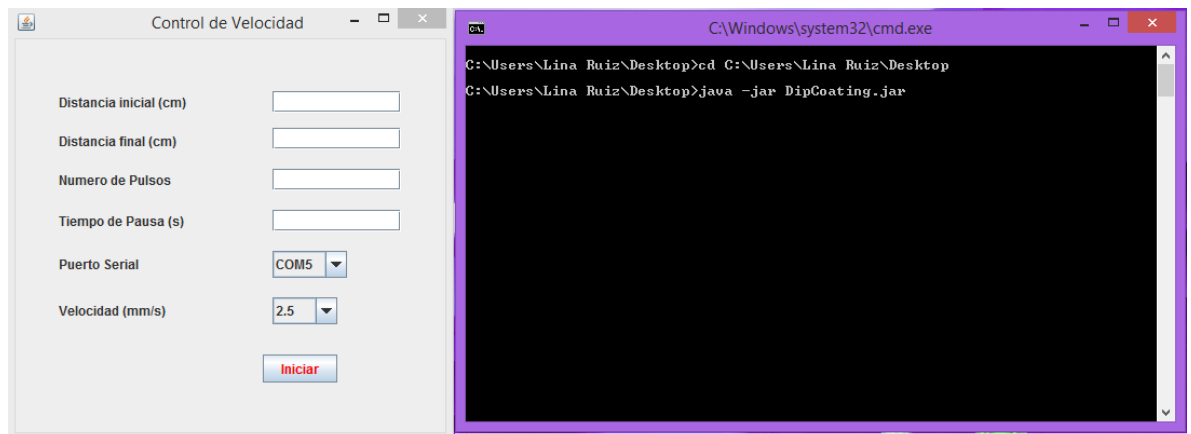


12. Para finalizar encienda la fuente, presionando el interruptor de POWER.

13. Ahora en el computador, el cual debe estar encendido...

14. Proceda a seleccionar y dar click en el icono  de ejecutar.bat hasta lograr abrir la interfaz gráfica del equipo.

Le aparecerán las siguientes ventanas en el escritorio.



15. En la anterior ventana, usted debe ingresar valores numéricos sin puntos o comas de la distancia inicial de ubicación del brazo, para posteriormente ubicar el *beaker* con la solución, además de la distancia final a la cual va a llegar el brazo para sumergir la placa dentro de la misma. Otras variables son: el número de pulsos, tiempo de pausa, número del puerto para la comunicación serial y la velocidad que utilizara en el proceso.

Recuerde que usted puede validar el número del puerto al cual se conectó el equipo siguiendo la siguiente ruta.

Configuración-Panel de control- administrador de dispositivos-Puertos COM y LPT.

Los valores mínimos y máximos permitidos por el equipo están definidos en la siguiente tabla.

Valores permitidos por la interfaz de Java		
Valores	Mínimo	Máximo
Distancia (cm)	8	47
Numero de pulsos	1	999
Tiempo de pausa (s)	1	60
Puerto Serial	COM1	COM12
Velocidad (mm/s)	2.5	17.5

Tabla 9. Valores permitidos por la interfaz de Java

Advertencia

Debe tener en cuidado cuando se ingresen los datos de número de pulsos y velocidad debido a que si ambos se encuentran en sus valores máximos, el equipo puede no realizar el desplazamiento, debido a la limitación del controlador para generar varios pulsos a una mayor velocidad.

16. Seleccione el botón INICIAR de la interfaz, y el equipo debe empezar con su desplazamiento a la distancia inicial
17. Al llegar a la distancia inicial, se generará una corta pausa. Es ahí donde se debe apagar la fuente de voltaje desde el interruptor de encendido POWER de la fuente.
18. Ubique el *beaker* con la solución debajo del brazo, además de sujetar la placa a recubrir con la pinza del equipo.
19. Nuevamente presione el interruptor de POWER para desplazar la placa dentro de la solución.
20. Al finalizar, el equipo llegará a la distancia final.
21. Si desea retirar la placa de la solución, se recomienda ingresar nuevamente los datos de distancia y demás a la interfaz, este proceso se realiza reseteando el microcontrolador

Como resetear el microcontrolador

- Apague la fuente.
- Cierre la interfaz de usuario o el software del equipo.
- Asegúrese que el *switch* S3 se encuentre en la posición *OFF*
- Presione el botón *Reset* y posteriormente el botón de programación del microcontrolador.
- Abra el programa he ingrese nuevamente todos los datos.

Nota: Este equipo le permite al usuario configurar la velocidad de retiro e ingreso de la placa dentro de la solución a diferentes velocidades y parámetros.

Normas de Seguridad

- Utilice y cumpla con todas las normas establecidas de protección personal en el laboratorio. Se recomienda el uso de gafas, delantal, guantes, etc., antes de empezar a realizar todos los procesos con el equipo.
- Valide siempre que el equipo se encuentre conectado de forma correcta, evitando cortos o accidentes de tipo eléctrico, que puedan llegar a afectar su integridad y deteriorar el funcionamiento del dispositivo.
- No coma ni fume durante la manipulación del equipo. Esto puede causar daños irreparables en los circuitos o causar incendios en el lugar.
- Recuerde evitar derramar las soluciones químicas utilizadas en el proceso, sobre el acrílico, superficies de madera, circuitería y demás superficie es que no se encuentran protegidas con la placa de acero inoxidable.
- Nunca aumente el voltaje o la polaridad establecidos en el presente manual.
- No engrase el tornillo con ningún tipo de aceite. Se recomienda usar una grasa con mayor densidad, preferiblemente de litio, para evitar cualquier derrame de esta sustancia sobre el motor disminuyendo así su vida útil.

Configuraciones eléctricas del equipo

	Voltaje (V)	Corriente(A)
Fuente 1	5	4
Fuente 2	12	4

Tabla 10. Especificaciones eléctricas del equipo

Especificaciones eléctricas de la Fuente.

Voltaje (V)	Corriente(A)	Potencia(W)
0-30 (Ajustable)	0-5	150

Tabla 11. Especificaciones eléctricas de la Fuente.

Especificaciones eléctricas del motor

Número de fases	Fuente de voltaje (V)	Corriente por fase (A)
2	24	2

Tabla 12. Datos eléctricos del motor de pasos.

Desarmando el equipo

1. Retire con un destornillador de pala los tornillos de ensamble del tope superior
2. Retire las guías jalándolas hacia arriba de la tapa superior de las mismas. Ayúdese de ser necesario con ambas manos.
3. Con un destornillador de estrella, retire todos los tornillos que sujetan la tapa de acrílico y retire la misma, no sin antes haber desconectado cada uno de los cables que van a ambas tarjetas del controlador y al circuito de potencia.
4. Despegue el sensor con mucho cuidado y guárdelo.
5. Retire con un destornillador de estrella la placa de protección de acero inoxidable ubicado sobre la superficie.
6. Con una llave pequeña hexagonal, introdúzcala en la ranura lateral, con el fin de aflojar el tornillo sin fin que está sujeto por tres tonillos de un acople, que lo mantienen centrado y sujeto al eje del motor.
7. Retire el tornillo sin fin, teniendo cuidado de no ir a generar juego en el eje de centro.
8. Retire los cuatro tornillos de ensamble que sujetan la última tabla de madera que sostiene el equipo y retírela.
9. De esta forma, podrá tener acceso al motor y validar su correcto funcionamiento al girar el eje con los dedos.
10. En caso de producirse cambios en la temperatura de los circuitos, humo u olores extraños, se recomienda apagar la fuente para evitar daños peores al sistema.

ANEXO 4 - DISEÑO DE LA MAQUETA

Materiales

- 2 varillas de $\frac{1}{2}$ cada una de 50 cm.
- 1 Tornillo de $\frac{3}{8}$ roscado de 50 cm.
- 1 Platina calibre 12 de 33 largo por 10 de ancho.
- 2 Tuercas de $\frac{3}{8}$.
- 2 arandelas $\frac{3}{8}$.
- 1 perfil en U tallado calibre 20 de 33 cm de ancho X 6 cm largo.
- 2 remaches pop.
- 1 Soporte para tornillo de $\frac{3}{8}$.
- 1 base en madera triples de 35X35 cm.
- 2 puntos de Soldadura autógena.
- 8 puntos soldadora eléctrica.
- Colbón para madera.

Herramientas

- Taladro.
- Brocas de $\frac{3}{8}$ $\frac{1}{8}$ y $\frac{1}{2}$.
- Segueta.
- Martillo.
- Alicates de presión (hombre solo).
- Alicates.
- Destornillador de estrella.
- Remachadora pop.
- Lija
- Mordaza o prensa.

Elementos de medida Metro, nivel y escuadra.

Procedimiento

Primero, se hizo soldar las dos varillas de $\frac{1}{2}$ a la platina metálica calibre N 12 con soldadura autógena. Posteriormente, se procedió a hacer los tres orificios en el perfil en U, para que pasaran las dos varillas de $\frac{1}{2}$ y la varilla roscada que debía estar ubicada en el centro.

