



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA
MAESTRÍA EN CIENCIAS DE LA ELECTRÓNICA
OPCIÓN EN AUTOMATIZACIÓN

“Robot CNC para impresora 3D”**

T E S I S

Presentada para obtener el título de:
Maestro en Ciencias de la Electrónica Opción en Automatización

Presenta:

Lic. Alfredo López Beltrán*

Directores:

Dr. Sergio Vergara Limon

Dra. Amparo D. Palomino Merino

Dr. Fernando Reyes Cortés

Puebla, México

Diciembre 2016

*BECARIO CONACYT

** TRABAJO FINANCIADO POR EL PROYECTO VIEP

BUAP®

Agradecimientos

Realmente no tengo palabras para agradecer a todas las personas que me apoyaron para la realización de esta tesis, ya que hubo muchos contratiempos, sin embargo, nunca dejaron de animarme a seguir adelante, y sin ellos quizá otra historia sería.

A mi amigo de toda la vida Ricardo y a mis nuevos amigos Luis, Ricardo Torres, Juan, Daniel, Miguel, Oswaldo, Froylan y Araceli.

A mis asesores, Dr. Fernando Reyes Cortés y la Dra. Amparo D. Palomino Merino, resaltando a mi asesor principal, que con sus enseñanzas y regaños fui aprendiendo de él, Dr. Sergio Vergara Limon.

Le agradezco a mis sinodales por contribuir con sus conocimientos y consejos académicos para la correcta elaboración del presente trabajo. Destacando a la Dra. Aurora Vargas Treviño quién no dejó de apoyarme en todo momento.

A mis profesores de la maestría por darme las herramientas a lo largo de los estudios de posgrados.

A mi familia que siempre me motivo a seguir adelante y no bajar los brazos, gracias Mamá, Papá, a mi hermana Lorena y mi otra hermana Adriana. A mi novia Eldy que conocí en la maestría, y me apoyo con todo su cariño.

Al CONACYT por el apoyo económico que me dio para mi estancia en esta maestría y apoyar creación de tecnología y nuevos investigadores.

A la VIEP por haber financiado este trabajo de tesis en su totalidad y así haber llegado a una excelente culminación de esta.

A la facultad de ciencias de la electrónica de la BUAP, por brindarme la oportunidad de adquirir los conocimientos que ahora tengo y poder concluir satisfactoriamente mis estudios de posgrados.

CONTENIDO

Introducción	i
Capítulo 1. Descripción del Robot CNC.....	1
1.1 Características generales.....	1
1.2 Modelo dinámico del robot cartesiano.....	2
1.3 Mecánica del robot.....	12
1.4 Electrónica del robot.....	25
1.5 Conclusiones.....	34
Capítulo 2. Firmware del sistema de movimiento.....	35
2.1 División del sistema.....	35
2.2 Sistema de transferencia de datos.....	35
2.3 Sistema de conteo.....	37
2.4 Generación de la señal PWM.....	41
2.5 Conclusiones.....	43
Capítulo 3. Software de control de movimiento de los ejes X, Y y Z.....	44
3.1 Funcionamiento de software inicial.....	44
3.5 Conclusiones.....	49
Capítulo 4. Instrumentación de la herramienta.....	50
4.1 Extrusor de material.....	51
4.1.1 Firmware de control de temperatura del extrusor de material.....	57
4.2 Base caliente.....	59
4.3 Alimentador de material.....	61
4.3.1 Firmware de control de velocidad del alimentador.....	62
4.4 Conclusiones.....	65
Capítulo 5. Software de la impresora 3D.....	66
5.1 Laminador y compilador.....	71

5.2 Parámetros de la herramienta de impresión 3D.....	80
5.3 Conclusiones.....	84
Capítulo 6. Resultados experimentales.....	85
6.1 Resultados de control del CNC.....	85
CONCLUSIONES GENERALES.....	90
REFERENCIAS.....	91

Apéndice A

Constancia de ponencia

Apéndice B

Artículo

Apéndice C

TOEFL

Introducción

En la actualidad, los robots son muy importantes en diferentes áreas, tanto productivas como de la salud e investigación, ya que tienen la versatilidad de cumplir con diversas actividades mientras se les controle apropiadamente.

Los progenitores más directos de los robots, fueron los telemanipuladores. En 1948, R. C. Goertz desarrolló en el Argonne National Laboratory el primer telemanipulador, con el objetivo de manejar elementos radiactivos sin riesgo para el operador.

En año de 1942, en Bendix Corporation surgió lo que se podría llamar un control numérico verdadero, debido a una necesidad impuesta por la industria aeronáutica para la realización de hélices de helicópteros de diferentes configuraciones. Para el año de 1952 John C. Parsons y el Instituto Tecnológico de Massachusetts desarrollaron la primera máquina fresadora de control numérico, construida con bulbos, usando un código binario y cinta perforadora.

La sustitución del operador por un programa de ordenador que controlase los movimientos del manipulador, dio paso al concepto de robot. La primera patente de un dispositivo robótico fue solicitada en 1954 por el británico C.E. Kenward. Sin embargo, fue George C. Devol, un ingeniero norteamericano, el que estableció las bases del robot industrial moderno.

A partir de 1976, se aplicó el microprocesador a las computadoras, se dio un enorme salto en el desarrollo del CNC, haciéndose posible las interpolaciones rectas y curvas entre ejes.

El año 1980 fue llamado primer año de la era robótica porque la producción de robots industriales aumentó ese año un 80 % respecto al año anterior.

Con respecto a las impresoras 3D, en 1984 Charles Hull, el co-fundador de 3D Systems, inventa el método de la estereolitografía, proceso de impresión, orientado a maquetas para la prueba de prototipos antes de su fabricación en cadena. Seis años después, S. Scott Crump, fundador de Stratasys, desarrolla la técnica de Fused Deposition Modeling (FDM), que consistía en la creación de objetos tridimensionales mediante la superposición de capas de material fundido que posteriormente solidificaba con la forma deseada. Con el tiempo este método permitió una mayor difusión de la impresión 3D.

Para el año de 1992, aparece la primera máquina de impresión 3D del tipo SLA (estereolitográfico) en el mercado, fue desarrollada por la empresa 3D Systems. El funcionamiento básico de esta máquina consiste en que un láser UV va solidificando un fotopolímero, un líquido con la viscosidad y color parecido al de la miel, el cual va fabricando partes tridimensionales capa por capa.

Durante 1993, un grupo de estudiantes del MIT concibe la impresión 3D por inyección. Y un año después, fundaron su propia empresa, Z Corporation, y dos años más tarde, en 1995, inician la venta de los primeros equipos basados en esta tecnología.

Sin embargo, la impresión 3D, también fue ganando terreno en el sector salud, a finales de los años noventa, en el Instituto de Wake Forest de Medicina Regenerativa, desarrolla el primer órgano criado en laboratorio que se implementó en humanos fue un aumento de la vejiga urinaria utilizando recubrimiento sintético con sus propias células. Para el 2002, los científicos diseñan un riñón en miniatura completamente funcional y con la capacidad de filtrar sangre y producir orina diluida en un animal. El desarrollo llevó a la investigación en el Instituto de Wake Forest de Medicina Regenerativa el objetivo de imprimir los órganos y tejidos con tecnología de impresión 3D.

El 2005 es un año que cambia el rumbo de las impresoras 3D, debido a que el Dr. Adrián Bowyer funda RepRap, en la Universidad de Bath, en Inglaterra, una iniciativa de código abierto para construir una impresora 3D que puede imprimir la mayoría de sus propios componentes. Este hecho constituye la entrada del Open-source en la historia de la impresión tridimensional y un gran paso hacia su normalización en el mercado.

Continuando con el estilo de estereolitografía, en el 2006 se construye la primera máquina del tipo SLS (Sinterización de laser selectivo) viable. Básicamente, este tipo de máquina utiliza un láser para fundir materiales en el proceso de impresión 3D. Este descubrimiento abre las puertas a la personalización masiva y a la demanda de fabricación de piezas industriales, y más tarde, prótesis.

Ese mismo año, Object, un proveedor de materiales e impresoras 3D, crea una máquina con la capacidad de imprimir en múltiples materiales, incluyendo polímeros y elastómeros. La máquina permite que una parte sea fabricada con una gran variedad de densidades y propiedades de material.

En el 2008, tras su lanzamiento en 2005, el proyecto RepRap saca a la luz Darwin, la primera impresora 3D con capacidad de imprimir la mayoría de sus propios componentes, permitiendo a los usuarios que ya tienen una, hacer más impresoras para sus amigos o incluso reparar componentes de la suya.

En este mismo año, se documenta, la primera persona que camina sobre una pierna de prótesis impresa en 3D, con todas las partes, rodilla, pie, etc., impresa en una misma compleja estructura sin ningún tipo de montaje. Este tipo de avances permiten que los fabricantes de prótesis realicen desarrollos a medida en el sector de las prótesis.

Un año después, la empresa Organovo ingenia la impresora 3D MMX Bioprinter, la primera capaz de fabricar tejidos orgánicos con la tecnología del Dr. Gabor Forgacs, que es utilizada para imprimir el primer vaso sanguíneo.

Durante el 2009, Industrias MakerBot, una compañía de hardware de código abierto para las impresoras 3D, comienza la venta de kits de montaje que permiten a los compradores fabricar sus propias impresoras 3D y productos.

Después del 2010, los ingenieros de la Universidad de Southampton diseñaron y planearon el primer avión impreso en 3D. Este avión no tripulado se construye en siete días, con un presupuesto de 7.000€. La impresión 3D permite que sus alas tengan forma elíptica, una característica normalmente cara que ayuda a mejorar la eficiencia aerodinámica y reduce al mínimo la resistencia inducida. Kor Ecologic nos presenta Urbee, un prototipo de coche que trata de ser lo más eficiente posible con el medio ambiente, siendo toda su carrocería diseñada e impresa en 3D. Trata de ser un coche eficiente en cuanto a consumo de gasolina y en cuanto a su coste de producción. La empresa materialise ha sido la primera empresa en ofrecer un servicio de impresión 3D de oro de 14 Kilates y plata de ley. Esta opción va a permitir abrir un nuevo mercado a los joyeros con diseños más económicos utilizando este material.

En los últimos años, Doctores e ingenieros holandeses trabajan con una impresora 3D especialmente diseñada por la empresa LayerWise, la cual permite imprimir prótesis de mandíbulas personalizadas. Este grupo ha podido implantar una mandíbula a una mujer de 83 años de edad que sufría una infección de hueso crónica. Esta tecnología se está estudiando más profundamente con el objetivo de poder promover el crecimiento de nuevo tejido óseo.

Impresión 3D

Una impresora 3D es un dispositivo capaz de generar un objeto sólido tridimensional a partir de un modelo digital, mediante la adición de material, es decir, a través de la superposición de capas de material, en vez de la remoción del material como sucede en los procesos tradicionales.

Hoy en día, son comúnmente utilizadas para la creación de prototipos, matricería o prefabricación de piezas principalmente en el sector industrial y en arquitectura. Sin embargo, en el sector médico se ocupan para la creación de prótesis médicas, pues permiten adaptarlas a las características particulares de cada paciente con facilidad.

Tipos de impresoras 3D

- Estereolitografía

Fue el primer método concebido. Consiste en la aplicación de un láser ultravioleta a una resina sensible a la luz contenida en un cubo. El láser va solidificando la resina en capas, hasta que el objeto adquiere la forma deseada.

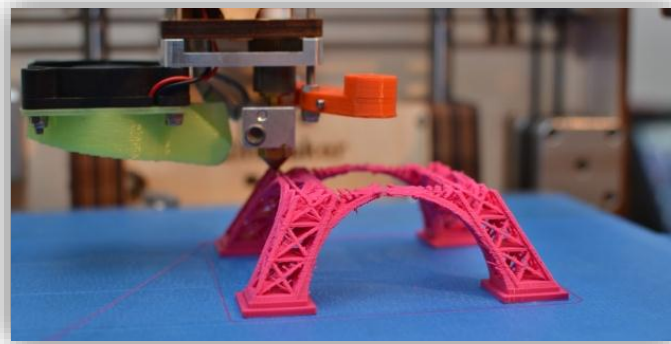
- Impresión por láser (selective laser sintering o SLS).

En este proceso la materia prima está en forma de polvo, en un contenedor, y se utiliza un láser para fundir y solidificar la pieza.

Algunas ventajas son, que no se necesitan columnas de soporte, la calidad es muy alta, se pueden hacer piezas de diferentes materiales (metal, plástico, etc.), pero su principal desventaja es el costo de la máquina y de la materia prima, puesto que son muy elevados para este proceso.

- Extrusión (Deposición de Material Fundido)

Este tipo de impresión, está basado en la extrusión en caliente de un filamento de plástico. El proceso consiste en empujar un hilo de plástico a través de un dispositivo, denominado extrusor, que se calienta hasta una temperatura capaz de fundir ligeramente el material de plástico utilizado, (actualmente se ocupa ABS o PLA), sin llegar a derretirlo por completo, al llegar a la punta del dispositivo, que es mucho más fina que el filamento original, se va expulsando un hilo muy fino y delgado de plástico que se va depositando en una base caliente, para que el hilo extruido se vaya quedando pegado, y así la máquina pueda ir creando la pieza capa a capa.



Impresión 3D mediante extrusión de material

Impresoras de extrusión 3D en el mercado.

Algunas de las marcas y modelos más reconocidos a nivel internacional de impresoras 3D se mencionan a continuación.

Cube 3D printer

Esta impresora se comercializa atacando el mercado doméstico, es decir está dirigido principalmente para que cada persona tenga una impresora 3D en su hogar.

Algunas de las características fundamentales de esta impresora son:

- ⊕ Comodidad en la palma de tu mano: Utiliza una aplicación llamada Cubify para iOS y Android, la cual permite preparar e imprimir desde su teléfono inteligente dirigiendo a Cube. Además, cuenta con colecciones de diseños propios de la marca.
- ⊕ Utilizando un color imprime dos veces más rápido que otras impresoras.
- ⊕ Cuenta con cartuchos de nueva tecnología no-obstrucción y control de humedad, lo que asegura una vida más larga y el uso total del material, mejora de la calidad de impresión y la sostenibilidad. Por otro lado, detecta automáticamente el tipo de material dependiendo del cartucho insertado.
- ⊕ Posee un certificado de seguridad bajo la norma IEC 60953, haciéndola fácil de usar para niños mayores de 8 años.
- ⊕ Sus dimensiones con cartucho son: 33.5 (W) x 34.3 (H) x 24.1 (d) cm
- ⊕ Su área de trabajo es: 15,25 x 15,25 x 15,25 cm

- ⊕ Resolución: 70 micras, en modo rápido 200 micras.



Impresora 3D CUBE.

Makerbot Replicator Impresora de escritorio 3D

Esta impresora es de la quinta generación de esta empresa. Siendo el más nuevo estándar en impresoras 3D para escritorio.

Pequeña descripción del equipo:

- ⊕ Cuenta con conectividad de wi-fi, USB y Ethernet.
- ⊕ Está equipada con una cámara, que toma imágenes de la impresión durante el proceso lo que permite monitorear la pieza en desarrollo y el equipo.
- ⊕ Tiene una aplicación para celular inteligente y una nube en la que se guardan de manera automática las imágenes capturadas de la cámara.

La resolución de la capa es de 100 micras.

Dimensiones: 25.2L x 19.9W x H15.0 cm.

Una de las características más importantes es que no necesita postproducción, ya que las capas de impresión son delgadas como una hoja de papel.



Impresora 3D MakerBot Replicator

Impresora RepRap I3 Prusa

La característica más importante de esta marca, es que cuenta con arquitectura abierta, es decir puedes construir tu impresora 3D a la medida de tus necesidades, en base a los kits

comerciales y en caso de que no te sirvan, te orientan para imprimir tus propias piezas y armar tu impresora, volviéndola una de las más populares entre los usuarios.

I3 Prusa (iteración 3) es el diseño más nuevo y más actual de la impresora 3D RepRap por Core Developer Prusa. El i3 incorpora las lecciones aprendidas de los dos diseños anteriores Prusa, así como otros diseños RepRap modernas populares.

Este kit, es para un área de trabajo de 20cm x 20cm x 20cm. Cuenta con 5 motores a pasos NEMA 17.



Impresora I3 Prusa.

Una de las actividades más importantes para un país, es el desarrollo de tecnología ya que influye directamente en todas las áreas importantes, como la economía, el medio ambiente pero principalmente en la sociedad. Los países desarrollados que cuentan con alta tecnología en sus industrias, les permite aumentar sus índices de producción y calidad a costos reducidos, favoreciéndolos en la competencia comercial tanto nacional como internacional, generando una cinergia económica-social.

La falta de desarrollo tecnológico en nuestro país, es una de las principales causas de que muchas de las empresas mexicanas no puedan competir con empresas extranjeras. Por lo anterior es muy importante por la naturaleza de nuestra maestría, ya que al desarrollar proyectos de alta tecnología vamos adquiriendo experiencia en el desarrollo de maquinaria que un futuro puedan utilizarse en sistemas automatizados de producción, apoyando al crecimiento y fortalecimiento de este sector en nuestro país.

Un robot de coordenadas cartesianas (también llamado robot cartesiano) es un robot cuyos tres ejes principales de control son lineales (se mueven en línea recta en lugar de rotar) y forman ángulos rectos unos respecto de los otros. Además de otras características, esta configuración mecánica simplifica las ecuaciones en el control de los brazos robóticos.