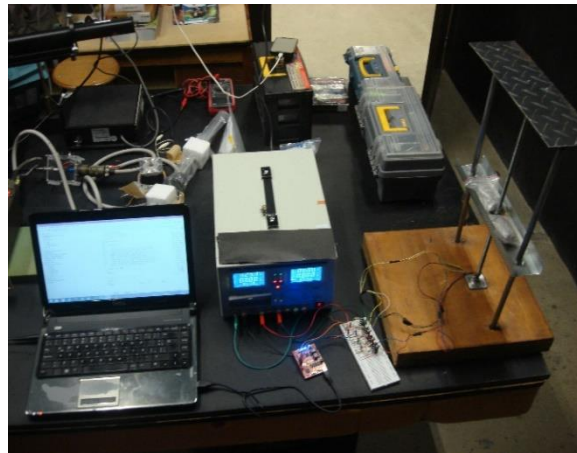
Además de lo anterior, se solicitó que se le hiciera al tornillo sin fin un orificio del tamaño del eje del motor en el centro, para posteriormente soldarle una tuerca y ubicar allí un tornillo pequeño que permitiera fijar el motor y generar el movimiento de ambas piezas.

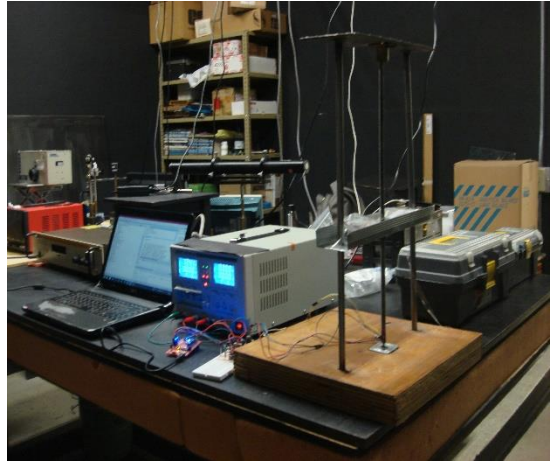
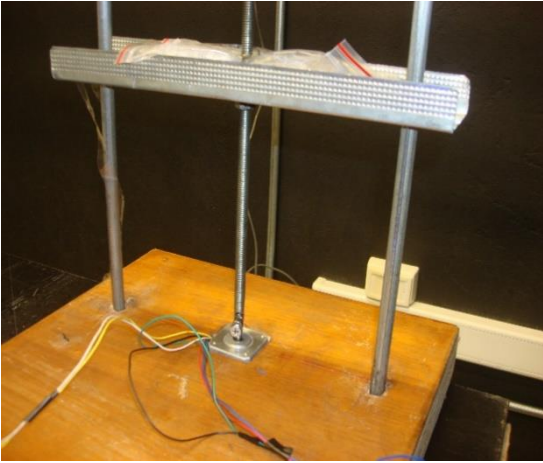
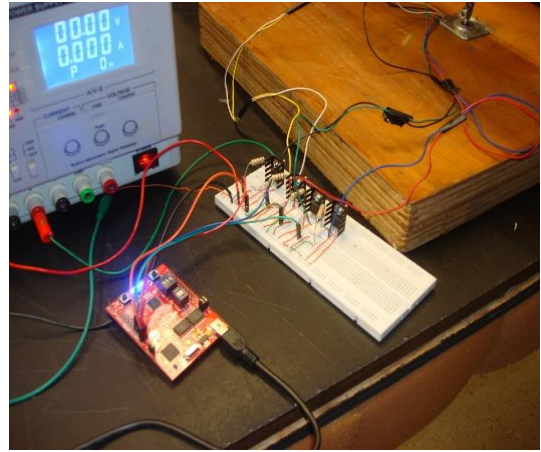
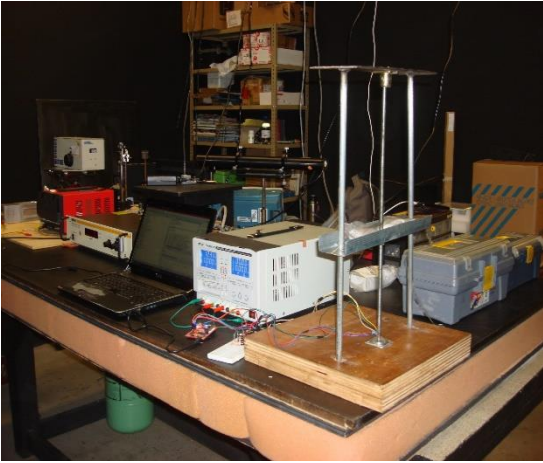
Luego, se llevó el perfil en U para que le soldaran por cada lado del orificio del centro, una tuerca con su arandela, siendo necesario utilizar soldadura autógena por el tipo de material del perfil y su grosor.

Posteriormente, se llevó el motor con todos los materiales anteriormente modificados donde el carpintero, para que realizara la base en madera con los orificios para el motor y las dos varillas de $\frac{1}{2}$.

Finalmente se realizó la lubricación de las piezas móviles y su ajuste, debido a que las piezas no fueron vaciadas y realizadas con los moldes de un diseño mecánico, provocando vibración y movimientos descontrolados que pueden afectar el proceso químico y el recubrimiento.

Fotos





ANEXO 5 – PROGRAMAS REALIZADOS

5.1 Código control de velocidad equipo dipcoating.

```
// Velocidades que genera el equipo 25Hz 50Hz 75Hz 100Hz 125Hz 137Hz 150Hz 162Hz 175Hz.

#include "DSP28x_Project.h"

#include "f2802x_common/include/clk.h" //Libreria que genera la señal de reloj
#include "f2802x_common/include/flash.h" //Libreria utilizada para arrancar el programa desde la flash
#include "f2802x_common/include/gpio.h" //Libreria para configurar los puertos de entrada y salida
#include "f2802x_common/include/pie.h" //Libreria para realizar interrupciones.
#include "f2802x_common/include/pll.h" // libreria para generar un valor de referencia.
#include "f2802x_common/include/cap.h" //Libreria para generar la captura de las señales del sensor
#include "f2802x_common/include/pwm.h" //Libreria para generar la señal del módulo pwm.
#include "f2802x_common/include/wdog.h" //libreria que controla el perro.
#include "f2802x_common/include/sci.h" //Libreria para configurar la comunicacion serial.
#include "f2802x_common/include/timer.h"//Libreria para usar el timer

typedef struct {

                                // Generacion de estructuras;

    PWM_Handle myPwmHandle;

    uint16_t EPwm_CMPA_Direction;

    uint16_t EPwm_CMPB_Direction;

    uint16_t EPwmTimerIntCount;

    uint16_t EPwmMaxCMPA;

    uint16_t EPwmMinCMPA;

    uint16_t EPwmMaxCMPB;

    uint16_t EPwmMinCMPB;

} EPWM_INFO;

#define EPWM2_RETARDO_SUBIDA 300 //Retardos de tiempo establecidos.
#define EPWM2_RETARDO_BAJADA 300

#define EPWM_CMP_UP 1 //Define el nivel de señal a comparar.
#define EPWM_CMP_DOWN 0
```

```

// Funciones creadas

__interrupt void epwm1_isr(void);           //Función atención a la interrupción epwm1 y pwm2
__interrupt void epwm2_isr(void);
__interrupt void cpu_timer0_isr(void);
__interrupt void scia_rx(void);           //Generación de Interrupción por los datos entrantes
interrupt void ecap1_isr(void);           //Lectura del tiempo medido por evento en CAP1

void ConfigurarPIE(void);                 //Función configuración inicial de las interrupciones.
void ConfigurarCAP(void);                 //Configurar CAP1
void Configurar_ePWM_arriba(void);        //Configura el módulo PWM movimiento hacia arriba.
void Configurar_ePWM_abajo(void);        //Configura el módulo PWM movimiento hacia abajo.
void HabilitarCAP(void);                  //Activa el módulo Cap

void InhabilitarCAP(void);                //Desactiva el modulo Cap.

void HabilitarTimer(void);                //Habilita el módulo timer, genera una señal de reloj.

void HabilitarPWM(void);                  //Habilita el modulo Pwm.

void ObtenerDatos(void);                  //Guarda los datos ingresados por el usuario.
void AjustarVelocidad(void);              //Ajuste de la frecuencia del módulo PWM.
void EfectuarMovimiento(void);            //Toma medidas Ecap, valida topes y efectúa movimiento.
void ObtenerMedida(void);                  //Toma las medidas ingresadas por el Cap.

uint16_t RecibirNumero(void);             //Toma el numero ingresado y valida que no sea un carácter.

void ImprimirVector(void);                //Imprime información guardada en el vector.

void ImprimirNumero(uint32_t numero);      //Imprime número.
void scia_init(void);                      //Inicialización del módulo SCI-Serial
void scia_xmit(char a);                    //Enviar carácter

void print(char *msg);                    //Enviar cadena

char scia_recibir(void);                   //Recibe caracteres

void retardo_us(uint16_t i);               //1 us aproximado

```

```

//Handle de los modulos utilizados.

EPWM_INFO epwm1_info;
EPWM_INFO epwm2_info;
CLK_Handle myClk;           // Handle Reloj
FLASH_Handle myFlash;      // Handle memoria Flash
GPIO_Handle myGpio;        // Handle puertos de entrada y salida
PIE_Handle myPie;          // Handle interrupciones.
PWM_Handle myPwm1, myPwm2; // Handle Pwm
SCI_Handle mySci;          // Handle puerto serial
CAP_Handle myCap;          // Handle modulo Cap
CPU_Handle myCpu;          // Handle CPU
PLL_Handle myPll;          // Handle PLL
WDOG_Handle myWDog;        // Handle DOG
TIMER_Handle myTimer0;     // Handle TIMER

// variables Globales

uint16_t EPWM1_TIMER_TBPRD=0;
uint16_t EPWM1_MITAD_CMPA=0;
uint16_t EPWM1_MITAD_CMPB=0;
uint16_t EPWM2_TIMER_TBPRD=0;
uint16_t EPWM2_MITAD_CMPA=0;
uint16_t EPWM2_MITAD_CMPB=0;

uint16_t vector[10];       //Arreglo para recibir caracteres por parte del serial.
uint16_t cifras[10];       //Numero de cifras del vector.
uint16_t n=0;              //Numero de posiciones usadas en vector.
uint16_t p=0;              //Numero de posiciones usadas en cifras.
uint16_t tope=0;           //Bandera para verificar que la nuez se encuentre en los limites
                             permitidos
//uint16_t k=0;            //
uint16_t numero_pulsos=0;  //Numero de pulsos por movimiento
uint16_t distancia=0;      //Distancia deseada
uint32_t valor=0;          //Guarda los datos tomados por el Ecap.

uint32_t medida=0;

uint16_t m=0;

```

```

uint16_t j=0;uint32_t numero=0;

uint32_t v=1;

uint16_t npasos=10;           //Numero de pasos generados continuamente en cada movimiento.
uint16_t velocidad=0;        //Variable que almacena la velocidad enviada por la interface.

int32_t distanciaini=0;      //Distancia inicial

int32_t intercepto=53265;    //Pendiente he intercepto que describe el comportamiento del equipo.
int32_t pendiente=3304;     //

uint32_t cuentas=0;         //Variable que almacena el valor de las cuentas iniciales "Distancia
Final" al que debe llegar el equipo.

uint32_t cuentasini=0;      //Variable que almacena el valor de las cuentas iniciales "Distancia
Inicial" al que debe llegar el equipo.

int time=0;

int giro=0;                  //Bandera para el sentido de giro

_Bool capturaCompleta=false; //Bandera que determina si finalizo de tomar medidas del Ecap

unsigned long timer0IntCount;

// Confuguraciones dadas por el fabricante para usar la memoria Flash.

extern Uint16 RamfuncsLoadStart;
extern Uint16 RamfuncsLoadSize;
extern Uint16 RamfuncsRunStart;

void main(void) {

// Inicialización de todos los handles necesarios para esta aplicación.

myClk = CLK_init((void *) CLK_BASE_ADDR, sizeof(CLK_Obj));
myCpu = CPU_init((void *) NULL, sizeof(CPU_Obj));
myFlash = FLASH_init((void *) FLASH_BASE_ADDR, sizeof(FLASH_Obj));
myGpio = GPIO_init((void *) GPIO_BASE_ADDR, sizeof(GPIO_Obj));
myPie = PIE_init((void *) PIE_BASE_ADDR, sizeof(PIE_Obj));
myPll = PLL_init((void *) PLL_BASE_ADDR, sizeof(PLL_Obj));
myPwm1 = PWM_init((void *) PWM_ePWM1_BASE_ADDR, sizeof(PWM_Obj));
myPwm2 = PWM_init((void *) PWM_ePWM2_BASE_ADDR, sizeof(PWM_Obj));
myWdog = WDOG_init((void *) WDOG_BASE_ADDR, sizeof(WDOG_Obj));
mySci = SCI_init((void *) SCIA_BASE_ADDR, sizeof(SCI_Obj));
myCap = CAP_init((void *)CAPA_BASE_ADDR, sizeof(CAP_Obj));

```

```

myTimer0 = TIMER_init((void *)TIMER0_BASE_ADDR, sizeof(TIMER_Obj));
timer0IntCount = 0;

//inicializacion basica

WDOG_disable(myWDog);          //Deshabilitar el perro
CLK_enableAdcClock(myClk);     //Habilitar reloj Adc
(*Device_cal )();
CLK_disableAdcClock(myClk);    //Deshabilitar el reloj modulo ADC

//Seleccion ocsilador interno pll 60MHz

CLK_setOscSrc(myClk, CLK_OscSrc_Internal);
CLK_setClkOutPreScaler(myClk, CLK_ClkOutPreScaler_SysClkOut_by_1);
PLL_setup(myPll, PLL_Multiplier_12, PLL_DivideSelect_ClkIn_by_2);

//Inhabilita todas las interrupciones

PIE_disable(myPie);           //Deshabilita todas las interrupciones
PIE_disableAllInts(myPie);    //Deshabilita todas las interrupciones
CPU_disableGlobalInts(myCpu); //Deshabilita las interrupciones globales
CPU_clearIntFlags(myCpu);     //Limpia las banderas de las interrupciones CPU

// Declarar los pines del PWM

GPIO_setPullUp(myGpio, GPIO_Number_0, GPIO_PullUp_Disable);
GPIO_setPullUp(myGpio, GPIO_Number_1, GPIO_PullUp_Disable);
GPIO_setMode(myGpio, GPIO_Number_0, GPIO_0_Mode_EPWM1A);
GPIO_setMode(myGpio, GPIO_Number_1, GPIO_1_Mode_EPWM1B);
GPIO_setPullUp(myGpio, GPIO_Number_2, GPIO_PullUp_Disable);
GPIO_setPullUp(myGpio, GPIO_Number_3, GPIO_PullUp_Disable);
GPIO_setMode(myGpio, GPIO_Number_2, GPIO_2_Mode_EPWM2A);
GPIO_setMode(myGpio, GPIO_Number_3, GPIO_3_Mode_EPWM2B);
GPIO_setMode(myGpio, GPIO_Number_18, GPIO_18_Mode_XCLKOUT);

// Declarar los pines 28 y 29 para la comunicación serial

GPIO_setPullUp(myGpio, GPIO_Number_28, GPIO_PullUp_Enable);
GPIO_setPullUp(myGpio, GPIO_Number_29, GPIO_PullUp_Disable);

```

```

GPIO_setQualification(myGpio, GPIO_Number_28, GPIO_Qual_ASync);
GPIO_setMode(myGpio, GPIO_Number_28, GPIO_28_Mode_SCIRXDA);
GPIO_setMode(myGpio, GPIO_Number_29, GPIO_29_Mode_SCITXDA);

//Declarar los pines del CAP

GPIO_setPullUp(myGpio, GPIO_Number_5, GPIO_PullUp_Enable);
GPIO_setQualification(myGpio, GPIO_Number_5, GPIO_Qual_Sync);
GPIO_setMode(myGpio, GPIO_Number_5, GPIO_5_Mode_ECAPI);
GPIO_setPullUp(myGpio, GPIO_Number_4, GPIO_PullUp_Enable);
GPIO_setMode(myGpio, GPIO_Number_4, GPIO_4_Mode_GeneralPurpose);
GPIO_setDirection(myGpio, GPIO_Number_4, GPIO_Direction_Output);

//Inicializar SCIA
scia_init();

print("Hola posicionamiento");

//Inicializar vector en este pone toda la información enviada por el puerto serial.

uint16_t j=0;
for(j=0;j<10;j++){
    vector[j]=0;
    cifras[j]=0;
}

ObtenerDatos(); //Llama a la función obtener datos.

//Iniciar funcionamiento

ObtenerMedida(); //Llama a la función Obtener medida.

//Llegar distancia inicial

if(medida<cuentasini)
    giro=1;
else
    giro=0;