Los robots cartesianos constan de una serie de componentes generales que se repiten en todas las concepciones que existen de este tipo de robot. Del acoplamiento de servomotores y controladores de posicionamiento con sistemas de ejes surgen robots cartesianos de gran precisión y rendimiento para la realización de tareas de posicionamiento en uno o varios ejes.

Es por esto que se decidió, desarrollar este robot cartesiano del tipo CNC, para poder optimizar nuestros procesos al montaje de los FPGAs, en las placas desarrolladas por la facultad de ciencias de la electrónica, sin embargo, este equipo es de gran versatilidad, lo cual nos permite utilizarlo para diversas tareas. Al implementar una herramienta extra en este sistema podemos maquilar piezas de aluminio, madera, nylon entre otros materiales.

En esta tesis abordaremos una de las aplicaciones que se le puede dar a los robots cartesianos (con 3 grados de libertad), el cual utilizaremos para la implementación de una impresora 3D. Las impresoras 3D son dispositivos que han ido ganando mucha fama a nivel industrial y de diseño, debido a las ventajas que presenta para el diseño de prototipos de cualquier índole, teniendo un bajo costo de inversión y poco tiempo de fabricación, sin embargo, también han sido utilizadas para el área de la salud, ya que se ha conseguido “imprimir” algunos tipos de órganos y prótesis, que se adecuan a cada persona ya que son “diseños especiales” para ellos.

Objetivo General:

“DISEÑAR Y CONSTRUIR UN ROBOT CARTESIANO PARA IMPRIMIR EN 3D”

Objetivos Específicos:

1. Estudiar la cinemática del robot.
2. Estudiar la dinámica del robot.
3. Implementar la herramienta para imprimir en 3D.
4. Diseñar e instrumentar la electrónica del sistema de control del robot.
5. Desarrollar un controlador de trayectoria para comparar su desempeño con el controlador de posición actual.
6. Desarrollar un programa laminador.
7. Implementar un compilador.
8. Desarrollar el software para imprimir en 3D.
9. Realizar las pruebas de la impresora 3D.
10. Publicación de resultados.

Con respecto al contenido de esta tesis, en el capítulo 1, abordaremos todo lo relacionado con las características propias del CNC, además se hará un análisis del modelo dinámico del sistema, empezando desde la cinemática directa hasta las ecuaciones de Euler-Lagrange, además se explicará de manera general la mecánica del sistema.

En el capítulo 2, se mencionará una descripción del firmware que se emplea para el control de posición de los motores de cada eje, utilizando el puerto PCI como interfaz.

En el capítulo 3, tendremos una descripción breve del software inicial con el cual se empezó a trabajar para la implementación de la impresora 3D.

Por otro lado, en los capítulos 4 y 5, se hace alusión a la herramienta instrumentada para la realización de la impresión 3D, el firmware y software utilizado para el control de la misma.

Finalmente, en el capítulo 6, se plantearán los resultados experimentales obtenidos y las conclusiones generales de la tesis, siendo esto la culminación del trabajo realizado.

Capítulo 1. Descripción del robot CNC

Actualmente, se cuenta con un robot CNC, el cual tiene grandes prestaciones por lo que se pensó implementar como impresora 3D, para satisfacer la necesidad de realizar prototipos rápidos, tanto para la investigación como para el desarrollo de nuevos proyectos.

1.1. Características generales

Características de la impresora 3D	
Descripción	Requerimiento
Número de grados de libertad	3 (ejes X, Y y Z)
Desplazamiento en eje X y Y	<ul style="list-style-type: none">■ Tornillo embalado de 12mm■ Guía de alta precisión de 32mm■ Bloque para guía de 32mm
Eje X	$\Delta x = .500\mu\text{m}$
Eje Y	$\Delta y = .500\mu\text{m}$
Desplazamiento en eje Z	<ul style="list-style-type: none">■ Tornillo embalado de 12mm■ Guía de alta precisión de 15mm■ Bloque para guía de 15mm
Eje Z	$\Delta z = .500\mu\text{m}$
Temperatura del extrusor	190°C a 210°C
Temperatura de la base caliente	70°C a 80°C
Velocidad de alimentación de material	De 2 a 4cm por segundo
Área de trabajo	210mm x 210mm
Alimentación	120VCA y 12VCD
Velocidad de impresión	Variable. Máx. de 5cm/s

Tabla 1: Requerimientos para la impresión 3D

Para poder utilizar el robot CNC como una impresora 3D se quería una resolución mínima de $.500\mu\text{m}$ por eje, para tener una impresión con buena calidad. Por otro lado, se necesita de una herramienta de impresión el extrusor y la base caliente necesitamos realizar un control de temperatura para poder mantener una temperatura de trabajo de 190°C a 210°C, y de 70°C a 80°C respectivamente. Finalmente, necesitamos un regulador de velocidad para modular la alimentación del material a nuestro extrusor y tener una impresión apropiada.

Todo lo anterior debe estar controlado por el usuario, por lo que se propuso utilizar un FPGA para realizar este control, utilizando el puerto PCI de una computadora como interfaz, además se requiere de un programa para modificar y visualizar los parámetros del sistema, en este caso utilizamos LabView para hacer el software de control.

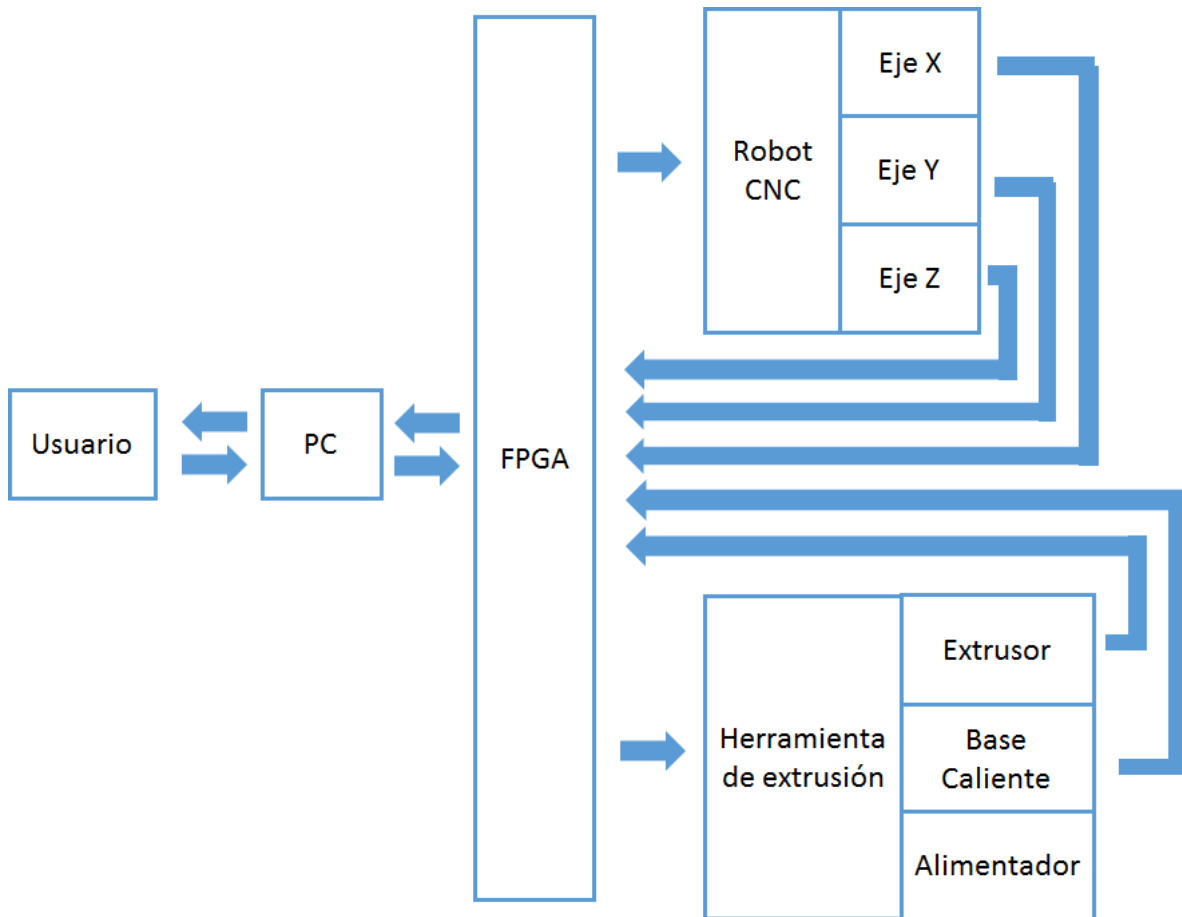


Figura 1.1: Descripción general del funcionamiento de la impresora 3D.

En el diagrama anterior se plantea la idea general de funcionamiento de nuestra impresora 3D. Se puede observar que tienen cinco lazos de control principales, tres para el control de movimiento y dos para el control de la herramienta de extrusión. Además, tendremos un lazo abierto de control, ya que se maneja un motor a pasos el cual no tiene retroalimentación.

Con lo anterior se comprueba que se satisface la necesidad mínima de precisión del sistema, por lo que es viable implementar la impresión 3D.

1.2. Modelo dinámico del robot cartesiano.

Para poder realizar un control apropiado para nuestro robot, es necesario tener en cuenta el modelo dinámico del mismo, esto nos servirá para analizar su comportamiento, al utilizar diferentes algoritmos de control y de esta manera poder utilizar el presente un mejor desempeño.

Para determinar el modelo dinámico, utilizaremos la metodología de Euler-Lagrange, la cual está ampliamente documentada.

A continuación, se muestra la representación geométrica del robot cartesiano de 3 grados de libertad, la cual nos ayuda a visualizar el comportamiento del manipulador.

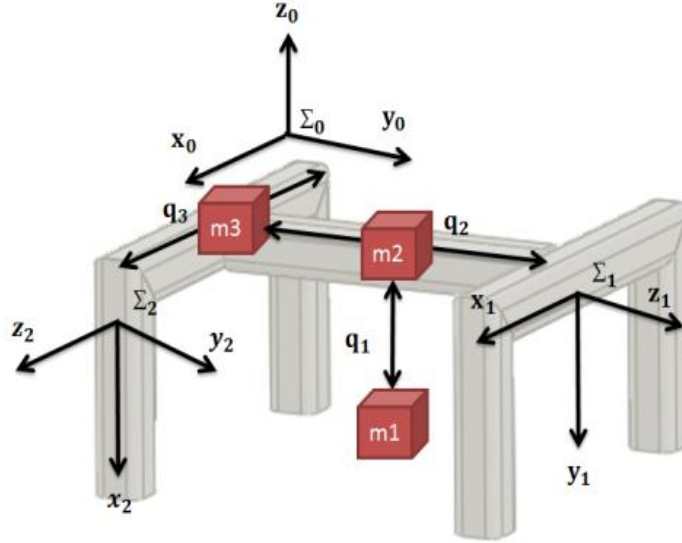


Figura 1.2: Representación del robot cartesiano de 3 grados de libertad.

Sabemos que, la energía total $E(\mathbf{q}, \dot{\mathbf{q}})$ (Hamiltoniano) de un robot manipulador de n grados de libertad es la suma de sus energías, cinética $K(\mathbf{q}, \dot{\mathbf{q}})$ y potencial $U(\mathbf{q})$. [7]

$$E(\mathbf{q}, \dot{\mathbf{q}}) = K(\mathbf{q}, \dot{\mathbf{q}}) + U(\mathbf{q}), \quad (1)$$

Donde $\mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^n$ representan los vectores de posición, en coordenadas generalizadas y velocidad articular.

Además, el lagrangiano $L(\mathbf{q}, \dot{\mathbf{q}})$ de un robot manipulador de n grados de libertad se define como la diferencia entre su energía cinética $K(\mathbf{q}, \dot{\mathbf{q}})$ y su energía potencial $U(\mathbf{q})$:

$$L(\mathbf{q}, \dot{\mathbf{q}}) = K(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q}), \quad (2)$$

Por otra parte, las ecuaciones de movimiento de Euler-Lagrange para un robot manipulador de n grados de libertad, están dadas por:

$$\frac{d}{dt} \left[\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right] - \frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} = \boldsymbol{\tau} - \mathbf{v}(\dot{\mathbf{q}}, \mathbf{f}_e), \quad (3)$$

Donde $\boldsymbol{\tau} \in \mathbb{R}^n$ es el vector de pares aplicados y $\boldsymbol{v}(\dot{\mathbf{q}}, \mathbf{f}_e) \in \mathbb{R}^n$ es el vector de fuerzas o pares de fricción que depende de la velocidad articular $\dot{\mathbf{q}}$ y de la fricción estática \mathbf{f}_e , que se encuentra presente en las articulaciones del mismo.

De forma equivalente:

$$\tau_i = \frac{d}{dt} \left[\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_i} \right] - \frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_i} + b_i \dot{q}_i + f_{ci} \text{signo}(\dot{q}_i) + f_{ei} [1 - |\text{signo}(\dot{q}_i)|] \quad i = 1, \dots, n, \quad (4)$$

Donde b, f_c son coeficientes de fricción viscosa y de Coulomb, respectivamente, f_e representa los coeficientes de fricción estática de cada una de los servomotores del robot manipulador. [8]

De esto, podemos decir que dependiendo del número grados de libertad, será el número de ecuaciones dinámicas del robot manipulador.

Para determinar el modelo de cinemática directa utilizaremos la metodología de Denavit-Hartenberg, la cual ofrece un procedimiento sencillo y cuya estructura queda en representación de transformaciones homogéneas.

La representación de Denavit-Hartenberg, se puede expresar por el producto de cuatro transformaciones básicas:

$$H_{i-1}^i = H_{R_{z_{i-1}}}(\theta_i) H_{T_{z_{i-1}}}(d_i; \beta_i) H_{T_{x_{i-1}}}(l_i) H_{R_{x_{i-1}}}(\alpha_i), \quad (5)$$

La transformación homogénea total se obtiene como $H_0^n = H_0^1 H_1^2 \dots H_{n-1}^n$.

Tomando la figura 1.2.1 como referencia para obtener la tabla de parámetros de Denavit-Hartenberg, necesitamos realizar una rotación auxiliar debido a que los ejes del sistema son ortogonales, a continuación, se presenta proceso realizado para abordar esta problemática.

Paso 1, se realiza una rotación de $-\frac{\pi}{2}$ sobre el eje x_0 :

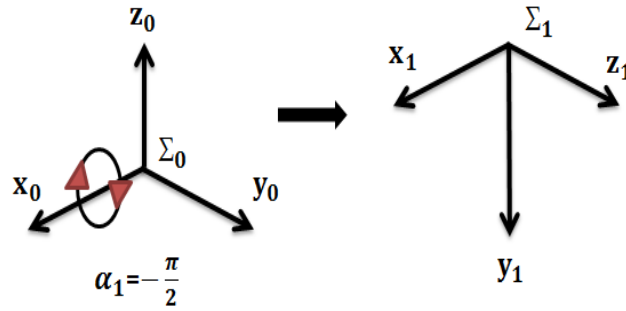


Figura 1.3: Representación de la primera rotación de Denavit-Hartenberg, para obtener el sistema de referencia Σ_1 .

Paso 2, como no hay una rotación posible para que el eje z_1 se alineé al eje z_2 del sistema de referencia Σ_2 , se hace una rotación previa sobre el eje z_1 , lo que nos lleva:

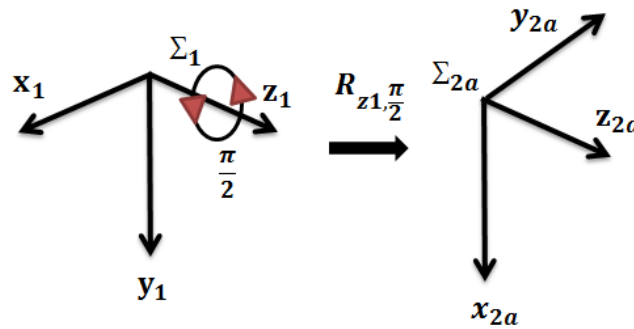


Figura 1.4: Representación del sistema Σ_{2a} , después de la rotación del eje z_1 , el cual nos sirve como un sistema auxiliar para poder alcanzar la alineación deseada en el sistema Σ_2 .

Paso 3, finalmente se realiza una rotación sobre el eje x_{2a} , alcanzando la alineación del eje z_1 , con respecto al eje z_2 .

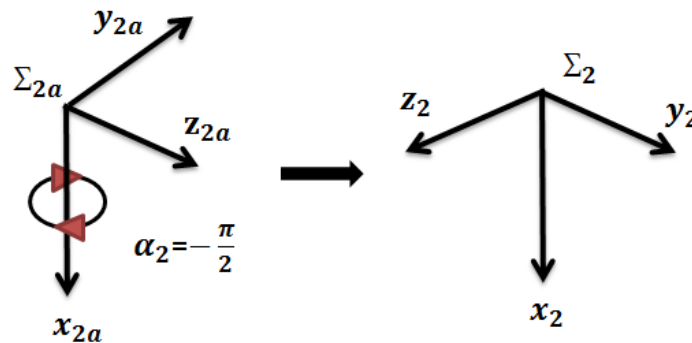


Figura 1.5: Representación del sistema referenciado Σ_2 , después de la última rotación realizada sobre el eje x_{2a} , del sistema Σ_{2a} .

Con la información anterior podemos adquirir los parámetros necesarios para llenar la tabla de Denavit-Hartenberg, que nos queda de la siguiente manera:

Eslabón	l_i	α_i	d_i	θ_i
1	0	$-\pi/2$	d_1	0
2	0	$-\pi/2$	d_2	0
3	0	0	d_3	0

Tabla 2: Tabla de parámetros de Denavit-Hartenberg.