```

```

//Compara si la medida tomada está por debajo de la distancia inicial ingresada

if(giro==1){
    while (medida <= cuentasini) {
        EfectuarMovimiento();
    }
}

//Compara si la medida tomada está por encima de la distancia inicial ingresada
else{
    while (medida >= cuentasini) {
        EfectuarMovimiento();
    }
}
print("Posicion inicial\r");

//Espera a que se inicie el movimiento

retardo_us(10000);

//Compara si la medida tomada está por debajo de la distancia de posicionamiento ingresada

if(medida<cuentas)
    giro=1;
else
    giro=0;

if(giro==1){

//Compara si la medida tomada es menor o igual a la distancia de posicionamiento ingresada

    while(medida <= cuentas){
        EfectuarMovimiento();
    }
}
else{

// Compara si la medida tomada está por encima de la distancia de posicionamiento ingresada
    while(medida >= cuentas){

```

```

        EfectuarMovimiento();
    }
}
print("Posicion final\r");

//Fin del programa.

//Inhabilita todas las interrupciones

PWM_disableInt(myPwm1);
PWM_disableInt(myPwm2);
PWM_clearIntFlag(myPwm1);
PWM_clearIntFlag(myPwm2);
CLK_disableTbClockSync(myClk);
PIE_clearInt(myPie, PIE_GroupNumber_3);
TIMER_stop(myTimer0);
TIMER_disableInt(myTimer0);
PIE_clearInt(myPie, PIE_GroupNumber_1);

InhabilitarCAP();
PIE_disable(myPie);
PIE_disableAllInts(myPie);
CPU_disableGlobalInts(myCpu);
CPU_clearIntFlags(myCpu);
}

// Recibe datos ingresados por el usuario desde la interfaz de java y los guarda en la variable vector.

void ObtenerDatos(void) {
    distanciaini=RecibirNumero();
    distancia=RecibirNumero();
    numero_pulsos=RecibirNumero();
    time=RecibirNumero();
    velocidad=RecibirNumero();
    AjustarVelocidad();
}

void ObtenerMedida(void) {
    timer0IntCount=0;           //Inicializa esta variable el 0
}

```



```

ConfigurarPIE();           //Configura todas las interrupciones necesarias
HabilitarCAP();           //Configura cap

    capturaCompleta=false; // Mientras sea falso continúa tomando medidas.

    GPIO_setHigh(myGpio, GPIO_Number_4);
    retardo_us(10);
    GPIO_setLow(myGpio, GPIO_Number_4);
    while(!capturaCompleta){}
    CAP_rearm(myCap);
    medida=valor;
    ImprimirNumero(valor);

    cuentasini=(uint32_t)((distanciaini*pendiente)+intercepto); //Calcula el valor de cuentas para la
    distancia inicial.
    cuentas=(uint32_t)((distancia*pendiente)+intercepto); //Calcula el valor de cuentas para
    llegar al posicionamiento.

}

// Pone el valor del TBPRD Y CMPA de acuerdo a la frecuencia seleccionada enviada por la interface.
void AjustarVelocidad(void){
    switch(velocidad){ //nota: velocidad es una variable global

        case 1: //Velocidad de 25 Hz

            EPWM1_TIMER_TBPRD=9375;
            EPWM1_MITAD_CMPA=4687;
            EPWM1_MITAD_CMPB=4687;

            EPWM2_TIMER_TBPRD=9375;
            EPWM2_MITAD_CMPA=4687;
            EPWM2_MITAD_CMPB=4687;

            break;

        case 2://Velocidad de 50 Hz

            EPWM1_TIMER_TBPRD=4687;
            EPWM1_MITAD_CMPA=2343;

```

```
EPWM1_MITAD_CMPB=2343;
```

```
EPWM2_TIMER_TBPRD=4687;
```

```
EPWM2_MITAD_CMPA=2343;
```

```
EPWM2_MITAD_CMPB=2343;
```

```
break;
```

```
case 3://Velocidad de 75 Hz
```

```
EPWM1_TIMER_TBPRD=3125;
```

```
EPWM1_MITAD_CMPA=1562;
```

```
EPWM1_MITAD_CMPB=1562;
```

```
EPWM2_TIMER_TBPRD=3125;
```

```
EPWM2_MITAD_CMPA=1562;
```

```
EPWM2_MITAD_CMPB=1562;
```

```
break;
```

```
case 4://Velocidad de 100Hz
```

```
EPWM1_TIMER_TBPRD=2343;
```

```
EPWM1_MITAD_CMPA=1171;
```

```
EPWM1_MITAD_CMPB=1171;
```

```
EPWM2_TIMER_TBPRD=2343;
```

```
EPWM2_MITAD_CMPA=1171;
```

```
EPWM2_MITAD_CMPB=1171;
```

```
break;
```

```
case 5: //Velocidad de 125 Hz
```

```
EPWM1_TIMER_TBPRD=1875;
```

```
EPWM1_MITAD_CMPA=937;
```

```
EPWM1_MITAD_CMPB=937;
```

```
EPWM2_TIMER_TBPRD=1875;
```

```
EPWM2_MITAD_CMPA=937;
```

```
EPWM2_MITAD_CMPB=937;
```

```
break;
```

```
case 6: //Velocidad de 137.5
```

```
EPWM1_TIMER_TBPRD=1704;
```

```
EPWM1_MITAD_CMPA=852;
```

```
EPWM1_MITAD_CMPB=852;
```

```
EPWM2_TIMER_TBPRD=1704;
```

```
EPWM2_MITAD_CMPA=852;
```

```
EPWM2_MITAD_CMPB=852;
```

```
break;
```

```
case 7 : //Velocidad de 150 Hz
```

```
EPWM1_TIMER_TBPRD=1562;
```

```
EPWM1_MITAD_CMPA=781;
```

```
EPWM1_MITAD_CMPB=781;
```

```
EPWM2_TIMER_TBPRD=1562;
```

```
EPWM2_MITAD_CMPA=781;
```

```
EPWM2_MITAD_CMPB=781;
```

```
break;
```

```
case 8: //Velocidad de 162.5
```

```
EPWM1_TIMER_TBPRD=1442;
```

```
EPWM1_MITAD_CMPA=721;
```

```
EPWM1_MITAD_CMPB=721;
```

```
EPWM2_TIMER_TBPRD=1442;
```

```
EPWM2_MITAD_CMPA=721;
```

```
EPWM2_MITAD_CMPB=721;
```

```
break;
```

```
case 9 : //Velocidad 175
```

```

        EPWM1_TIMER_TBPRD=1339;

        EPWM1_MITAD_CMPA=669;

        EPWM1_MITAD_CMPB=669;

        EPWM2_TIMER_TBPRD=1339;

        EPWM2_MITAD_CMPA=669;

        EPWM2_MITAD_CMPB=669;

        break;

    default:

        EPWM1_TIMER_TBPRD=1562;

        EPWM1_MITAD_CMPA=781;

        EPWM1_MITAD_CMPB=781;

        EPWM2_TIMER_TBPRD=1562;

        EPWM2_MITAD_CMPA=781;

        EPWM2_MITAD_CMPB=781;

        break;

    }

}

void ConfigurarPIE(void) {

    // la bandera hace la configuración de todas las interrupciones 1 sola vez

    PIE_enable(myPie);

    EALLOW;

    EDIS;

    //Registra los handler en el vector tabla de interrupciones por medio de ISR funciones de atención
    a interrupciones.

    PIE_registerPieIntHandler(myPie, PIE_GroupNumber_3, PIE_SubGroupNumber_1,
    (intVec_t) & epwm1_isr); //Habilita el grupo 3 int del pwm y llena la tabla con 2 int

    PIE_registerPieIntHandler(myPie, PIE_GroupNumber_3, PIE_SubGroupNumber_2,
    (intVec_t) & epwm2_isr);

    PIE_registerPieIntHandler(myPie, PIE_GroupNumber_4, PIE_SubGroupNumber_1, &ecap1_isr);
    //Habilita el grupo 4 int cap y llena la tabla con 1 interrupción

    PIE_registerPieIntHandler(myPie, PIE_GroupNumber_1, PIE_SubGroupNumber_7,
    (intVec_t)&cpu_timer0_isr); //Habilita las interrupciones del timer0 sub grupo 7

    EDIS;

}

```

```

void HabilitarCAP(void) {
    PWM_disableInt(myPwm1);           //Deshabilita las interrupciones PWM1
    PWM_disableInt(myPwm2);           //Deshabilita las interrupciones PWM2
    PWM_clearIntFlag(myPwm1);         //Limpia las banderas de las interrupciones PWM1
    PWM_clearIntFlag(myPwm2);         //Limpia las banderas de las interrupciones PWM2
    ConfigurarCAP();
    CPU_enableInt(myCpu, CPU_IntNumber_4); //Habilita las interrupciones del Cap
    PIE_enableCaptureInt(myPie);           //Habilitador para capturar la interrupción
    permite la comunicacion entre el micro y cada modulo.
    CPU_enableGlobalInts(myCpu);         //Habilita las interrupciones globales
    CPU_enableDebugInt(myCpu);
}

void EfectuarMovimiento(void) {
    timer0IntCount=0;
    //Toma medidas del cap
    capturaCompleta=false;
    GPIO_setHigh(myGpio, GPIO_Number_4);
    retardo_us(10);
    GPIO_setLow(myGpio, GPIO_Number_4);
    while (!capturaCompleta){}
    CAP_rearm(myCap);
    medida=valor;
    ImprimirNumero(valor);

    // Valida si el equipo llego al límite superior y suspende movimiento.
    if(valor>220000) {
        PWM_clearIntFlag(myPwm1);
        PWM_clearIntFlag(myPwm2);
        PWM_disableInt(myPwm1);
        PWM_disableInt(myPwm2);
        tope=1;
    }

    // Valida si el equipo llego al límite inferior y suspende movimiento.
    if(valor<40000) {
        PWM_clearIntFlag(myPwm1);
        PWM_clearIntFlag(myPwm2);
        PWM_disableInt(myPwm1);
        PWM_disableInt(myPwm2);
    }
}

```

```

    tope=1;
}

if(tope==0){ // si tope es igual a cero el equipo empieza el movimiento
    HabilitarTimer();
    //Configura PWM por primera vez
    HabilitarPWM();
}

//Espera a que se hagan los movimientos de n pasos
while(timer0IntCount!=npasos){} // recibe interrupciones del timer hasta que llega a 10
}

```

```

uint16_t RecibirNumero(void){
    char letra;
    uint16_t m=0;
    uint16_t j=0;
    uint16_t numero=0;
    uint16_t v=1;
    while((letra=scia_recibir())!='\r'){
        if(letra>47 && letra<59){
            vector[m]=(uint16_t)(letra-48);
            m++;
        }
    }
    for(j=0;j<m;j++){ //????
        numero=numero+(vector[m-j-1]*v);
        v=v*10;
    }
    n=m;
    return numero;
}

```

// Recibe el promedio y lo guarda en la variable vector.

```

uint32_t RecibirPromedio(void){
    char letra;

```

```

uint16_t m=0;
uint16_t j=0;
uint32_t numero=0;
uint32_t v=1;
while((letra=scia_recibir())!='\r'){
    if(letra>47 && letra<59){
        vector[m]=(uint16_t)(letra-48);
        m++;
    }
}

//Saca cada una de las cifras del vector
for(j=0;j<m;j++){
    numero=numero+((uint32_t)vector[m-j-1]*v);
    v=v*10;
}
n=m;
return numero;
}

// imprime la información del vector sin separar unidades decenas cent
// Guarda los datos enviados desde java en las diferentes variables
// Imprime un numero diferenciando unidades decenas etc

void ImprimirNumero(uint32_t numero){
    uint32_t modulo=10;
    uint32_t divisor=1;
    uint16_t i=0;
    p=0;
    while(numero/divisor!=0){
        modulo=modulo*10;
        divisor=divisor*10;
        p++;
    }
    modulo=10;
    divisor=1;
    for(i=0;i<p;i++){
        cifras[i]=(uint16_t)((numero%modulo)/divisor);
        modulo=modulo*10;
    }
}

```

```

        divisor=divisor*10;
    }
    for(i=0;i<p;i++){
        scia_xmit(cifras[p-i-1]+48);
    }
    scia_xmit('\r');
}

//Retardo aproximado de 1 ms

void retardo_us(uint16_t i){
    uint16_t j=0;
    for(j=0;j<i;j++){
        __asm(" NOP");
    }
}

//Transmitir un carácter imprime una letra o un solo numero

void scia_xmit(char a) {
    while (SCI_getTxFifoStatus(mySci) != SCI_FifoStatus_Empty) {
    }
    SCI_putDataBlocking(mySci, a);
}

//Transmitir un mensaje

void print(char * msg) {
    int i;
    i = 0;
    while (msg[i] != '\0') {
        scia_xmit(msg[i]);
        i++;
    }
}

//Recibir un carácter

char scia_recibir(void) {

```



```

    char a;

    a = SCI_getDataBlocking(mySci);

    if (a != NULL)

        return a;

    else

        return NULL;

}

//Atención a interrupción del módulo eCap

interrupt void ecap1_isr(void){

    valor=CAP_getCap1(myCap);

    CAP_clearInt(myCap, CAP_Int_Type_CEVT1);

    CAP_clearInt(myCap, CAP_Int_Type_Global);

    PIE_clearInt(myPie, PIE_GroupNumber_4);

    capturaCompleta=true;

}

// Atención a la interrupción del PWM

// Frec. PWM = 500 Hz

__interrupt void epwm1_isr(void) {

    if (epwm1_info.EPwmTimerIntCount == numero_pulsos) {

        PWM_setCmpA(myPwm1,0x0000); //Duty del 100% porcentaje de la señal que se encuentra en un
siclo

        PWM_setCmpB(myPwm1,0x0000);

        PWM_setActionQual_CntUp_CmpB_PwmB(myPwm1, PWM_ActionQual_Clear); //.. mira si se encuentra en
un valor medio entre 0 y T genera cambio de estado.

        PWM_setActionQual_CntDown_CmpB_PwmB(myPwm1, PWM_ActionQual_Clear); // MOD AQ toma decisión si
la señal se pone alto o bajo comp PWA con el valor del contador cc

        //Fase 180°. Empieza en 0 y termina en 1

        PWM_setActionQual_CntUp_CmpA_PwmA(myPwm1, PWM_ActionQual_Clear); // Set PWM1B on event B, up
count

        PWM_setActionQual_CntDown_CmpA_PwmA(myPwm1, PWM_ActionQual_Clear); // Clear PWM1B on event B,
down count

        PWM_disableInt(myPwm1);

    } else {

        epwm1_info.EPwmTimerIntCount++;

    }

    PWM_clearIntFlag(myPwm1);

    PIE_clearInt(myPie, PIE_GroupNumber_3);

```

```

}

__interrupt void epwm2_isr(void) {
    if(epwm2_info.EPwmTimerIntCount == numero_pulsos){
        PWM_setCmpA(myPwm2,0x0000);
        PWM_setCmpB(myPwm2,0x0000);
        //Fase 90°
        PWM_setActionQual_Zero_PwmA(myPwm2, PWM_ActionQual_Clear);
        PWM_setActionQual_Period_PwmA(myPwm2, PWM_ActionQual_Clear);
        //Fase 270°
        PWM_setActionQual_Zero_PwmB(myPwm2, PWM_ActionQual_Clear);
        PWM_setActionQual_Period_PwmB(myPwm2, PWM_ActionQual_Clear);
        PWM_disableInt(myPwm2);
    }
    else{
        epwm2_info.EPwmTimerIntCount++;
    }
    PWM_clearIntFlag(myPwm2);
    PIE_clearInt(myPie, PIE_GroupNumber_3);
}

```

//Atención a interrupción del timer

```

__interrupt void cpu_timer0_isr(void){

    if(timer0IntCount==npasos){
        //Inhabilita PWM
        PWM_disableInt(myPwm1);
        PWM_disableInt(myPwm2);
        PWM_clearIntFlag(myPwm1);
        PWM_clearIntFlag(myPwm2);
        CLK_disableTbClockSync(myClk);
        PIE_clearInt(myPie, PIE_GroupNumber_3);
        //Inhabilita Timer
        TIMER_stop(myTimer0);
        TIMER_disableInt(myTimer0);
        PIE_clearInt(myPie, PIE_GroupNumber_1);
        //Hace que el timer pare
    }
}

```

```

        timer0IntCount=npasos+1;
    }
    else{
        timer0IntCount++;
        HabilitarPWM();
    }
}

//Inicialización de la comunicación serial
void scia_init(){
    CLK_enableSciaClock(myClk); // Habilita el reloj de la comunicación serial
    SCI_disableParity(mySci); // Deshabilita la paridad
    SCI_setNumStopBits(mySci, SCI_NumStopBits_One);
    SCI_setCharLength(mySci, SCI_CharLength_8_Bits); //Selecciona longitud del paquete 8 bits
    SCI_enableTx(mySci); // Habilita la transmisión.
    SCI_enableRx(mySci); // Habilita la recepción
    SCI_setBaudRate(mySci, SCI_BaudRate_115_2_kBaud); // fija la velocidad de transmisión y recepción.
    SCI_enable(mySci); // Habilita el modulo serial
    return;
}

//Configura las interrupciones del timer 0

void HabilitarTimer(void) {
    TIMER_stop(myTimer0);

    TIMER_setPeriod(myTimer0, 60 * 1000000*time); //esta línea multiplicada por una cte da el valor de
    donde interrumpe
    TIMER_setPreScaler(myTimer0, 0);
    TIMER_reload(myTimer0);
    TIMER_setEmulationMode(myTimer0, TIMER_EmulationMode_StopAfterNextDecrement); // no se hacen
    emulaciones
    TIMER_enableInt(myTimer0); //Habilita la interrupción del timer 0
    TIMER_start(myTimer0); //configura el bit del TSS en 0
    CPU_enableInt(myCpu, CPU_IntNumber_1); //habilita la interrupción de la CPU timer0
    PIE_enableTimer0Int(myPie); // Habilita la interrupción o en el módulo PIE
}

//Genera el movimiento del equipo habilita todos los módulos necesarios para esto.

void HabilitarPWM(void) {