Sustituyendo los parámetros obtenidos, en la ecuación (5), y después de realizar las operaciones algebraicas necesarias, tendremos, que para el primer eslabón:

$$H_0^1 = H_{R_{z_0}}(0)H_{T_{z_0}}(d_1)H_{T_{x_0}}(0)H_{R_{x_0}}\left(-\frac{\pi}{2}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (6)$$

La cinemática directa será:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix}, \quad (7)$$

Para el segundo eslabón, necesitamos determinar primero la transformación homogénea del segundo eslabón al primer eslabón, de lo que obtendremos:

$$H_1^2 = H_{R_{z_1}}(0)H_{T_{z_1}}(d_2)H_{T_{x_1}}(0)H_{R_{x_1}}\left(-\frac{\pi}{2}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

La transformación homogénea total será:

$$H_0^2 = H_0^1 H_1^2, \quad (9)$$

Esto nos lleva:

$$H_0^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (10)$$

De lo anterior, tenemos que la cinemática directa para el segundo eslabón será:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ d_2 \\ d_1 \end{bmatrix}, \quad (11)$$

Finalmente, para el tercer eslabón tendremos que utilizar la transformación homogénea auxiliar anteriormente explicada, lo que nos lleva:

$$H_{2a} = H_{R_{z_1}} \left(\frac{\pi}{2} \right)^T H_1^2 = H_{R_{z_1}} \left(\frac{\pi}{2} \right)^T H_{R_{z_1}}(0) H_{T_{z_1}}(d_2) H_{T_{x_1}}(0) H_{R_{x_1}} \left(-\frac{\pi}{2} \right)$$

$$H_{2a} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (12)$$

Además, obtendremos la transformación homogénea del tercer eslabón al segundo, de la siguiente manera:

$$H_2^3 = H_{R_{z_2}}(0) H_{T_{z_2}}(d_3) H_{T_{x_2}}(0) H_{R_{x_2}}(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (13)$$

Por lo que la transformación homogénea resultante será:

$$H_0^3 = H_0^1 R_{z, \frac{\pi}{2}}^T H_1^2 H_2^3 = \begin{bmatrix} 0 & 0 & 1 & d_3 \\ 0 & -1 & 0 & d_2 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (14)$$

De la ecuación 14, obtendremos que la cinemática directa queda de la siguiente manera:

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} d_3 \\ d_2 \\ d_1 \end{bmatrix}, \quad (15)$$

Tomando en cuenta que las coordenadas $[d_3 \ d_2 \ d_1]^T$, son coordenadas cartesianas, las expresaremos en coordenadas generalizadas como $[q_3 \ q_2 \ q_1]^T$, para fines prácticos.

Del modelo de cinemática directa podemos obtener la cinemática diferencial, que es la derivada con respecto al tiempo de la cinemática directa, es decir:

$$\mathbf{v}_i = \frac{d}{dt} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \frac{\partial f_R(l_i, \mathbf{q}_i)}{\partial \mathbf{q}_i} = J(\mathbf{q}) \dot{\mathbf{q}} \quad i = 1, \dots, n, \quad (16)$$

dónde:

\mathbf{v}_i . Simboliza la velocidad lineal

$\partial f_R(l_i, \mathbf{q}_i)$. Función vectorial que relaciona las coordenadas articulares \mathbf{q}_i y las propiedades geométricas del sistema mecánico l_i , con las coordenadas cartesianas.

$J(\mathbf{q})$. Representa el jacobiano del robot o jacobiano analítico.

Para nuestro robot manipulador esta ecuación queda de la siguiente manera, para cada eslabón:

$$\mathbf{v}_1 = \frac{d}{dt} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 \end{bmatrix}, \quad (17)$$

$$\mathbf{v}_2 = \frac{d}{dt} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \dot{q}_2 \\ \dot{q}_1 \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{q}_2 \\ \dot{q}_1 \end{bmatrix}, \quad (18)$$

$$\mathbf{v}_3 = \frac{d}{dt} \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_3 \\ \dot{q}_2 \\ \dot{q}_1 \end{bmatrix} = \begin{bmatrix} \dot{q}_3 \\ \dot{q}_2 \\ \dot{q}_1 \end{bmatrix}, \quad (19)$$

Para resolver la ecuación (2), es necesario determinar la energía cinética, que incluye el movimiento de traslación y rotación, está definida como:

$$K(\mathbf{q}, \dot{\mathbf{q}}) = \frac{m_i \mathbf{v}_i^T \mathbf{v}_i}{2} + \frac{I_i [\sum_i^n \dot{q}_i]^2}{2} = \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} \quad i = 1, \dots, n, \quad (20)$$

dónde:

m_i . Simboliza el vector de las masas del sistema.

$\mathbf{v}_i^T \mathbf{v}_i$. Representa la rapidez lineal del centro de masa de cada eslabón.

$$\mathbf{v}^2 = \|\mathbf{v}\|^2 = \mathbf{v}^T \mathbf{v}, \quad (21)$$

Para nuestro caso, \mathbf{v}^2 queda representado como:

$$\mathbf{v}_1^2 = [0 \quad 0 \quad \dot{q}_1] \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 \end{bmatrix} = \dot{q}_1^2, \quad (22)$$

$$\mathbf{v}_2^2 = [0 \quad \dot{q}_2 \quad \dot{q}_1] \begin{bmatrix} 0 \\ \dot{q}_2 \\ \dot{q}_1 \end{bmatrix} = \dot{q}_2^2 + \dot{q}_1^2, \quad (23)$$

$$\mathbf{v}_3^2 = [\dot{q}_3 \quad \dot{q}_2 \quad \dot{q}_1] \begin{bmatrix} \dot{q}_3 \\ \dot{q}_2 \\ \dot{q}_1 \end{bmatrix} = \dot{q}_3^2 + \dot{q}_2^2 + \dot{q}_1^2, \quad (24)$$

I_i . Es el momento de inercia del i-ésimo eslabón.

$M(\mathbf{q})$. Es la matriz de inercia del manipulador, sin embargo, para nuestro caso está matriz es una matriz de masas constantes, ya que no tiene efecto inercial, la cual denominaremos como M.

Por otro lado, en nuestro robot cartesiano solo contamos con movimiento de traslación sobre los ejes x, y, z, de esto, podemos decir que la energía de rotación para estos ejes es igual a cero, por lo que nuestra ecuación de energía cinética quedará de la siguiente manera:

$$K(\mathbf{q}, \dot{\mathbf{q}}) = \frac{m_i \mathbf{v}_i^T \mathbf{v}_i}{2} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} \quad i = 1, \dots, n, \quad (25)$$

Sustituyendo los valores de las ecuaciones 22, 23 y 24, en la ecuación 25 tendremos:

$$\begin{aligned} K(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{1}{2} [m_1 \dot{q}_1^2 + m_2 (\dot{q}_2^2 + \dot{q}_1^2) + m_3 (\dot{q}_3^2 + \dot{q}_2^2 + \dot{q}_1^2)] \\ K(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{1}{2} [\dot{q}_1^2 (m_1 + m_2 + m_3) + \dot{q}_2^2 (m_2 + m_3) + \dot{q}_3^2 m_3], \end{aligned} \quad (26)$$

Una vez determinada la energía cinética, podemos encontrar la matriz de masas, la cual tomará la siguiente forma:

$$\mathbf{M} = \begin{bmatrix} m_1 + m_2 + m_3 & 0 & 0 \\ 0 & m_2 + m_3 & 0 \\ 0 & 0 & m_3 \end{bmatrix}, \quad (27)$$

Como sabemos, la energía potencial $U(\mathbf{q})$ no tiene una forma específica como en el caso de la energía cinética, depende de la geometría del robot en general, sin embargo, tiene una

dependencia exclusivamente del vector de posición (\mathbf{q}), ya que se considera su presencia a campos conservativos como la fuerza de gravedad, por lo tanto, para nuestro robot manipulador, la energía potencial solo afectará el primer eslabón, puesto que para el segundo y tercer eslabón se mantiene constante. La energía potencial se encuentra representada de la siguiente manera:

$$U(\mathbf{q}) = m_1 g q_1, \quad (28)$$

donde, g es la constante gravitacional.

Por parte, sabemos que $\mathbf{g}(\mathbf{q})$ es el vector de pares gravitacionales obtenido como el gradiente de la energía potencial, es decir:

$$\mathbf{g}(\mathbf{q}) = \frac{\partial U(\mathbf{q})}{\partial \mathbf{q}}, \quad (29)$$

Por lo tanto, vector de pares gravitacionales de nuestro robot queda de la siguiente manera:

$$\mathbf{g}(\mathbf{q}) = g \begin{bmatrix} m_1 \\ 0 \\ 0 \end{bmatrix}, \quad (30)$$

Retomando la ecuación 2, y sustituyendo los valores de la energía cinética y la energía potencial obtendremos:

$$L(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} [\dot{q}_1^2 (m_1 + m_2 + m_3) + \dot{q}_2^2 (m_2 + m_3) + \dot{q}_3^2 m_3] - m_1 g q_1, \quad (31)$$

Como ya hemos obtenido el lagrangiano, podemos encontrar los valores de la ecuación 3, lo que nos lleva:

$$\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_1} = (m_1 + m_2 + m_3) \dot{q}_1, \quad (32)$$

$$\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_2} = (m_2 + m_3) \dot{q}_2, \quad (33)$$

$$\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_3} = m_3 \dot{q}_3, \quad (34)$$

$$\frac{d}{dt} \left[\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_1} \right] = (m_1 + m_2 + m_3) \ddot{q}_1, \quad (35)$$

$$\frac{d}{dt} \left[\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_2} \right] = (m_2 + m_3) \ddot{q}_2, \quad (36)$$

$$\frac{d}{dt} \left[\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_3} \right] = m_3 \ddot{q}_3, \quad (37)$$

$$\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_1} = m_1 g, \quad (38)$$

$$\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_2} = 0, \quad (39)$$

$$\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_3} = 0, \quad (40)$$

Sabemos que el modelo dinámico cartesiano se encuentra dado por:

$$\mathbf{f}_\chi = M_\chi \ddot{\chi} + C_\chi \dot{\chi} + \mathbf{g}_\chi + \mathbf{f}_{e_\chi}, \quad (41)$$

donde, $\chi = [x \ y \ z]^T$, son coordenadas cartesianas.

\mathbf{f}_χ representa la dinámica no lineal de un robot manipulador en el espacio de variables cartesianas.

M_χ simboliza la matriz de inercia cartesiana.

C_χ representa la matriz de fuerzas centrípetas y de Coriolis cartesiana.

\mathbf{g}_χ es el vector de pares gravitacionales cartesiano.

\mathbf{f}_{e_χ} simboliza el fenómeno de fricción cartesiano.

Conocemos que la cinemática diferencial y la aceleración articular se encuentran relacionadas por:

$$\dot{\chi} = J(\mathbf{q})\dot{\mathbf{q}} \Rightarrow \dot{\mathbf{q}} = J^{-1}(\mathbf{q})\dot{\chi}, \quad (42)$$

$$\ddot{\mathbf{q}} = J^{-1}(\mathbf{q})\ddot{\chi} + \left[\frac{d}{dt} J^{-1}(\mathbf{q}) \right] \dot{\chi}, \quad (43)$$

$$\ddot{\chi} = J^{-1}(\mathbf{q})\ddot{\mathbf{q}} - J^{-1}(\mathbf{q})\dot{J}(\mathbf{q})(\mathbf{q})J^{-1}(\mathbf{q})\dot{\chi}, \quad (44)$$

En nuestro caso no se cuenta con la matriz de fuerzas centrípetas y de Coriolis, ya que es igual a cero, con respecto a la matriz de inercia cartesiana nosotros contamos con una matriz de masas, mencionada con anterioridad.

Por lo tanto, el modelo dinámico cartesiano está dado por:

$$\mathbf{f}_\chi = \begin{bmatrix} m_1 + m_2 + m_3 & 0 & 0 \\ 0 & m_2 + m_3 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \mathbf{g} \begin{bmatrix} m_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} f_{c1} & 0 & 0 \\ 0 & f_{c2} & 0 \\ 0 & 0 & f_{c3} \end{bmatrix} \begin{bmatrix} \text{signo}(\dot{q}_1) \\ \text{signo}(\dot{q}_2) \\ \text{signo}(\dot{q}_3) \end{bmatrix} + \begin{bmatrix} f_{e1}[1 - |\text{signo}(\dot{q}_1)|] \\ f_{e2}[1 - |\text{signo}(\dot{q}_2)|] \\ f_{e3}[1 - |\text{signo}(\dot{q}_3)|] \end{bmatrix}, \quad (45)$$

Este modelo puede ser igualado a $\boldsymbol{\tau}$ gracias a la siguiente relación:

$$\boldsymbol{\tau} = J^T(\mathbf{q})\mathbf{f}_\chi, \quad (46)$$

Sin embargo, para realizar un acoplamiento de unidades entre τ (Nm) y f_x (N) se debe agregar una constante α la cual tendrá como valor 1 metro, por lo tanto, la ecuación 46 quedará de la siguiente manera:

$$\boldsymbol{\tau} = \alpha \mathbf{J}^T(\mathbf{q}) \mathbf{f}_x = \alpha \mathbf{f}_x \quad , \quad (47)$$

Obteniendo el resultado de pares aplicados de la metodología de Euler-Lagrange. Siendo estos los pares que se apliquen en cada eslabón de nuestro robot.

1.3. Mecánica del robot

Con respecto a la parte mecánica de nuestro robot cartesiano, tenemos un motorreductor de corriente directa de 12V con un encoder incremental de cuadratura acoplado en la parte posterior, para cada uno de los ejes; además tendremos rieles de alta precisión para garantizar el desplazamiento lineal y tornillos embalados para convertir el movimiento giratorio del motor en un movimiento de traslación lineal.

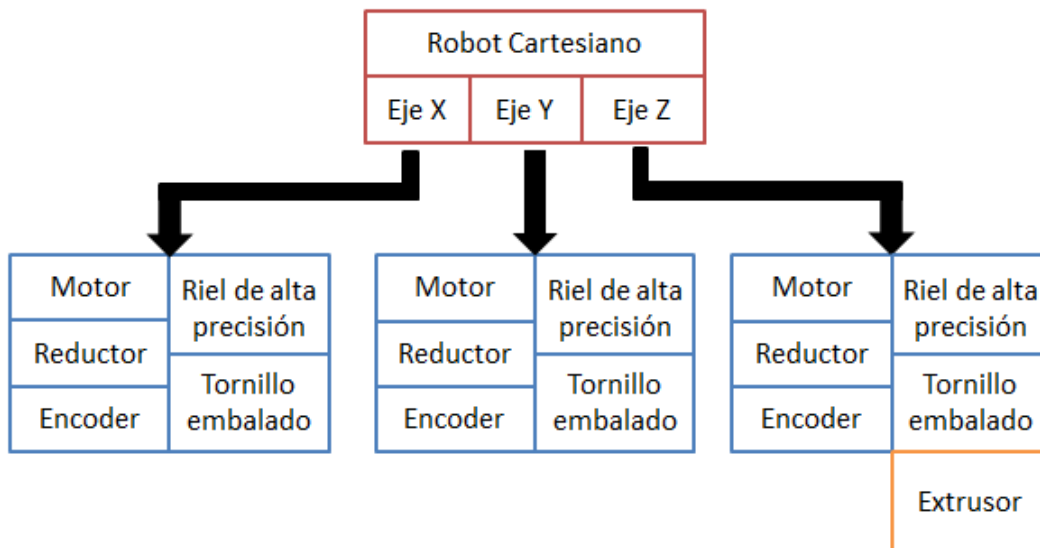


Figura 1.6: Diagrama a bloques de la mecánica del robot cartesiano.

1.3.1: Riel de alta precisión.

En nuestro manipulador se utilizaron rieles y carros lineales de alta precisión, con el fin de obtener el mejor desplazamiento lineal posible con una baja fricción.

Algunas de las características de estos elementos mecánicos son:

- Alta precisión en posicionamiento con alta repetitividad.
- Baja resistencia a la fricción. Esto ayuda a mantener su alta precisión por mayor tiempo.
- Alta rigidez, debido a su diseño de carga en apoyo sobre 4 puntos de la guía. [9]



Figura 1.7: Riel de alta precisión.

1.3.2: Tornillo embalado.

Como se mencionó anteriormente se ocuparon tornillos embalados para transformar el desplazamiento giratorio a lineal. Los tornillos embalados son capaces de desplazar grandes cargas, aunque para nuestra aplicación no es necesario, se pueden utilizar para otra aplicación más robusta, la cual requiera de esta cualidad.