```

```

CLK_disableTbClockSync(myClk); // inhabilita la vble tbclksync detiene el PWM

// si giro =1 el equipo se mueve hacia arriba si no hacia abajo.

if (giro==0)
    Configurar_ePWM_arriba();
else
    Configurar_ePWM_abajo();

CLK_enableTbClockSync(myClk); // habilita tbclksync empieza el pwm
CPU_enableInt(myCpu, CPU_IntNumber_3); // Habilita interrupciones CPU
PIE_enablePwmInt(myPie, PWM_Number_1); //habilita una interrupción
PIE_enablePwmInt(myPie, PWM_Number_2);
PWM_enableInt(myPwm1); //habilita la interrupción pw1
PWM_enableInt(myPwm2);
PIE_clearInt(myPie, PIE_GroupNumber_1);
}

// Deshabilita el módulo Cap las interrupciones y el contador.

void InhabilitarCAP(void) {
    //Inhabilitar CAP
    CAP_disableInt(myCap, CAP_Int_Type_All);
    CAP_disableCaptureLoad(myCap);
    CAP_disableTimestampCounter(myCap);
    CAP_clearInt(myCap, CAP_Int_Type_All);
}

void Configurar_ePWM_abajo(void) {

    //Configurar_ePWM1
    CLK_enablePwmClock(myClk, PWM_Number_1);

    //Setup TBCLK
    PWM_setPeriod(myPwm1, EPWM1_TIMER_TBPRD); //Envía el TBPRD
    PWM_setPhase(myPwm1, 0x0000); //
    PWM_setCount(myPwm1, 0x0000); //

    //Submódulo CC configurado a 50% de duty cycle ....comparador contador mod cc

```

```

PWM_setCmpA(myPwm1, EPWM1_MITAD_CMPA); //0
PWM_setCmpB(myPwm1, EPWM1_MITAD_CMPB); //180

// Setup counter mode
PWM_setCounterMode(myPwm1, PWM_CounterMode_UpDown); // Conteo UP-DOWN para PWM simétrico
PWM_disableCounterLoad(myPwm1); //PHSEN=0
PWM_setHighSpeedClkDiv(myPwm1, PWM_HspClkDiv_by_1); //
PWM_setClkDiv(myPwm1, PWM_ClkDiv_by_128); //Prescaler

// Setup shadowing
PWM_setShadowMode_CmpA(myPwm1, PWM_ShadowMode_Shadow); //
PWM_setShadowMode_CmpB(myPwm1, PWM_ShadowMode_Shadow); //
PWM_setLoadMode_CmpA(myPwm1, PWM_LoadMode_Zero); //
PWM_setLoadMode_CmpB(myPwm1, PWM_LoadMode_Zero); //

//PWM1 como maestro
PWM_setSyncMode(myPwm1, PWM_SyncMode_CounterEqualZero);

//Fase 0°. Empieza en 1 y termina en 0
PWM_setActionQual_CntUp_CmpA_PwmA(myPwm1, PWM_ActionQual_Set); //.. mira si se encuentra en un
valor medio entre 0 y T genera cambio de estado.
PWM_setActionQual_CntDown_CmpA_PwmA(myPwm1, PWM_ActionQual_Clear); // MOD AQ toma decisión si la
señal se pone alto o bajo comp PWA con el valor del contador cc

//Fase 180°. Empieza en 0 y termina en 1
PWM_setActionQual_CntUp_CmpB_PwmB(myPwm1, PWM_ActionQual_Clear); // Set PWM1B on event B, up count
PWM_setActionQual_CntDown_CmpB_PwmB(myPwm1, PWM_ActionQual_Set); // Clear PWM1B on event B, down
count

PWM_setIntMode(myPwm1, PWM_IntMode_CounterEqualZero); // Select INT on Zero event
PWM_enableInt(myPwm1);
// Enable INT
PWM_setIntPeriod(myPwm1, PWM_IntPeriod_FirstEvent); // Generate INT on 3rd event

epwm1_info.EPwm_CMPA_Direction = EPWM_CMP_UP;//
epwm1_info.EPwm_CMPB_Direction = EPWM_CMP_UP;
epwm1_info.myPwmHandle = myPwm1;
epwm1_info.EPwmTimerIntCount = 0; // Zero the interrupt counter

// Configurar_ePWM2
CLK_enablePwmClock(myClk, PWM_Number_2);

```

```

// Setup TBCLK
PWM_setPeriod(myPwm2, EPWM2_TIMER_TBPRD); // Acá manda TBPRD
PWM_setPhase(myPwm2, 0x0000); // No se hace nada acá
PWM_setCount(myPwm2, 0x0000); // No se hace nada acá

//Submódulo CC configurado a 50% de duty cycle
PWM_setCmpA(myPwm2, EPWM2_MITAD_CMPA); // Set compare A value
PWM_setCmpB(myPwm2, EPWM2_MITAD_CMPA); // Set Compare B value

// Setup counter mode
PWM_setCounterMode(myPwm2, PWM_CounterMode_UpDown); // Conteo UP-DOWN para PWM simétrico
PWM_enableCounterLoad(myPwm2); // No se hace nada acá
PWM_setHighSpeedClkDiv(myPwm2, PWM_HspClkDiv_by_1); // No se hace nada acá
PWM_setClkDiv(myPwm2, PWM_ClkDiv_by_128); //Prescaler

// Setup shadowing // No se hace nada acá
PWM_setShadowMode_CmpA(myPwm2, PWM_ShadowMode_Shadow); // sombra
PWM_setShadowMode_CmpB(myPwm2, PWM_ShadowMode_Shadow);
PWM_setLoadMode_CmpA(myPwm2, PWM_LoadMode_Zero);
PWM_setLoadMode_CmpB(myPwm2, PWM_LoadMode_Zero);

// //Fase 90°. Elmpieza en 1 y termina en 0
PWM_setActionQual_Zero_PwmA(myPwm2, PWM_ActionQual_Clear); // Set PWM2A on event A, up count.
PWM_setActionQual_Period_PwmA(myPwm2, PWM_ActionQual_Set); // Clear PWM2A on event B, down count
//
// //Fase 270°. Empieza en 0 y termina en 1
PWM_setActionQual_Zero_PwmB(myPwm2, PWM_ActionQual_Set); // Clear PWM2B on zero
PWM_setActionQual_Period_PwmB(myPwm2, PWM_ActionQual_Clear); // Set PWM2B on period

//Fase 0°. Empieza en 1 y termina en 0
PWM_setActionQual_CntUp_CmpA_PwmA(myPwm1, PWM_ActionQual_Set);
PWM_setActionQual_CntDown_CmpA_PwmA(myPwm1, PWM_ActionQual_Clear);

//Fase 180°. Empieza en 0 y termina en 1
PWM_setActionQual_CntUp_CmpB_PwmB(myPwm1, PWM_ActionQual_Clear); // Set PWM1B on event B, up count
PWM_setActionQual_CntDown_CmpB_PwmB(myPwm1, PWM_ActionQual_Set); // Clear PWM1B on event B, down
count

//Poner a este PWM como esclavo

```

```

PWM_setSyncMode(myPwm2, PWM_SyncMode_EPWMxSYNC);

PWM_setIntMode(myPwm2, PWM_IntMode_CounterEqualZero); // Select INT on Zero event

PWM_enableInt(myPwm2); // Enable INT

PWM_setIntPeriod(myPwm2, PWM_IntPeriod_FirstEvent); // Generate INT on 3rd event

////

epwm2_info.EPwmTimerIntCount = 0;

epwm2_info.EPwm_CMPA_Direction = EPWM_CMP_UP; // Start by increasing CMPA &
epwm2_info.EPwm_CMPB_Direction = EPWM_CMP_UP; // increasing CMPB

epwm2_info.myPwmHandle = myPwm2;
}

void Configurar_ePWM_arriba(void) {

//Configurar_ePWM1

CLK_enablePwmClock(myClk, PWM_Number_1);

// Setup TBCLK

PWM_setPeriod(myPwm1, EPWM1_TIMER_TBPRD); //Envia el valor de TBPRD
PWM_setPhase(myPwm1, 0x0000); //Pone la fase en cero
PWM_setCount(myPwm1, 0x0000); //Pone la variable contador del pwm1 en cero.

//Submódulo CC configurado a 50% de duty cycle
PWM_setCmpA(myPwm1, EPWM1_MITAD_CMPA); //0
PWM_setCmpB(myPwm1, EPWM1_MITAD_CMPB); //180

// Setup counter mode
PWM_setCounterMode(myPwm1, PWM_CounterMode_UpDown); // Conteo UP-DOWN para PWM simétrico
PWM_disableCounterLoad(myPwm1); //Deshabilita la variable Counter Pwm1
PWM_setHighSpeedClkDiv(myPwm1, PWM_HspClkDiv_by_1);
PWM_setClkDiv(myPwm1, PWM_ClkDiv_by_128); // Define el Prescaler 128

// Setup shadowing

PWM_setShadowMode_CmpA(myPwm1, PWM_ShadowMode_Shadow); //Pone el efecto sombra en los CmpA y CmpB
del PWM1.

PWM_setShadowMode_CmpB(myPwm1, PWM_ShadowMode_Shadow);
PWM_setLoadMode_CmpA(myPwm1, PWM_LoadMode_Zero);
PWM_setLoadMode_CmpB(myPwm1, PWM_LoadMode_Zero);

//PWM1 como maestro