Para nuestro robot manipulador fue importante utilizar este tipo tornillos, ya que también cuentan con la característica de reemplazar la fricción deslizante por una fricción de rodadura, con lo cual se incrementa la eficiencia del tornillo de 30% a un 90% ayudando a que el esfuerzo en los componentes motrices se vea disminuido significativamente y haciéndolos más durables. [10]

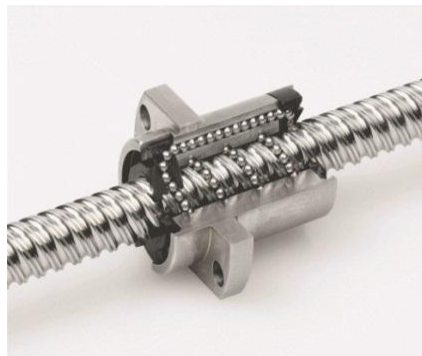
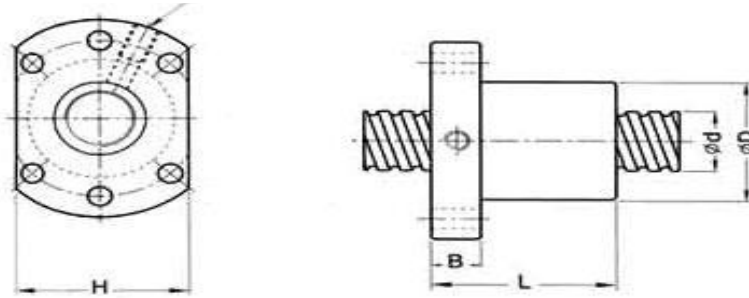


Figura 1.8: Tornillo embalado.



l: Lead Da: Ball Dia. n: Number of Circuits K: Stiffness (Kgf/m)

Type	Dimensions								
	d	l	Da	D	A	B	L	W	X
1605-3	16	5	3.175	28	48	10	42	38	5.5

unit: mm

Ca: Basic Dynamic Rating Load (Kgf) Coa: Basic Static Rating Load (Kgf)

G	H	Q	n	Ca	Coa	K
44	40	M6	4	780	1790	20

Figura 1.9: Dimensiones físicas del tornillo embalado.

1.3.3: Descripción de los motorreductores.

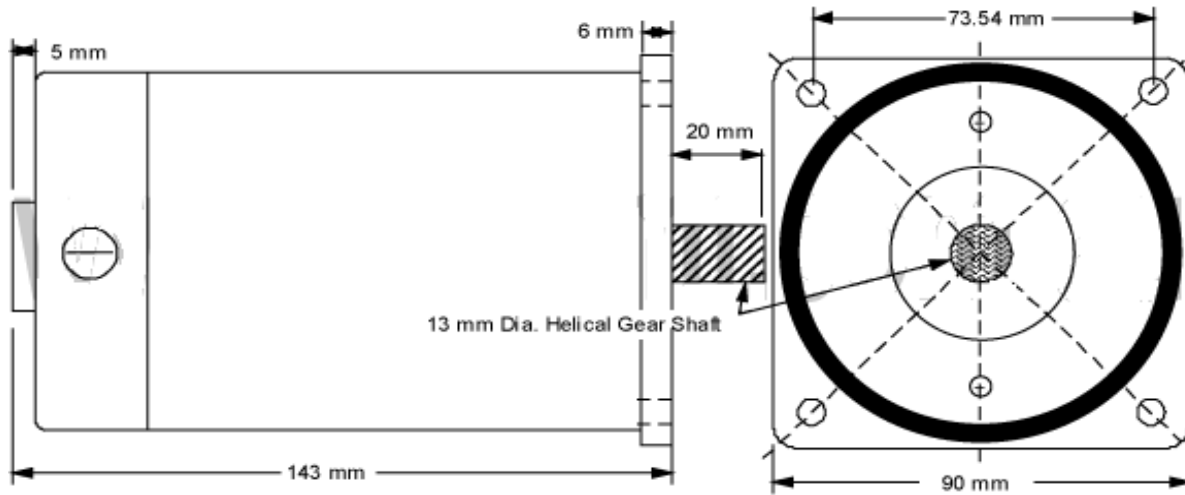
Para la selección de estos motorreductores se tuvieron tres consideraciones principales, el voltaje (12v), las revoluciones por minuto (más de 300) y finalmente la más importante el torque que se encontrara en un rango de 3 - 4Nm, esto para los ejes X y Y.

Con respecto al motor Z5D120-12 tenemos las siguientes características:

- Voltaje 12 v.
- Potencia 120 w.
- 3100 revoluciones por minuto.
- 2.2kg de peso. [11]

Modelo	Voltaje	Potencia	Parámetros sin carga		Parámetros con carga		Peso del motor
			Velocidad	Corriente	Velocidad	Corriente	
	V	W	r/min	A	r/min	A	kg
Z5D 120-12	12	120	3100	2	2500	15	2.2

Tabla 3: Características del motor Z5D120-12.



Z5D60-xxG to 120-xxG Series Motor

Figura 1.10: Dimensiones del motor.

Tamaño	3.54 pulgadas
Tipo de eje	Eje paralelo a la cabeza del reductor
Relación	9:1
Diámetro del eje de salida	15mm

Tabla 4.- Características básicas del reductor.



Figura 1.11: Reductor 5GU9K.

Ambos componentes tienen las siguientes características:

- La velocidad de salida del motorreductor es de 333 rpm.
- Tiene un torque de 3.21 N.m o 32.7 Kg^f*cm. [12]



Figura 1.12: Motor Z5D120-12 con reductor 5GU9K ensamblado.

Una vez adquiridos los motorreductores del robot CNC se procedió a caracterizarlos para determinar su curva característica, para poderlos utilizar como motores de transmisión directa (Direct drive), es decir, que el par que solicitemos en la entrada será el par de salida, mientras se encuentre en su rango de torque. Para esto utilizamos el freno prony.

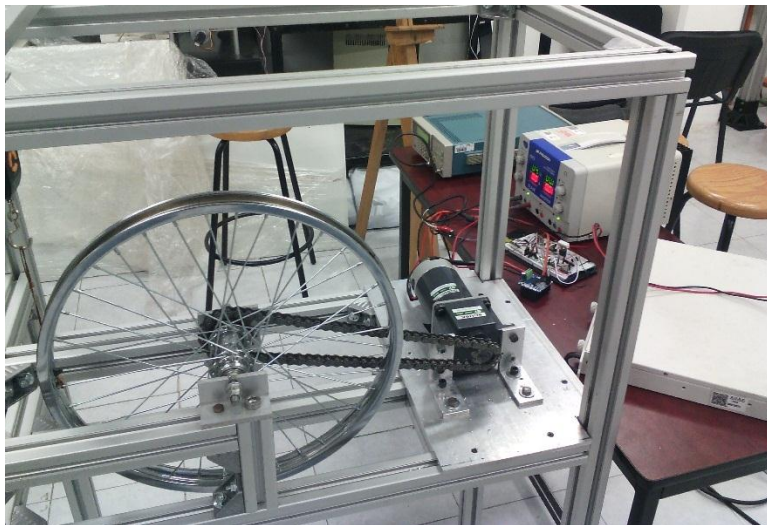


Figura 1.13: Freno prony.

El proceso consta de los siguientes pasos:

1. Determinar el torque máximo del motor variando el voltaje de alimentación, hasta el voltaje recomendado de trabajo.
2. Una vez encontrado el torque máximo, se buscará la mitad de ese torque máximo alimentado al motor con el voltaje del torque máximo, pero con una señal de PWM al 50% del ciclo de trabajo. Para conseguir esto, es necesario ir variando la frecuencia de la señal de PWM, desde los 50Hz hasta la frecuencia que determine el usuario, es decir, no tiene frecuencia máxima.
3. Se seleccionan las frecuencias o la frecuencia que nos proporcione el valor más cercano a la mitad del torque máximo.

4. Con cada una de las frecuencias seleccionadas se tendrá que ir variando el ciclo de trabajo desde el 10% hasta el 90% del ciclo de trabajo, alimentando al motor con el voltaje del torque máximo. Cada vez que se varíe el valor del ciclo de trabajo se tendrá que determinar el torque y graficar la respuesta del motor.
5. Se comparan las gráficas obtenidas y se determina la ecuación de la curva característica. Seleccionando la frecuencia que tenga nos ofrezca la respuesta más cercana a la lineal.

De esta manera, un motor convencional de DC podrá ser utilizado emulando un motor de transmisión directa. Aplicando el proceso anterior a estos dos motores se obtuvieron los siguientes resultados. Para el motor del eje X, se seleccionó una frecuencia de 2.45 KHz. Teniendo los siguientes resultados:

% Ciclo de trabajo de PWM	Torque (Nm)
0	0
10	2.8616
12	3.9494
14	4.3708
16	5.2528
18	5.9976
20	6.4288
22	7.595
24	8.1144
26	8.428
28	9.555
30	10.388
32	10.8192
34	11.6032
36	12.348

Tabla 5: Resultado del motor 1 (eje X).

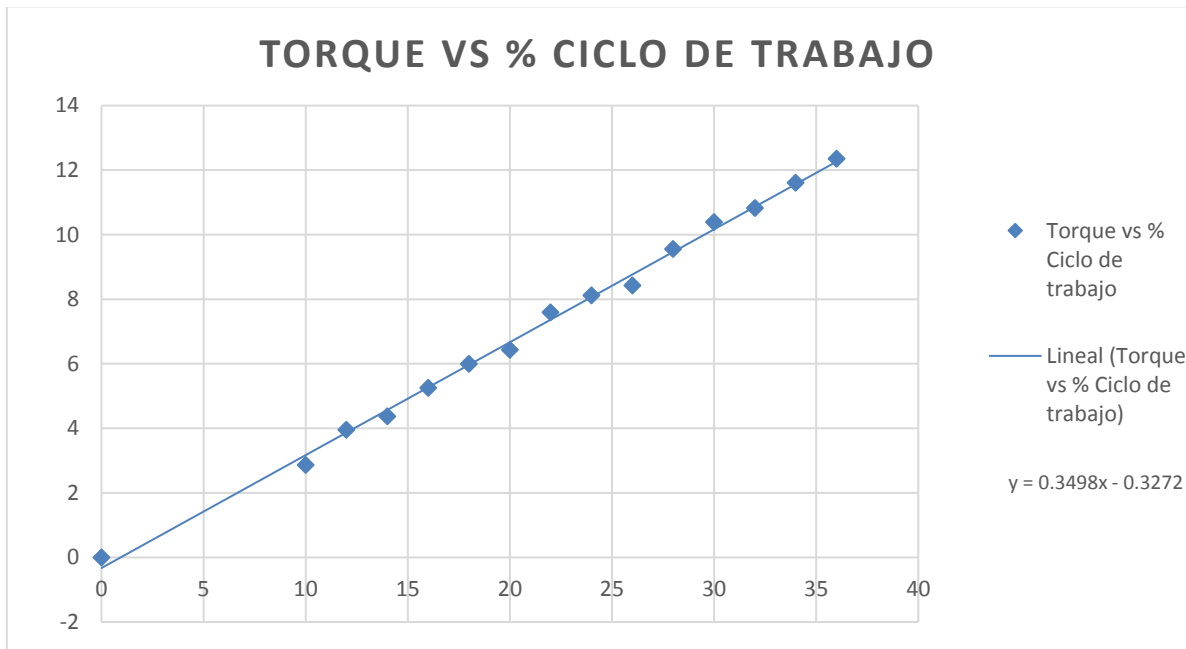


Figura 1.14: Gráfica resultante de la Tabla 4 (Eje X).

Para el motor del eje Y, se seleccionó una frecuencia de 2.5 KHz. Teniendo los siguientes resultados:

% Ciclo de trabajo de PWM	Torque (Nm)
0	0
10	3.3124
12	4.2924
14	5.1058
16	5.8016
18	6.3406
20	6.8208
22	7.6832
24	8.477
26	9.31
28	9.996
30	11.0152
32	11.7992
34	12.348

Tabla 6: Resultado del motor 2 (eje Y).

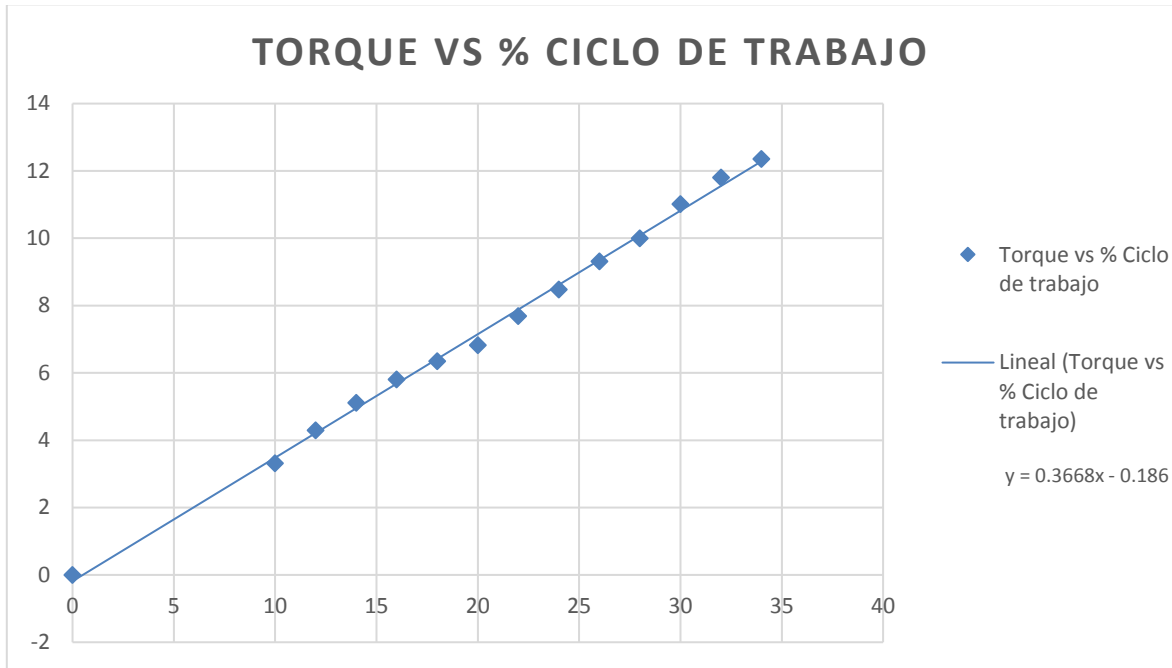


Figura 1.15: Gráfica resultante de la Tabla 5 (Eje Y).

Para el eje Z se empleó otro tipo de motor, el modelo Z4D40-12GN-30S, que cuenta con las siguientes características:

- ❖ Velocidad (rpm): 3100 rpm (No load).
- ❖ Voltaje: 12 volts.
- ❖ Potencia de salida: 40W.

Modelo	Voltaje	Potencia	Parámetros sin carga		Parámetros con carga		Peso del motor
			Velocidad	Corriente	Velocidad	Corriente	
	V	W	r/min	A	r/min	A	kg
Z4D 40-12	12	40	3100	2	2800	5.2	1.8

Tabla 7.- Características del motor Z4D40



Figura 1.16: Motor Z4D40-12GN-30S con reductor 4GN15K.

Y con el reductor 4GN15K, obtendremos las siguientes características:

- Velocidad: 190rpm.
- Torque: 1.01 Nm o 10.3 kgf*cm.

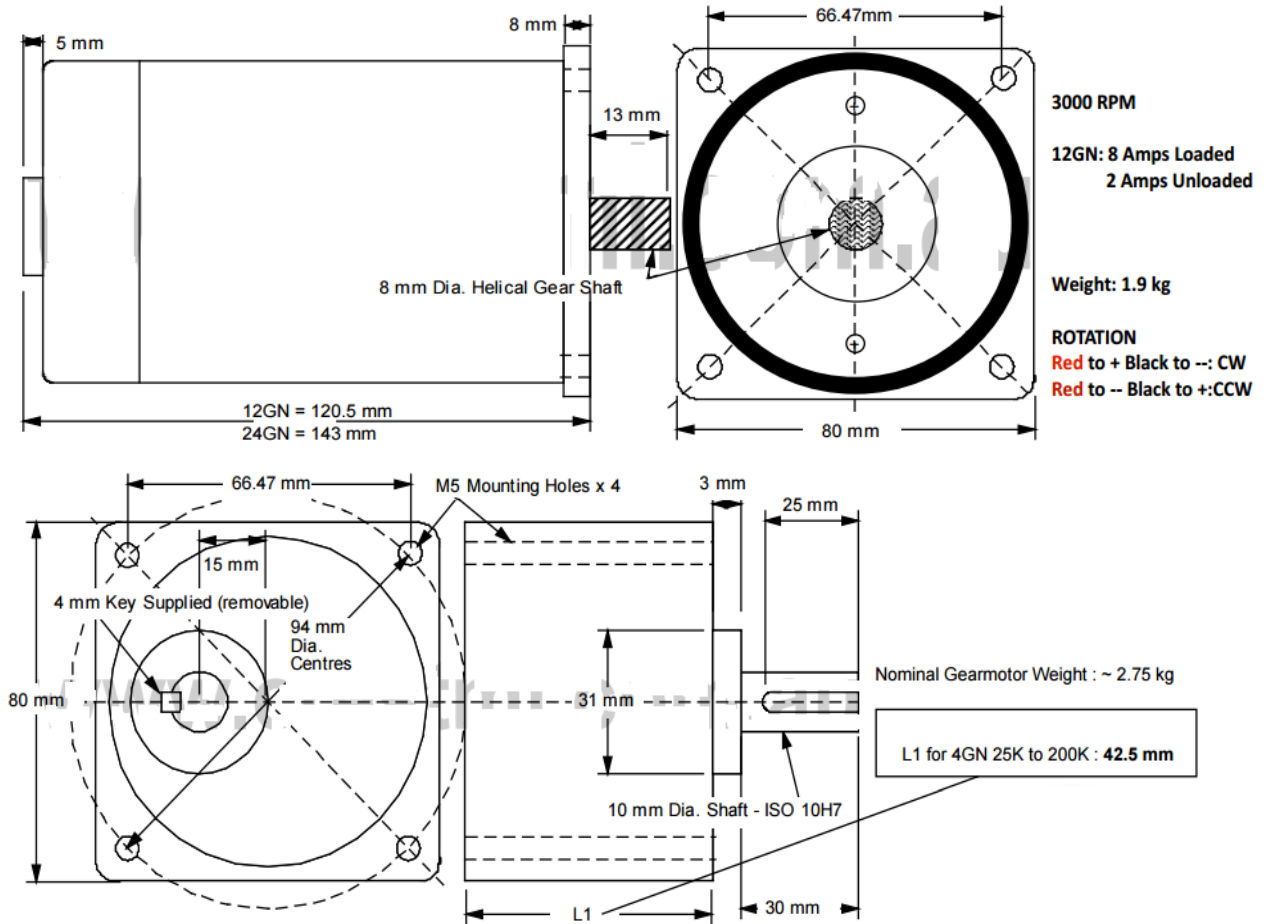


Figura 1.17: Dimensiones motor Z4D40-12GN-30S con reductor 4GN15K.