```

```

    PWM_setSyncMode(myPwm1, PWM_SyncMode_CounterEqualZero);

    //Fase 0°. Empieza en 1 y termina en 0

    PWM_setActionQual_CntUp_CmpB_PwmB(myPwm1, PWM_ActionQual_Set); //Mira si se encuentra en un
    valor medio entre 0 y T genera cambio de estado.

    PWM_setActionQual_CntDown_CmpB_PwmB(myPwm1, PWM_ActionQual_Clear); //MOD AQ toma decisión si la
    señal se pone alto o bajo comp PWA con el valor del contador cc

    //Fase 180°. Empieza en 0 y termina en 1

    PWM_setActionQual_CntUp_CmpA_PwmA(myPwm1, PWM_ActionQual_Clear); //Set PWM1B on event B, up
    count

    PWM_setActionQual_CntDown_CmpA_PwmA(myPwm1, PWM_ActionQual_Set); //Limpia Clear PWM1B on event
    B, down count

    PWM_setIntMode(myPwm1, PWM_IntMode_CounterEqualZero); //Pone la
    interrupcion en el evento cero.Select INT on Zero event

    PWM_enableInt(myPwm1);
        //Enable las interrupciones Pwm1

    PWM_setIntPeriod(myPwm1, PWM_IntPeriod_FirstEvent); //Genera
    la interrupcion en el 3er evento

    epwm1_info.EPwm_CMPA_Direction = EPWM_CMP_DOWN;

    epwm1_info.EPwm_CMPB_Direction = EPWM_CMP_DOWN;

    epwm1_info.myPwmHandle = myPwm1;

    epwm1_info.EPwmTimerIntCount = 0; //Pone en
    cero el contador de interrupcion. Zero the interrupt counter

    // Configurar_ePWM2

    CLK_enablePwmClock(myClk, PWM_Number_2);

    // Setup TBCLK

    PWM_setPeriod(myPwm2, EPWM2_TIMER_TBPRD);
        //Envia el valor de TBPRD

    PWM_setPhase(myPwm2, 0x0000); //Pone la fase
    del PWM2 en cero

    PWM_setCount(myPwm2, 0x0000); //Pone la variable Count del Pwm2
    en cero

    //Submódulo CC configurado a 50% de duty cycle

    PWM_setCmpA(myPwm2, EPWM2_MITAD_CMPA); //Pone el valor del comparador A

    PWM_setCmpB(myPwm2, EPWM2_MITAD_CMPA); //Set Compare B value // Pone
    el valor del comparador B

    // Setup counter mode

    PWM_setCounterMode(myPwm2, PWM_CounterMode_UpDown); //Conteo UP-
    DOWN para PWM simétrico

```



```

Pwm2 PWM_enableCounterLoad(myPwm2); //Habilita la variable contador del

PWM_setHighSpeedClkDiv(myPwm2, PWM_HspClkDiv_by_1);

PWM_setClkDiv(myPwm2, PWM_ClkDiv_by_128); //Pone el prescaler de 128

// Setup shadowing // No se hace nada acá

PWM_setShadowMode_CmpA(myPwm2, PWM_ShadowMode_Shadow); //Sombra
PWM_setShadowMode_CmpB(myPwm2, PWM_ShadowMode_Shadow);

PWM_setLoadMode_CmpA(myPwm2, PWM_LoadMode_Zero);
PWM_setLoadMode_CmpB(myPwm2, PWM_LoadMode_Zero);

//Fase 90°. Empieza en 1 y termina en 0 OJO ES CORRECTO
PWM_setActionQual_Zero_PwmB(myPwm2, PWM_ActionQual_Clear); //Set PWM2A on event A, up
count.
PWM_setActionQual_Period_PwmB(myPwm2, PWM_ActionQual_Set); //Clear PWM2A on event B,
down count

//Fase 270°. Empieza en 0 y termina en 1
PWM_setActionQual_Zero_PwmA(myPwm2, PWM_ActionQual_Set); //Limpia PWM2B en zero
PWM_setActionQual_Period_PwmA(myPwm2, PWM_ActionQual_Clear); //Pone en PWM2B un periodo

//Fase 0°. Empieza en 1 y termina en 0
PWM_setActionQual_CntUp_CmpA_PwmA(myPwm1, PWM_ActionQual_Set); //Genera una señal de Pwm1
CmpA en 1
PWM_setActionQual_CntDown_CmpA_PwmA(myPwm1, PWM_ActionQual_Clear); //Baja la señal de Pwm1 CmpA a 0

//Fase 180°. Empieza en 0 y termina en 1
PWM_setActionQual_CntUp_CmpB_PwmB(myPwm1, PWM_ActionQual_Clear); //Baja la señal de Pwm1 CmpB a 0
PWM_setActionQual_CntDown_CmpB_PwmB(myPwm1, PWM_ActionQual_Set); //Baja la señal de Pwm1 CmpB a 1

//Poner a este PWM como esclavo
PWM_setSyncMode(myPwm2, PWM_SyncMode_EPWMxSYNC);

PWM_setIntMode(myPwm2, PWM_IntMode_CounterEqualZero); //Select INT en el
evento Zero.

PWM_enableInt(myPwm2); //Habilita la
interrupcion Pwm2

PWM_setIntPeriod(myPwm2, PWM_IntPeriod_FirstEvent); //Genera la
interrupcion en el 3rd evento

epwm2_info.EPwmTimerIntCount = 0;

epwm2_info.EPwm_CMPA_Direction = EPWM_CMP_DOWN; // Empieza incrementando el CMPA y
epwm2_info.EPwm_CMPB_Direction = EPWM_CMP_DOWN; // Incrementa CMPB

epwm2_info.myPwmHandle = myPwm2;

```

```

}

void ConfigurarCAP(void) {

    CLK_enableEcap1Clock(myClk);           //Habilita el modulo Cap
    CAP_disableInt(myCap, CAP_Int_Type_All); //Deshabilita las interrupciones del Cap
    CAP_clearInt(myCap, CAP_Int_Type_All);  //Limpia las interrupciones Cap
    CAP_disableCaptureLoad(myCap);         //Deshabilita el modulo Cap
    CAP_disableTimestampCounter(myCap);     //Deshabilita el contador del modulo Cap

    CAP_setCapContinuous(myCap);           //Permite tomar medidas del Cap varias veces
    CAP_setStopWrap(myCap, CAP_Stop_Wrap_CEVTL);

    CAP_setCapEvtPolarity(myCap, CAP_Event_1, CAP_Polarity_Falling);
    CAP_setCapEvtReset(myCap, CAP_Event_1, CAP_Reset_Enable);

    CAP_enableSyncIn(myCap);               //Habilita el sincronismo de entrada módulo
Cap
    CAP_setSyncOut(myCap, CAP_SyncOut_SyncIn); //Activa el sincronismo de entrada y salida módulo Cap

    CAP_enableCaptureLoad(myCap);         //Habilita el módulo Cap

    CAP_enableTimestampCounter(myCap);     //Habilita el contador de tiempo módulo Cap
    CAP_enableCaptureLoad(myCap);         // ?????? POR QUE SE REPITE DOS VECES
    CAP_enableInt(myCap, CAP_Int_Type_CEVTL); //Habilita las interrupciones módulo Cap
}

```

5.2. Programa comentado interfaz gráfica en Java.

En java se crearon tres clases o archivos organizados de la siguiente manera

5.2.1. Código clase Com_micro

Esta clase Com_micro es la encargada de realizar las configuraciones necesarias para la comunicación serial.

```

package dipcoating;

import gnu.io.CommPortIdentifier;

import gnu.io.PortInUseException;

import gnu.io.SerialPort;