De forma análoga a los motores de los ejes X y Y, se realizó el proceso de caracterización obteniendo los siguientes resultados, ocupando una frecuencia de trabajo de 75Hz.

% Ciclo de trabajo de PWM	Torque (Nm)
0	0
10	0.0616077
20	0.3435096
30	0.6571488
40	0.840105

50	1.250823
60	1.437513
70	1.8930366
80	2.072259
90	2.5688544

Tabla 8: Resultado del motor 3 (eje Z).

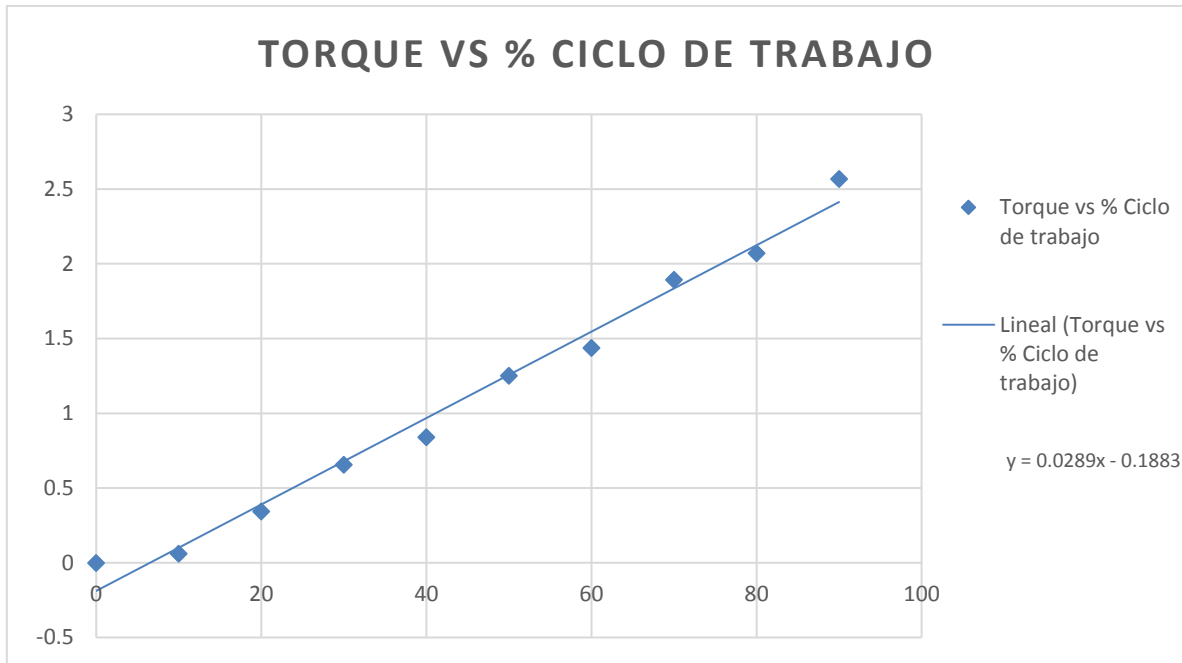


Figura 1.18: Gráfica resultante de la Tabla 6 (Eje Z).

Con lo anterior se justifica que los motores DC, tendrán un comportamiento lineal o muy cercano al lineal, si se cumplen las condiciones de eléctricas y de programación necesarias.

Resolución teórica del CNC.

En base a las características mecánicas con las que cuenta nuestro robot CNC, que más adelante se describirán, podemos decir que la resolución teórica de los ejes X y Y, es la siguiente:

$$\Delta_{x,y} = \frac{L_{x,y}}{M * R * C} = \frac{10mm}{1 * 9 * 4000} = .277\mu m \quad , \quad (1)$$

dónde:

$\Delta_{x,y}$ = Resolución teórica en el eje X, Y.

$L_{x,y}$ =Desplazamiento del eje X, Y.

M= Número de vueltas necesarias para desplazar $L_{x,y}$.

R = Reducción de la caja de engranes del motor

C = Resolución de conteos por vuelta del motor.

Para el eje z tenemos lo siguiente:

$$\Delta_z = \frac{L_z}{M * R * C} = \frac{10mm}{1 * 15 * 4000} = .166\mu m , \quad (2)$$

dónde:

Δ_z = Resolución teórica en el eje Z.

L_z =Desplazamiento del eje Z.

M= Número de vueltas necesarias para desplazar L_z .

R = Reducción de la caja de engranes del motor

C = Resolución de conteos por vuelta del motor.

1.3.4.- Encoder giratorio GTS06-VE-RA 1000.

Para poder ocupar los motores DC como servomotores fue necesario acoplar un encoder giratorio en la parte posterior de los motores y así poder determinar la posición actual de nuestro actuador. Los encoders utilizados fueron los GRTS06 que cuentan con las siguientes características:



Figura 1.19: Encoder giratorio GTS06-VE-RA 1000.

- 1000 pulsos/revolución.
- Alimentación de 5-24 vdc.

- Eje de 6mm de diámetro, 12mm de largo.
- Tamaño: 38mm de diámetro y 35.5 de largo.
- Salida: AB de cuadratura.
- Máxima velocidad mecánica: 5000 rev/min.
- Salida NPN tipo colector abierto (necesita una resistencia Pull-up, ya que, si es conectado a Vcc, las salidas se quemarían.) [13]

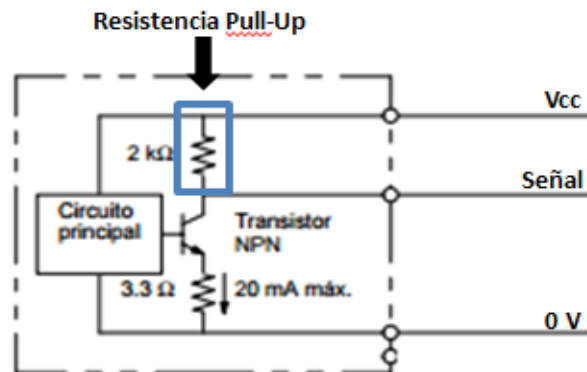


Figura 1.20: Ejemplo de encoder con salida NPN tipo colector abierto.

A continuación, se presenta la estructura mecánica de nuestro robot CNC, donde se muestra como se realizó el acoplamiento de los encoders con los motorreductores, y de los motorreductores al tornillo embalado. Además de cómo se realizó la sujeción del extrusor de material y del alimentador, que son nuestras herramientas de impresión 3D.

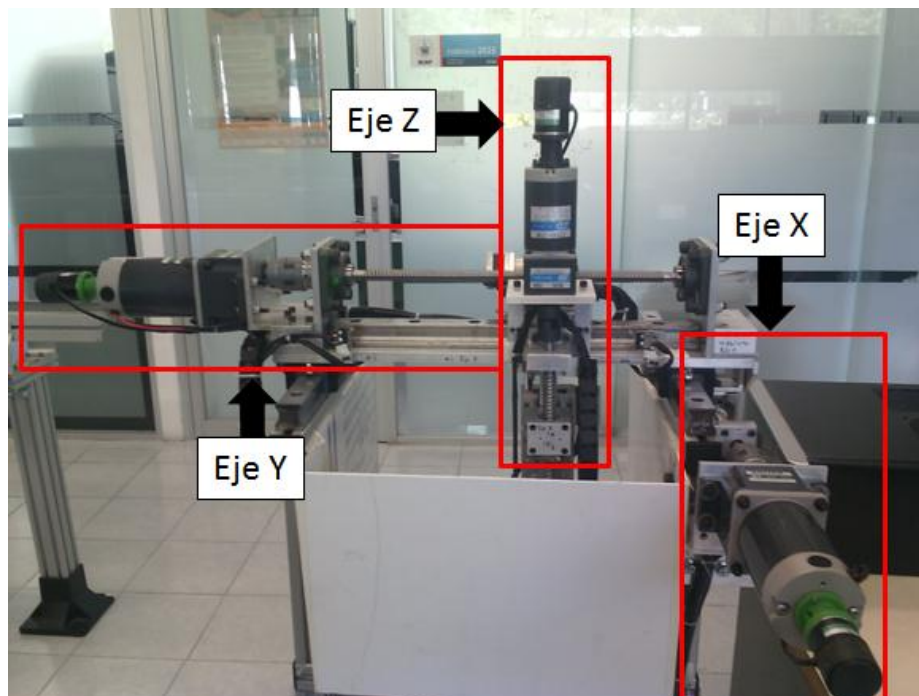


Figura 1.21: Robot CNC con herramienta 3D.

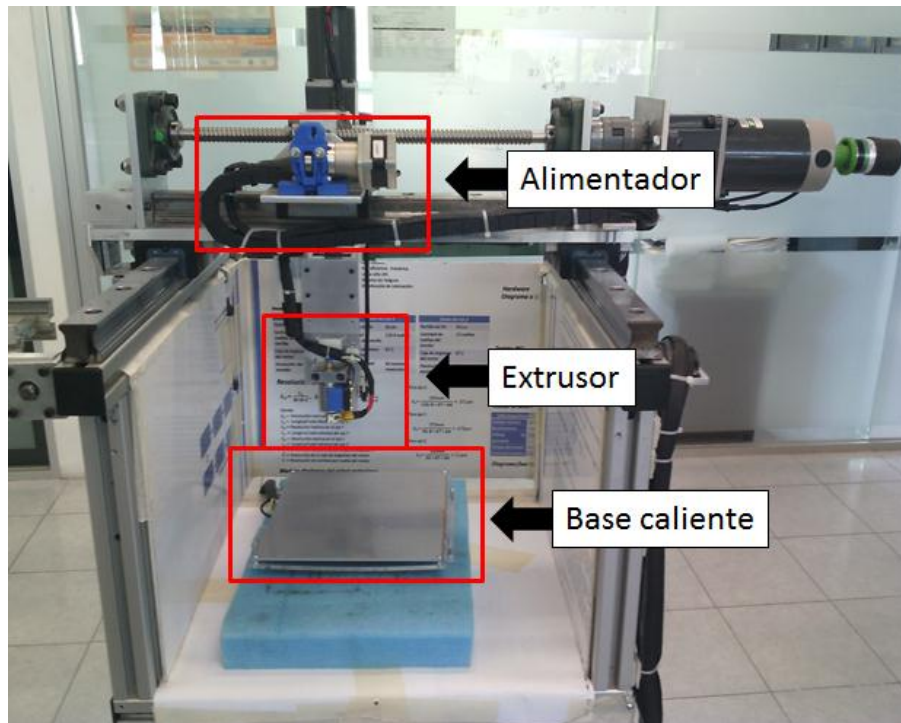


Figura 1.22: Robot CNC con herramienta 3D.

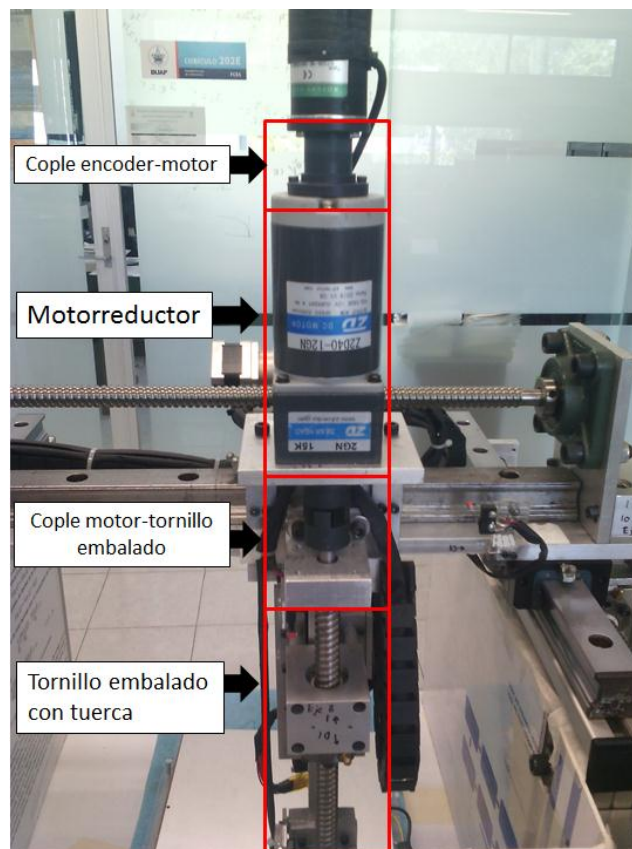


Figura 1.23: Eje Z.

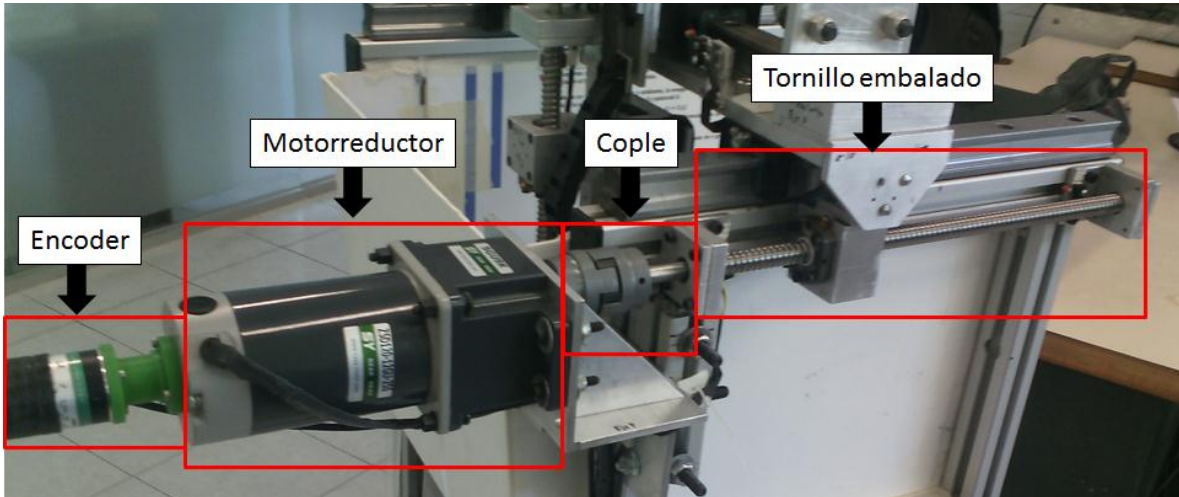


Figura 1.24: Eje X.

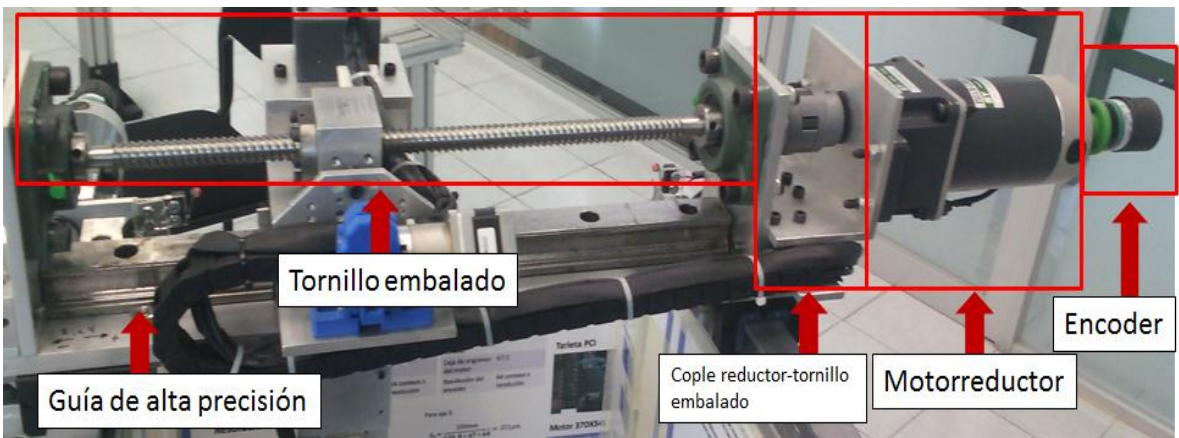


Figura 1.25: Eje Y.

De las imágenes anteriores podemos ver cómo están ensamblados mecánicamente los ejes de nuestro robot manipulador.

1.4. Electrónica del robot

Al actualizar la mecánica de nuestro CNC, fue necesario actualizar también la electrónica del sistema, ya que los motorreductores nuevos requerían de una etapa de potencia que soportara la demanda de corriente de los mismos. Por lo que se optó por implementar puentes “h” en configuración darlington que al final activarán unos transistores Tip 35C (NPN) y Tip 36C (PNP), que pueden drenar hasta 25A, en modo de saturación y corte.

Para realizar la configuración darlington se utilizaron transistores convencionales, 3904 (NPN) y 3906 (PNP), sin embargo, para poder saturar los transistores Tip 35C y 36C se ocuparon otros transistores Tip 35C y 36C, pero en modo de amplificación ya que se requiere de 1.5A para saturar a 15A los transistores Tip.

A continuación, se presenta el diagrama esquemático de los puentes “h”, implementados para los 3 motorreductores:

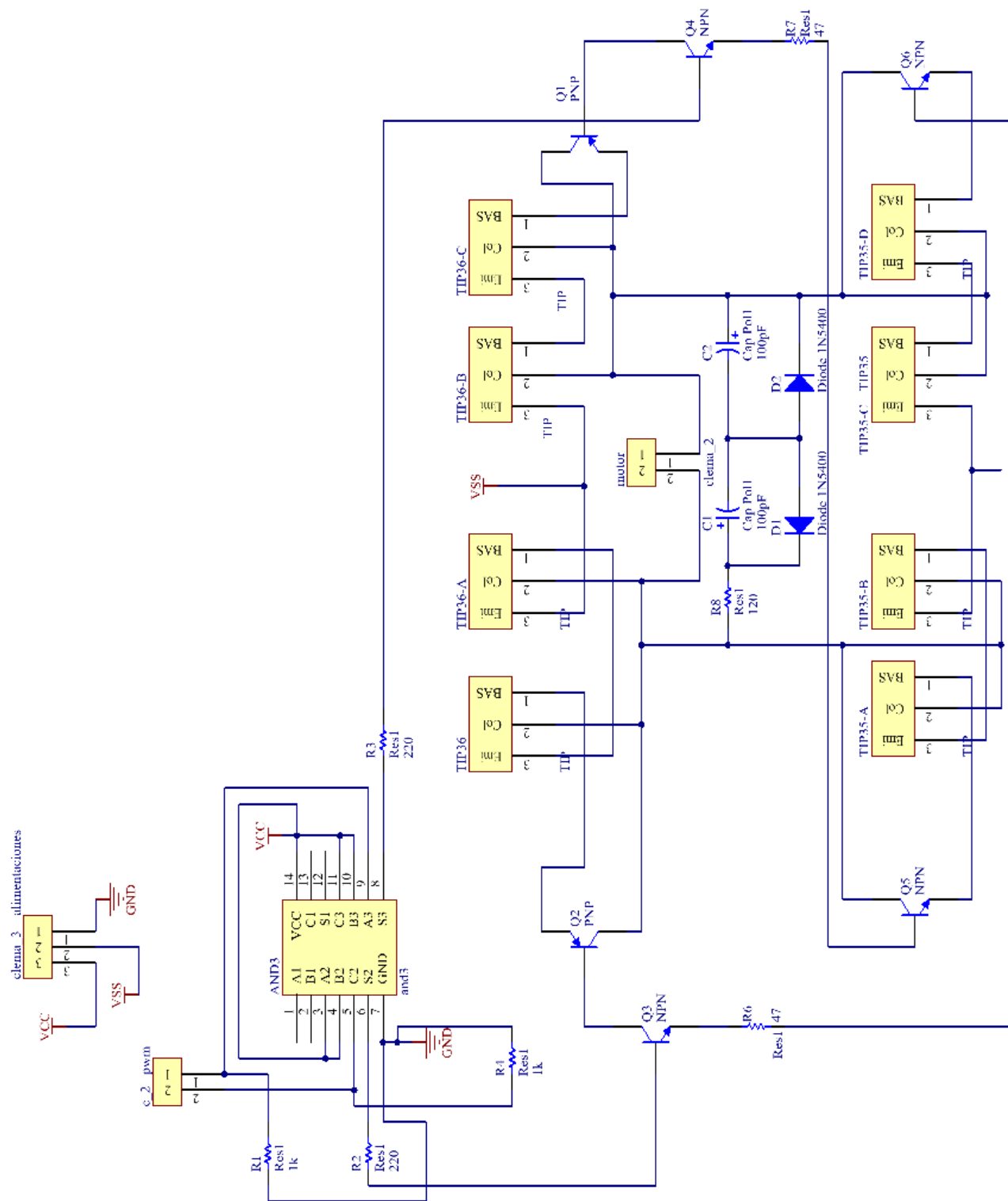


Figura 1.26: Puente “H”

Como se puede ver en el diagrama anterior se tiene una configuración resistencia-capacitor (red snubber), conectado al motor, esto con el afán de reducir los sobretiros a la hora de energizar el motor, esto durante la conmutación de la señal. Además, se maneja una compuerta AND3 la cual recibe la señal del FPGA, para después pasarla a nuestra etapa de potencia, sirviendo como etapa de acoplamiento de señal.

Para la lectura de los encoders, se tuvo que hacer un divisor de voltaje, ya que los encoders manejan un voltaje de 5V en sus señales de salida y el FPGA, solo maneja un voltaje en un rango de 3.2-3.4V, para ello se empleó el siguiente circuito:

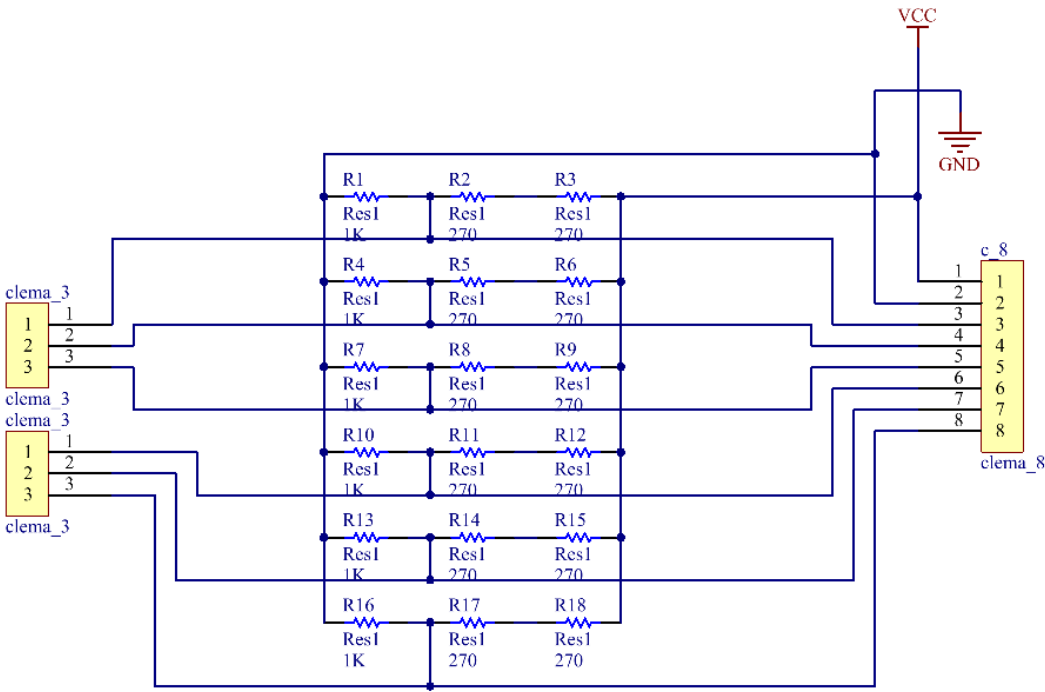


Figura 1.27: Divisor de voltaje

Se utilizaron resistencias de 1kΩ y de 270Ω para obtener un voltaje final de 3.24V, el cual puede leer nuestro FPGA.

Para controlar nuestro alimentador se utilizó una tarjeta de que tiene como base un L298, este componente maneja dos puentes “H”, los cuales si se controlan de manera correcta nos permiten enviar la secuencia a nuestro motor a pasos bipolar.

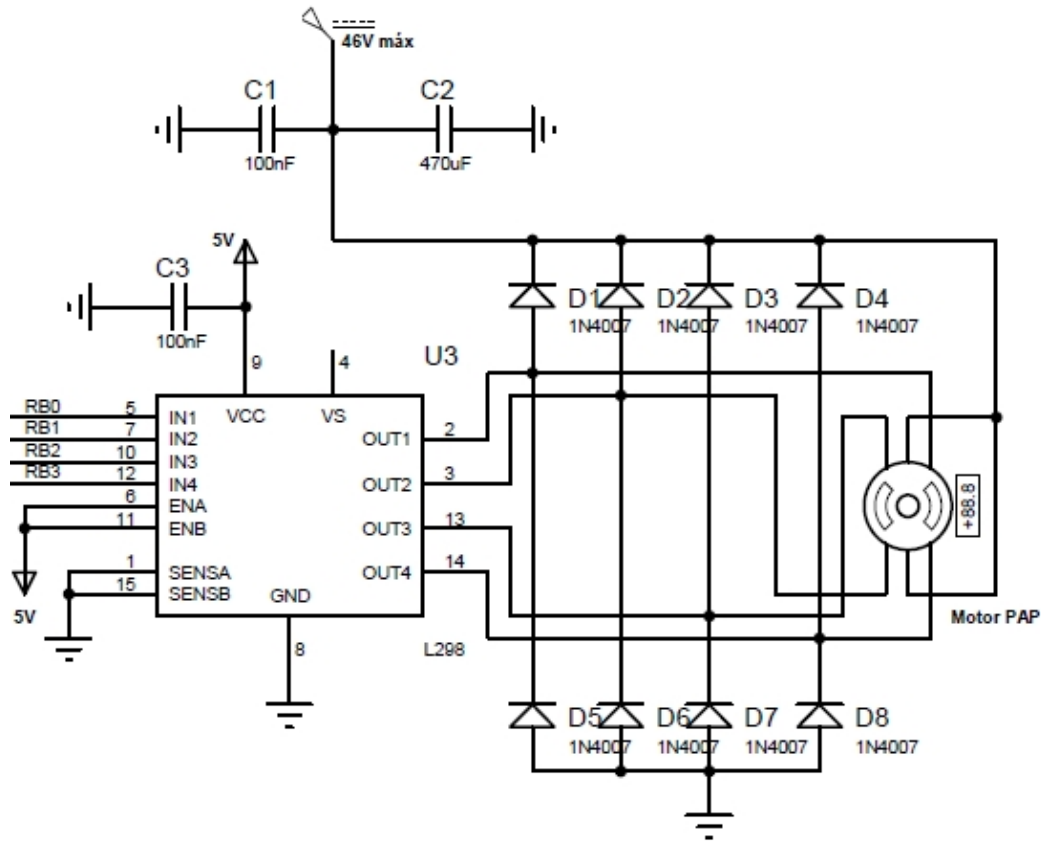


Figura 1.28: Circuito del alimentador de material.

Del circuito anterior tenemos que las salidas de control de la secuencia del FPGA se conecta a las entradas IN1, IN2, IN3 e IN4, y las salidas a las dos bobinas de nuestro alimentador de material.

Por otro lado, para nuestro caso de nuestros termistores se utiliza un divisor de voltaje para poder interpretar la variación de resistencia de nuestro termistor, y esta señal posteriormente pasa a un convertidor analógico-digital (ADC0804), que nos proporcionará una señal digital de 8 bits con la cual nuestro FPGA, puede realizar el algoritmo de control.

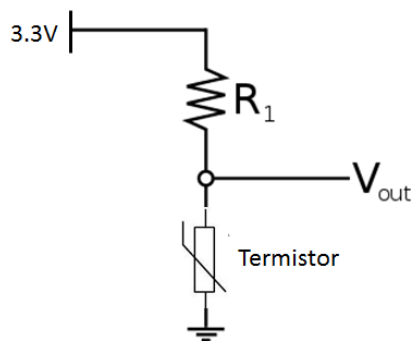


Figura 1.29: Divisor de voltaje termistor.

Para el extrusor se utilizó una resistencia (R1) de 4.45kΩ y para la base caliente se utiliza una resistencia (R1) de 672Ω, ya que con estas resistencias se obtuvieron los mejores resultados de sensado.

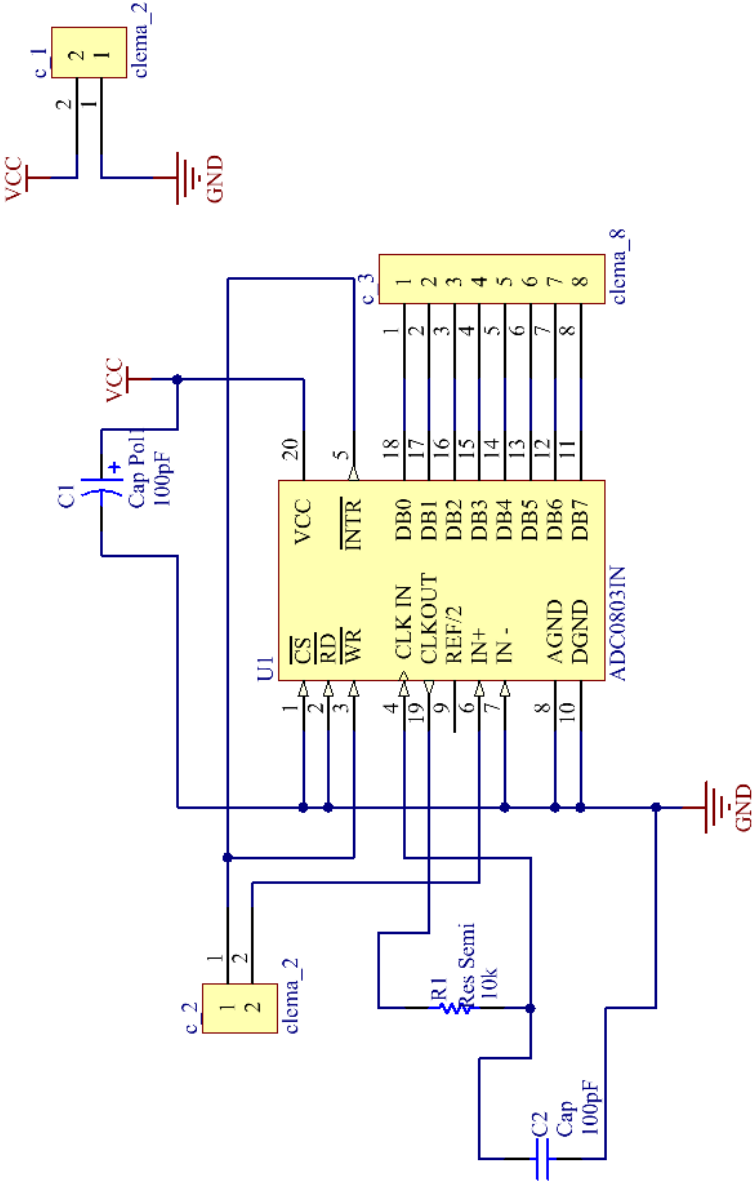


Figura 1.30: Circuito para ADC.

Se utiliza una resistencia Pull Down a la entrada del FPGA, donde se leen las señales de nuestro ADC, esto debido a que el FPGA tiene por default sus entradas en tercer estado.

Para la etapa de potencia de nuestro extrusor y de la base caliente se emplean, transistores NPN en configuración Darlington para garantizar saturación, de forma análoga a la de los puentes “h” implementados.

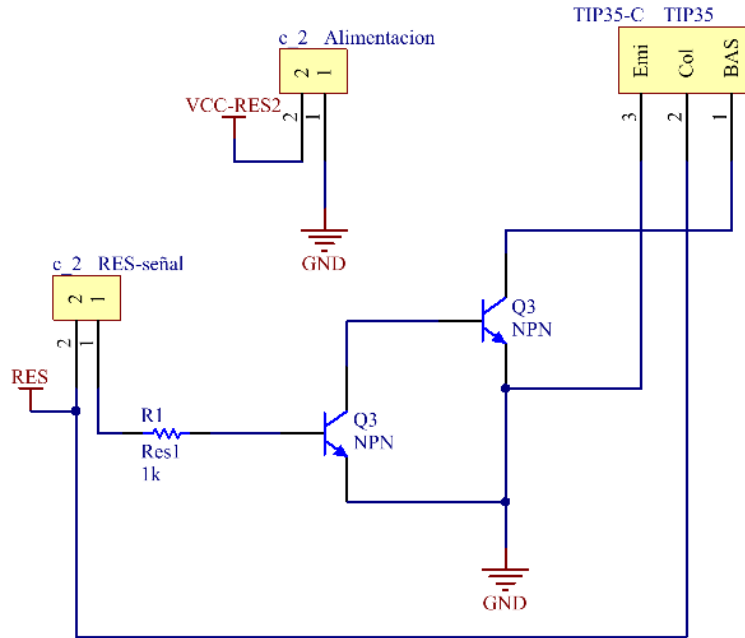


Figura 1.31: Darlington para extrusor y base caliente.

Finalmente, para alimentar toda nuestra etapa de potencia se utiliza una fuente conmutada convencional de computadora de 800 watts, lo que se realizó para aprovechar toda la potencia de nuestra fuente es unir todas sus salidas de voltaje del mismo valor en un solo nodo, para ello empleamos clemas de montaje din para poder distribuir la alimentación de la fuente.