```

```

import gnu.io.UnsupportedCommOperationException;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Enumeration;
import java.util.TooManyListenersException;

public class Com_micro {

    public static String      entrada;
    static String             puerto;
    static boolean            puertoEncontrado;
    static Enumeration        puertoLista;
    static CommPortIdentifier puertoId;
    static SerialPort         puertoSerial;
    static InputStream        inStream;
    static OutputStream        outStream;
    static boolean            activado=false;

    public static void IniciarCOM() throws TooManyListenersException{

//Buscando puertos seriales disponibles
puertoLista = CommPortIdentifier.getPortIdentifiers(); //Lista los puertos COM en el computador

        while (puertoLista.hasMoreElements()) {

            puertoId = (CommPortIdentifier) puertoLista.nextElement();

            if (puertoId.getPortType() == CommPortIdentifier.PORT_SERIAL) {

                if (puertoId.getName().equals(puerto)) {

//Si el puerto deseado se encuentra...

                    System.out.println("\nPuerto " + puerto + " encontrado.");

                    puertoEncontrado = true;

```

```

//Abriendo puerto deseado.
    try {
        puertoSerial =
            (SerialPort) puertoId.open("ventanaComunicacion", 2000);
    } catch (PortInUseException e) {
        System.out.println("\nPuerto en uso.\n\n");
        continue;
    }

//Abriendo los streams para la transmisión y recepción de datos
    try {
        inStream = puertoSerial.getInputStream();
    } catch (IOException e) {}

    try {
        outputStream = puertoSerial.getOutputStream();
    } catch (IOException e) {}

//Estableciendo los parámetros de la comunicación
    try {
        puertoSerial.setSerialPortParams(115200,
            SerialPort.DATABITS_8,
            SerialPort.STOPBITS_1,
            SerialPort.PARITY_NONE);
    } catch (UnsupportedCommOperationException e) {}

//Agregando soporte para generar interrupciones por datos entrantes.
    try {
        puertoSerial.addEventListener(new LectorCom(inStream));
    } catch (TooManyListenersException ex) {
        System.out.println("\nFallo al agregar las interrupciones.\n\n");
        System.exit(-1);
    }

    try {
        puertoSerial.notifyOnOutputEmpty(true);
    } catch (Exception e) {
        System.out.println("\nError al notificar la interrupción.\n\n");
        System.out.println(e.toString());
        System.exit(-1);
    }
}

```



```

String fin="\r";
public static String mensajeRx;

//constructor de la clase
public ventana_control(){
    initComponents();
}

//Inicializa y configura el puerto serial
private void PosicionActionPerformed(java.awt.event.ActionEvent evt) {

    //Selección del puerto utilizado
    Com_micro.puerto=String.valueOf(ListaPuertos.getSelectedItem());

// Bloque de tratar de inicializar puerto serial
try {
    Com_micro.IniciarCOM(); // llama a la clase Com_micro para inicializar el puerto
} catch (TooManyListenersException ex) {
    System.out.println("Puerto en uso.");
    //System.exit(-1); // Si falla, el programa termina
}

boolean bandera=true;
mensaje1=distanciainicial.getText(); //lee número de pulsos ingresado en el espacio

if(Integer.valueOf(mensaje1)<10 || Integer.valueOf(mensaje1)>47){
    distanciainicial.setText("Valor incorrecto");
    bandera=false;
}
else{
    mensaje1=String.valueOf(Integer.valueOf(mensaje1)-5);
    // Envía los datos de mensaje2 hacia el puerto.
    try {
        Com_micro.getOutputStream().write(mensaje1.getBytes());
        Com_micro.getOutputStream().write(fin.getBytes());
    } catch (IOException ex) {
        // Si falla, imprime mensaje en la consola
        System.out.println("Fallo al enviar bits");
    }
}

```

```

// retardo de 100 ms
try {
    Thread.sleep(100);
} catch (InterruptedException ex) {}
}

if(bandera==true){

    mensaje2=distanciafinal.getText(); //lee número de pulsos ingresado en
    el espacio.

    if(Integer.valueOf(mensaje2)<10 || Integer.valueOf(mensaje2)>47){
        distanciafinal.setText("Valor incorrecto");
        bandera=false;
    }
    else{
// Envía los datos de mensaje2 hacia el puerto.
        mensaje2=String.valueOf(Integer.valueOf(mensaje2)-5);

        try {
            Com_micro.getOutputStream().write(mensaje2.getBytes());
            Com_micro.getOutputStream().write(fin.getBytes());
        } catch (IOException ex) {
            // Si falla, imprime mensaje en la consola
            System.out.println("Fallo al enviar bits");
        }
        //Retardo de 100 ms
        try {
            Thread.sleep(100);
        } catch (InterruptedException ex) {}
    }
}

if(bandera==true){

    mensaje3=numeropulsos.getText(); //Lee numero de pulsos ingresado en
    el espacio
    if(Integer.valueOf(mensaje3)<1 || Integer.valueOf(mensaje3)>999){
        numeropulsos.setText("Valor incorrecto");
        bandera=false;
    }
}

```

```

}else{
    // Envia los datos de mensaje2 hacia el puerto.
    try {
        Com_micro.getOutputStream().write(mensaje3.getBytes());
        Com_micro.getOutputStream().write(fin.getBytes());
    } catch (IOException ex) {
        // Si falla, imprime mensaje en la consola
        System.out.println("Fallo al enviar bits");
    }
    // Pone un retardo de 100 ms
    try {
        Thread.sleep(100);
    } catch (InterruptedException ex) {}
}

}

if(bandera==true){
    mensaje4=tiempopausa.getText(); //lee número de pulsos ingresado en
                                   el espacio

    if(Integer.valueOf(mensaje4)<1 || Integer.valueOf(mensaje4)>60){
        tiempopausa.setText("Valor incorrecto");
    }else{
        // Envia los datos de mensaje4 hacia el puerto.
        try {
            Com_micro.getOutputStream().write(mensaje4.getBytes());
            Com_micro.getOutputStream().write(fin.getBytes());
        } catch (IOException ex) {
            // Si falla, imprime mensaje en la consola
            System.out.println("Fallo al enviar bits");
        }
        // Pone un retardo de 100 ms
        try {
            Thread.sleep(100);
        } catch (InterruptedException ex) {}
    }
}

```



```

}

if(bandera==true){
    mensaje5=String.valueOf(Velocidad.getSelectedItem()); // lee número de
pulsos ingresado en el espacio.
    // Envía los datos de mensaje 5 " velocidad" hacia el puerto.
    switch(mensaje5){

        case "2.5": //Velocidad de 25 Hz
            mensaje5="1";
            break;
        case "5": //Velocidad de 50 Hz
            mensaje5="2";
            break;
        case "7.5": //Velocidad de 75 Hz
            mensaje5="3";
            break;
        case "10": //Velocidad de 100Hz
            mensaje5="4";
            break;
        case "12.5": //Velocidad de 125 Hz
            mensaje5="5";
            break;
        case "13.75": //Velocidad de 137.5
            mensaje5="6";
            break;
        case "15" : //Velocidad de 150 Hz
            mensaje5="7";
            break;
        case "16.25": //Velocidad de 162.5
            mensaje5="8";
            break;
        case "17.5": //Velocidad 175
            mensaje5="9";
            break;
        default:
            mensaje5="7";
    }
}

```

```

        break;
    }

    try {
        Com_micro.getOutputStream().write(mensaje5.getBytes());
        Com_micro.getOutputStream().write(fin.getBytes());
    } catch (IOException ex) {
        // Si falla, imprime mensaje en la consola
        System.out.println("Fallo al enviar bits");
    }

    // Pone un retardo de 100 ms
    try {
        Thread.sleep(100);
    } catch (InterruptedException ex) {}
}

if(Com_micro.activado==true){
    try {
        Com_micro.getOutputStream().close();
    } catch (IOException ex) {
        Logger.getLogger(ventana_control.class.getName()).log(Level.SEVERE, null, ex);
    }

    try {
        Com_micro.getInputStream().close();
    } catch (IOException ex) {
        Logger.getLogger(ventana_control.class.getName()).log(Level.SEVERE, null, ex);
    }

    Com_micro.puertoSerial.close();
}
}

```

5.2.3. Código de Clase LectorCom

```

import static dipcoating.ventana_control.banderaSerial;
import gnu.io.SerialPortEvent;
import gnu.io.SerialPortEventListener;

```

```

import java.io.IOException;
import java.io.InputStream;

public class LectorCom implements SerialPortEventListener{

    private InputStream in;

    private byte[] buffer = new byte[64];

    public LectorCom (InputStream in){
        this.in = in;
    }

    public void serialEvent(SerialPortEvent arg0){

        int dato;
        int len = 0;
        try {
            while ((dato = in.read())> -1){ //Mientras haya datos para leer...
                if(dato=='\r'){
                    break;
                }else{
                    buffer[len++]=(byte) dato;
                }
            }

            ventana_control.mensajeRx=new String(buffer);
            for(int j=0;j<64;j++){
                buffer[j]=0;
            }
            System.out.println(ventana_control.mensajeRx);

        } catch (IOException ex) {
            System.exit(-1);
        }
    }
}

```