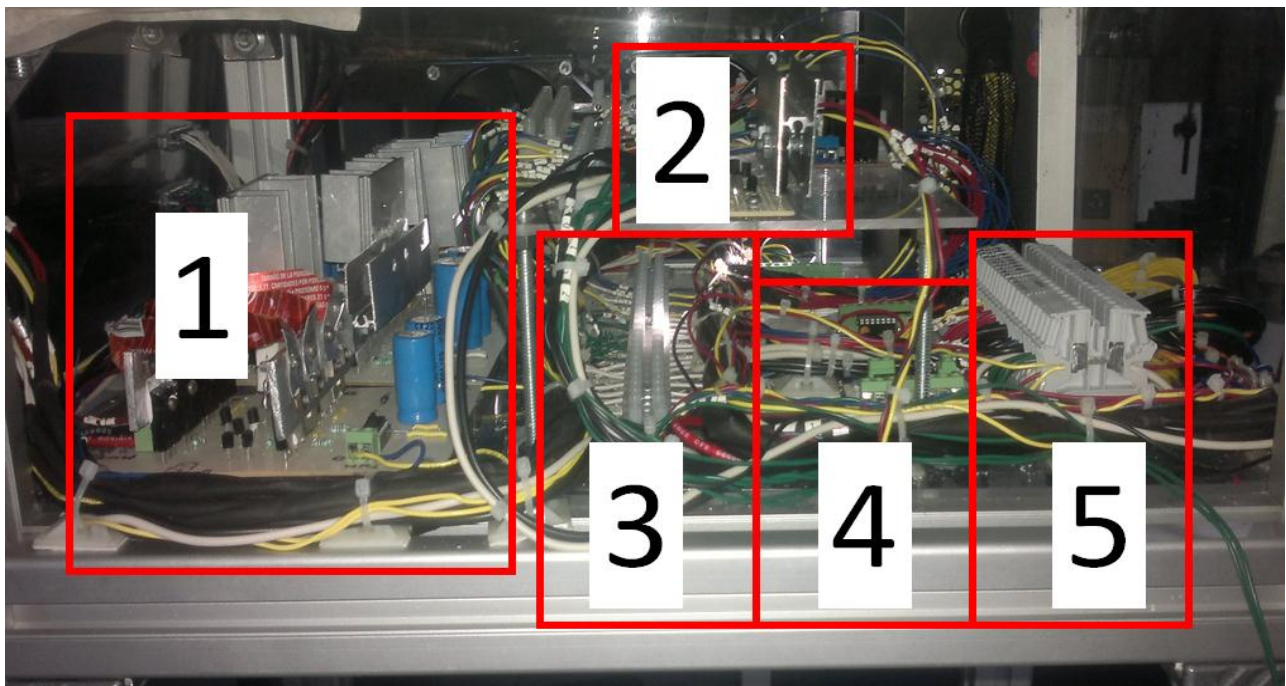


Figura 1.32: Tablero eléctrico del sistema

En la figura anterior se tiene cinco partes del tablero de control, la primera abarca los puentes “h” de cada eje, la segunda la etapa de potencia del extrusor y de la base caliente, en la tercera se tiene clemas de tornillo para distribuir las señales que se envían al FPGA, en la cuarta parte se tiene la etapa de los dos ADC, uno para la base caliente y otro para el extrusor, en la quinta parte se muestran las clemas de distribución de la alimentación general del sistema.



Figura 1.33: Vista lateral derecha del tablero de control.



Figura 1.34: Vista posterior del tablero de control.

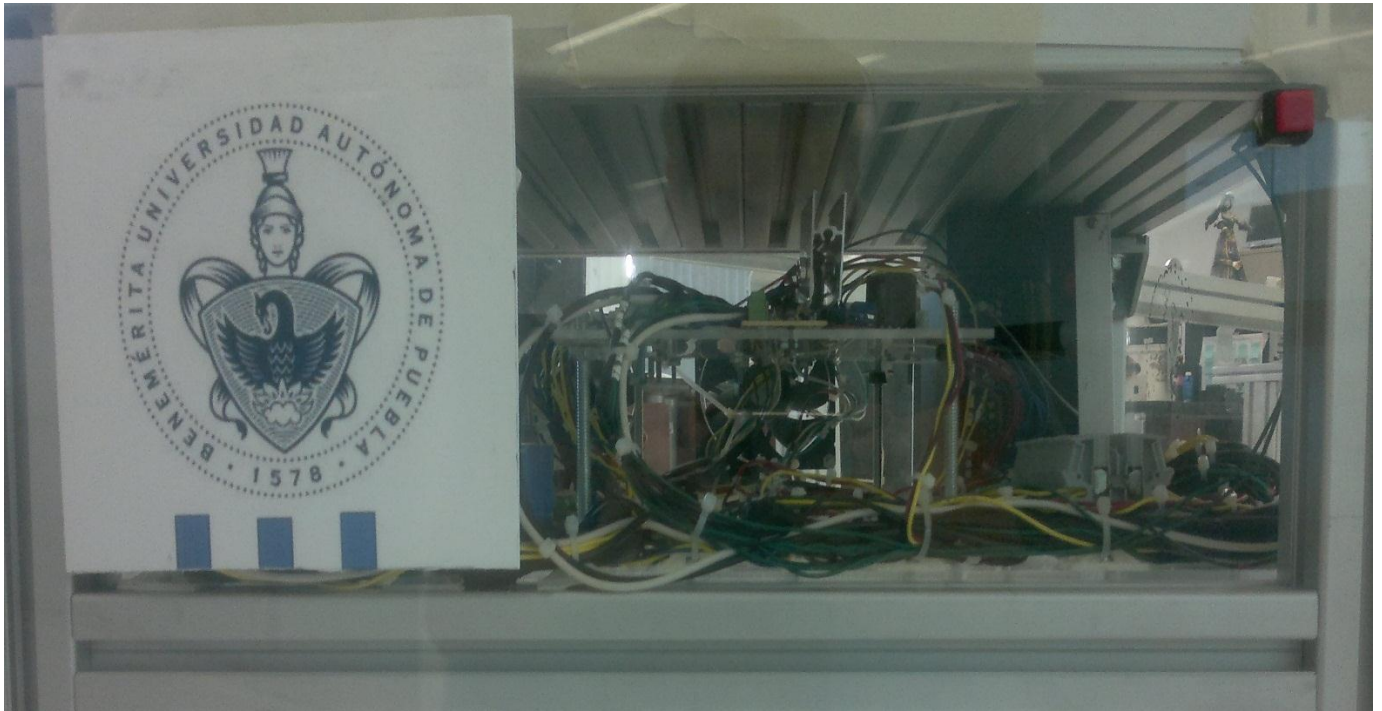


Figura 1.35: Vista frontal del tablero de control.

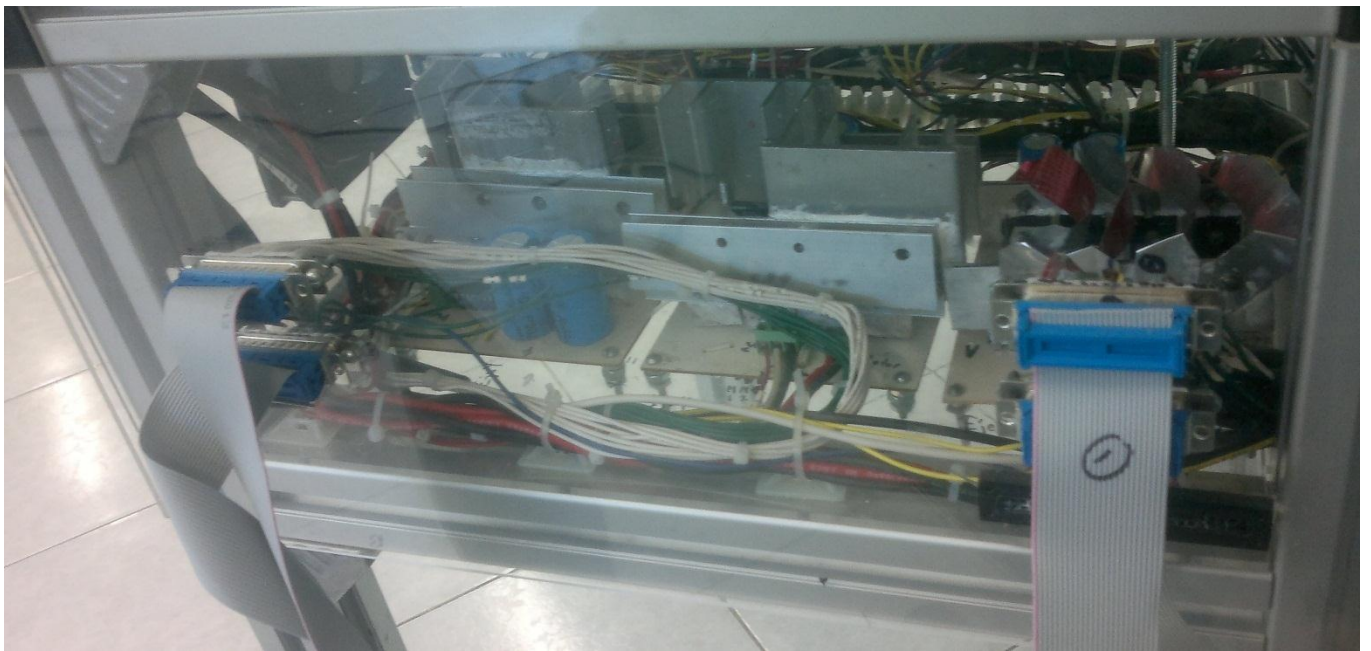


Figura 1.36: Vista lateral izquierda del tablero de control.

Para comunicar nuestro FPGA con la etapa de potencia, se empleó cable plano de 50 pines, que se conecta a la tarjeta hija (directo al FPGA) de la tarjeta Fotón 1, y posteriormente a dos conectores din 25, los cuales llegan a nuestra etapa de potencia.



Figura 1.37: Tarjeta fotón_1 con tarjeta hija y conexiones a los bancos 1 y 3 del FPGA.



Figura 1.38: Tarjeta fotón_1 en el puerto PCI.

1.4. Conclusiones.

Al principio del capítulo, se realiza una descripción de las etapas que conforman nuestro robot manipulador, lo que ayuda a ubicar mejor como se realiza el manejo y control de nuestra impresora 3D.

Además, se describen las principales características de los componentes mecánicos del sistema, que nos ayudan a entender mejor el funcionamiento del robot cartesiano como impresora 3D. Con ayuda del modelo dinámico de nuestro robot cartesiano podemos entender mejor el comportamiento del sistema al aplicar diferentes algoritmos de control y elegir el óptimo para nuestra aplicación.

Finalmente, se menciona la electrónica del sistema, la cual es una parte fundamental de nuestro robot manipulador, la cual se tuvo que adecuar a las necesidades del sistema.

Capítulo 2. Firmware del sistema de movimiento

2.1 División del sistema

El firmware del sistema se divide en cuatro bloques, en los cuales se realiza el control de temperatura del extrusor y de la base caliente, además del control de velocidad del alimentador del extrusor y lo más importante, la lectura de la señal de los encoders para poder determinar su posición y posteriormente generar la señal de control de los motores DC, algo que tienen en común estas cuatro secciones es que todas emplean el puerto PCI como interfaz. En el siguiente diagrama a bloques se muestra esta división.

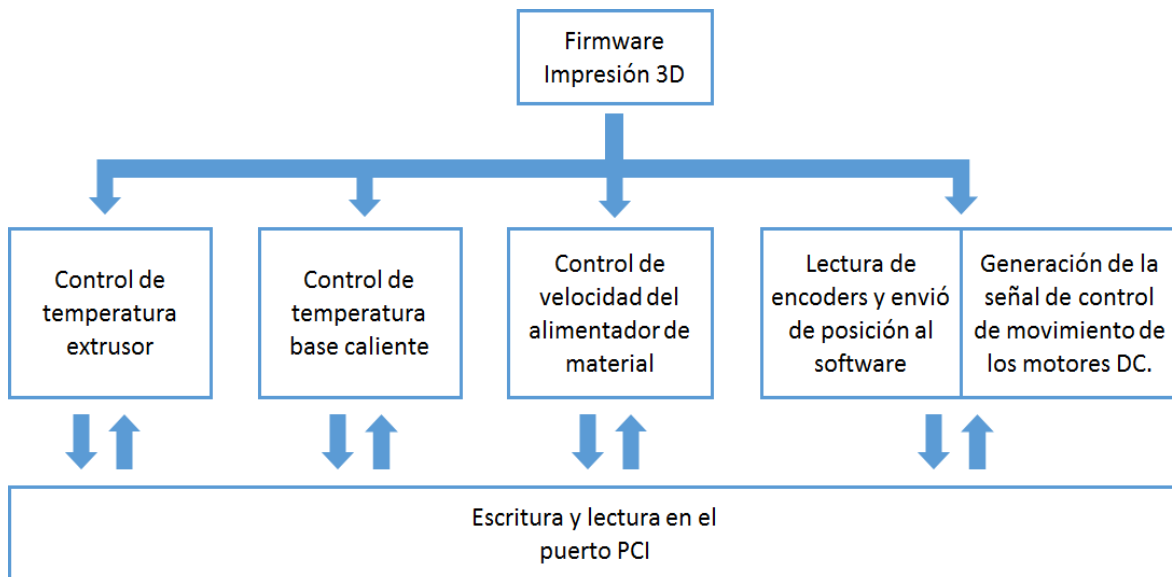


Figura 2.1: Diagrama a bloques del firmware del sistema.

En esta sección abordaremos el firmware de control de movimiento del robot CNC, además de la decodificación de los encoders y la generación de la señal de control, se mencionará el protocolo de comunicación del puerto PCI, siendo una parte básica en el firmware del sistema.

2.2 Sistema de transferencia de datos

Para la programación de nuestro FPGA EP3C10F256C6 de la familia Cyclone III, ocupamos el software Quartus II versión 13.0 web edition, realizando toda la programación en lenguaje AHDL. El firmware elaborado, tiene como parte principal el protocolo de comunicación del puerto PCI (lectura y escritura), ya que, utilizamos el FPGA para manejar el bus PCI, por

medio de la tarjeta de adquisición de datos, utilizando principalmente sus 32 señales de dirección y datos.

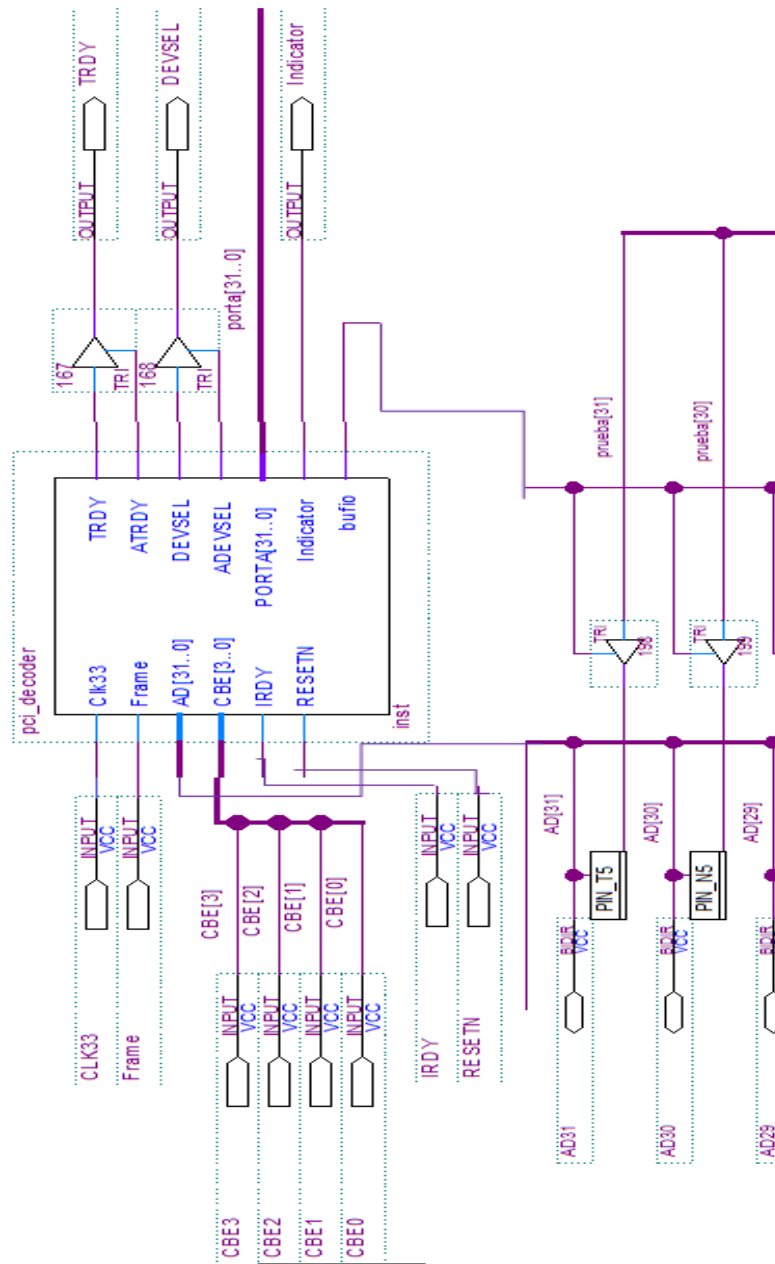


Figura 2.2: Fragmento del firmware del protocolo PCI.

El FPGA está conectado sobre una tarjeta hija, acoplado a una tarjeta de adquisición PCI, la cual se conecta al puerto PCI de la computadora, esto nos sirve como interfaz entre nuestro sistema físico y el software de control de nuestra impresora 3D.

Algunas características de la tarjeta de adquisición PCI son las siguientes:

- Conector header de 25 pines I/O en el banco 2.
- Puerto PCI de 32 bits en los bancos 3 y 4
- Voltaje de trabajo de 3.3V.

Esta tarjeta de adquisición es una parte fundamental en la electrónica de nuestro sistema, ya que por medio de sus I/O disponibles se reciben las señales de los encoders incrementales de cuadratura, estas señales se interpretan para proporcionarnos la posición actual del robot cartesiano en cada uno de los ejes. También se reciben las señales de los sensores de temperatura tanto del extrusor como de la base caliente que son herramientas necesarias para la realización de la impresión 3D.

Por medio de esta tarjeta también se envían las señales de PWM a la etapa de potencia, que son generadas mediante el software de control y el firmware programado en el FPGA.



Figura 2.3: Tarjeta fotón 1.

2.3 Sistema de conteo

En nuestro robot manipulador se implementaron servomotores, los cuales cuentan con encoders para poder terminar la posición de cada robot en cada eje. Por lo que se tuvo que realizar el firmware que nos permitiera poder interpretar las señales que nos proporcionan estos encoders. Esta parte del firmware se llama decoder, el decoder, es una máquina de estados que por medio de contadores nos indica cuantos pulsos hemos recibido de los encoders, además del sentido de giro, ya que en sentido horario el contador incrementará y en sentido inverso decrementará.

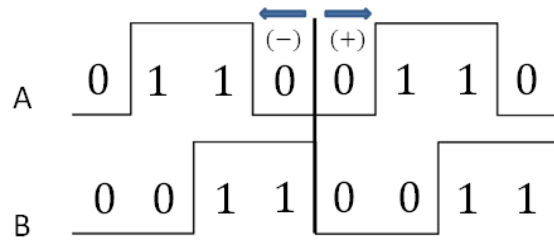


Figura 2.4: Combinaciones lógicas posibles del encoder incremental de cuadratura. Para ello, es necesario tener una señal de reloj que sea variable y este sincronizada con la velocidad de los motores.

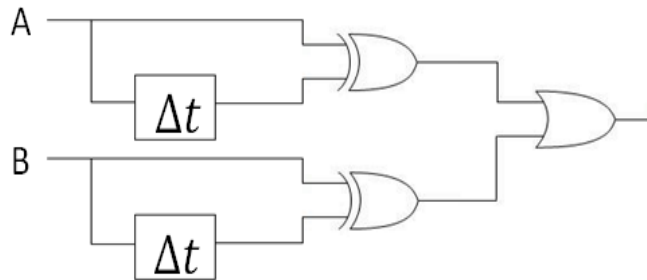


Figura 2.5: Circuito para la generación de la señal de reloj.

En el circuito anterior se muestra como a partir de las señales del encoder A y B, se genera la señal de reloj para nuestra máquina de estados. A continuación, se realiza la explicación más detallada del proceso.

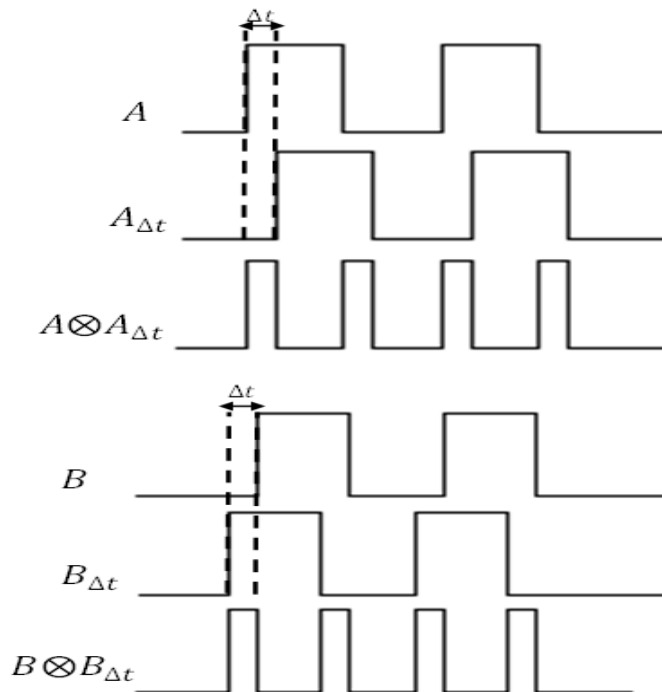


Figura 2.6 Diagrama de tiempos de las señales A y B.

En el diagrama de tiempos anterior se muestra como se obtienen las señales A, $A_{\Delta t}$, B y $B_{\Delta t}$, con estas señales se realiza la función XOR y se obtendrán $A \otimes A_{\Delta t}$ y $B \otimes B_{\Delta t}$, respectivamente. Después de esto se realiza la función OR para obtener la señal de reloj del decoder.

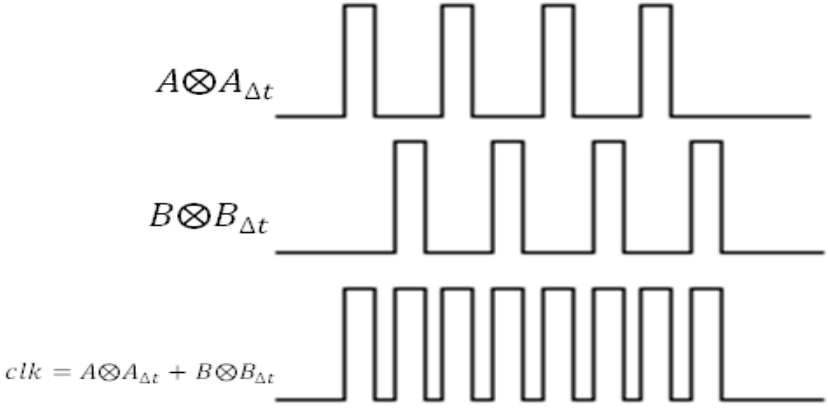


Figura 2.7 Señal de reloj generada.

La señal de reloj (clk), es creada con las señales A y B del encoder, lo que nos permite muestrearlas en nuestro decoder en todo momento, no importando la velocidad de giro del motor.

En el decoder, como se mencionó anteriormente se realizan conteos, para determinar la ubicación de nuestra herramienta. El siguiente diagrama de flujo expone el funcionamiento del decoder implementado.

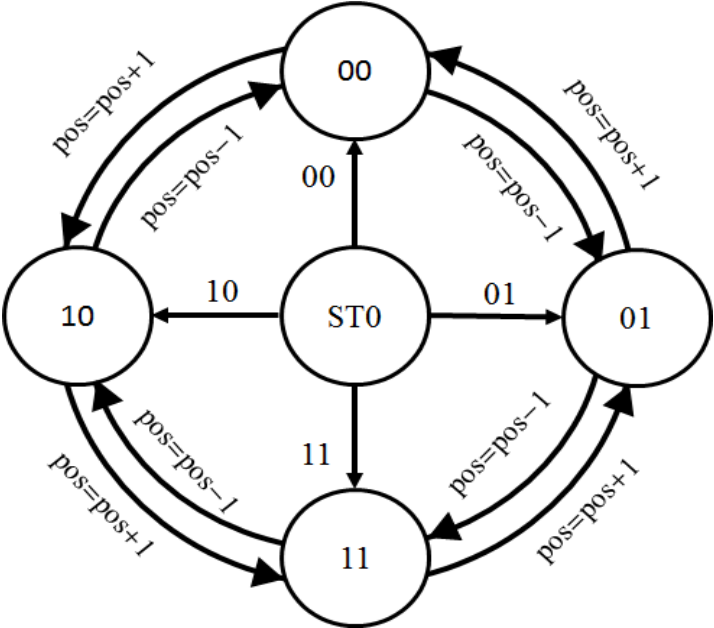


Figura 2.8 Diagrama de flujo del decoder.

Como se observa en el diagrama anterior se tiene un estado cero, el cual nos permite realizar el conteo de los pulsos del encoder a partir de cualquier combinación lógica de las señales A y B.

Al terminar este proceso tendremos la posición actual de nuestro robot manipulador, la cual será enviada por el puerto PCI para ser procesada por el software y utilizada por el algoritmo de control, para determinar el error de posición.

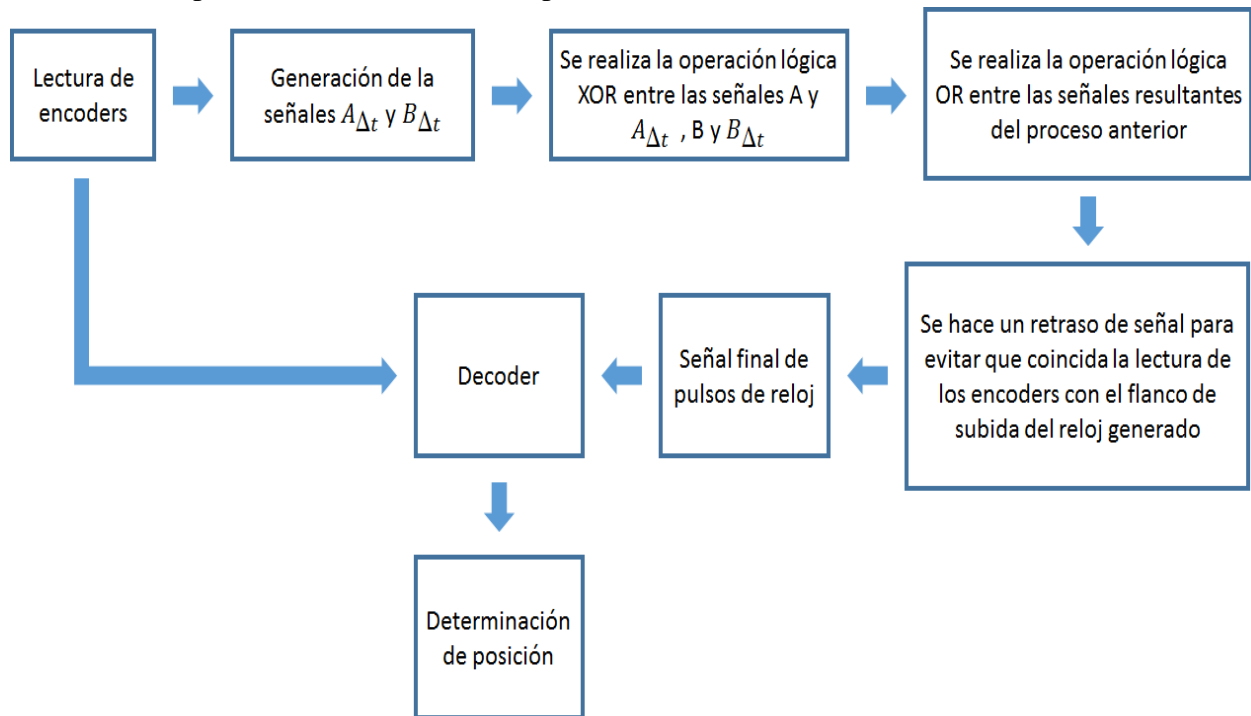


Figura 2.9: Diagrama a bloques del proceso completo para el conteo de pulsos.

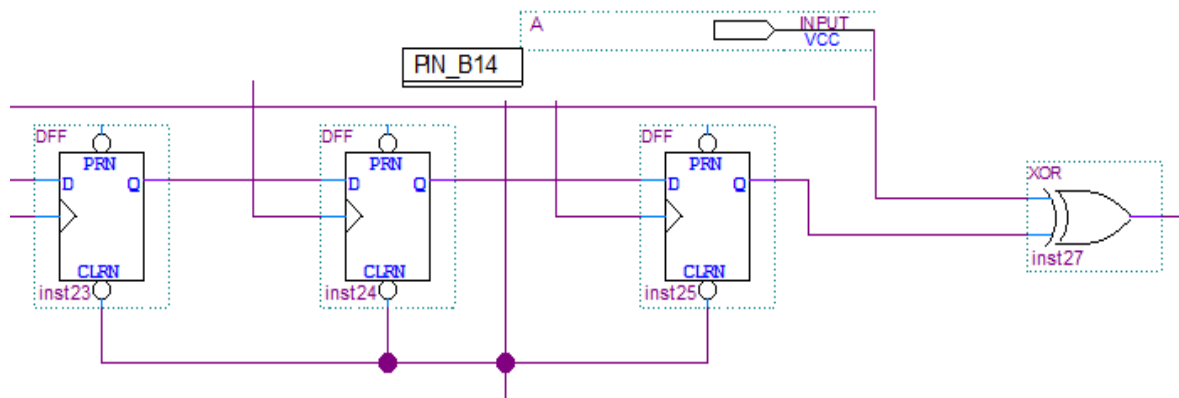


Figura 2.10: Diagrama a bloques para producir la señal $A_{\Delta t}$ y la operación lógica XOR.

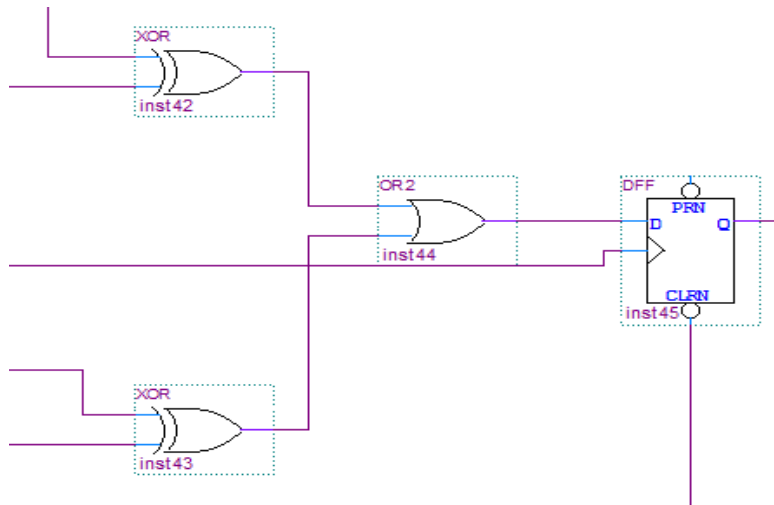


Figura 2.11: Etapa final del firmware para generar el pulso de reloj.

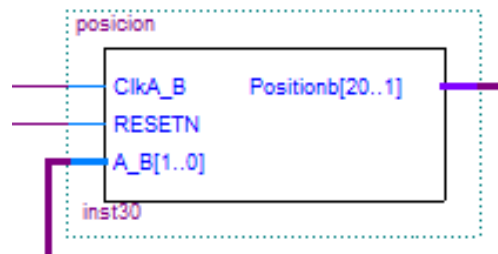


Figura 2.12: Bloque del firmware del decoder.

La posición actual será enviada en un dato de 20 bits, debido a la alta resolución que tiene nuestro sistema.

2.4 Generación de la señal de PWM

La modulación por ancho de pulsos es una técnica utilizada para regular la velocidad de giro de los motores eléctricos de inducción o asíncronos. Mantiene el par motor constante y no supone un desaprovechamiento de la energía eléctrica. Se utiliza tanto en corriente continua como en alterna, como su nombre lo indica, al controlar: un momento alto (encendido o alimentado) y un momento bajo (apagado o desconectado).

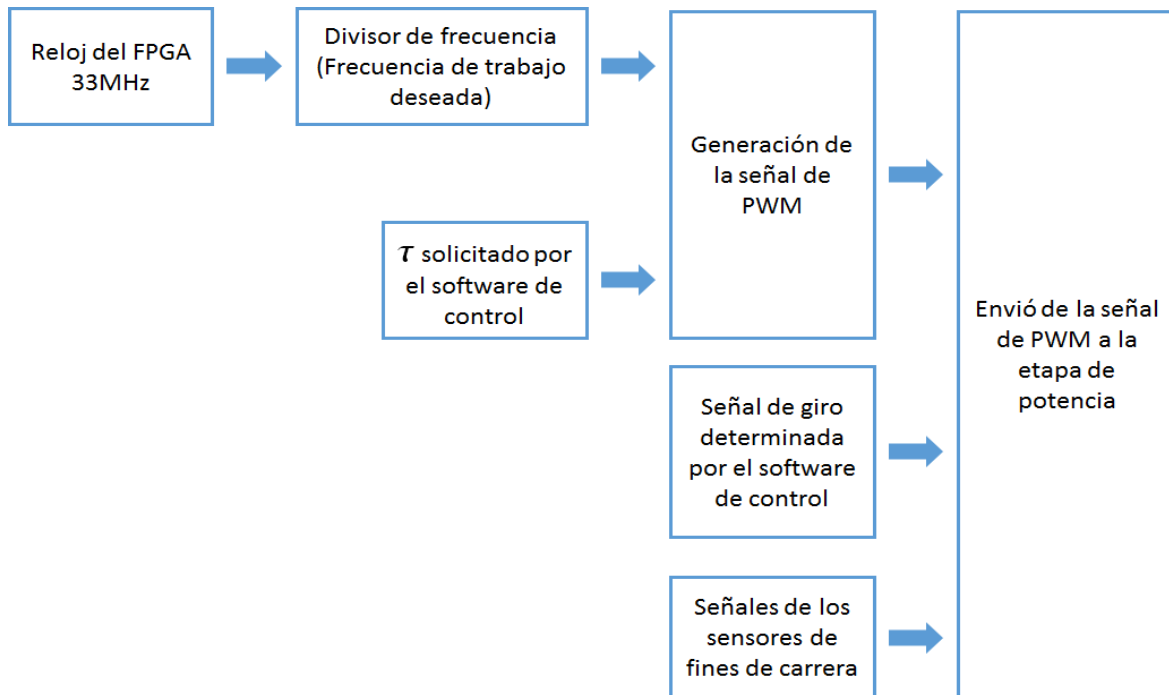


Figura 2.12: Diagrama del firmware generador de la señal de PWM.

Nuestro firmware cuenta también con una sección en la cual, se genera la señal de PWM que se envía a nuestra etapa de potencia, esta señal tendrá la frecuencia determinada durante la caracterización de los motores, para esto se tiene un bloque en el firmware que servirá como divisor de frecuencia, ya que nuestro FPGA funciona a una frecuencia de 33 MHz, que es la velocidad de transferencia del puerto PCI.

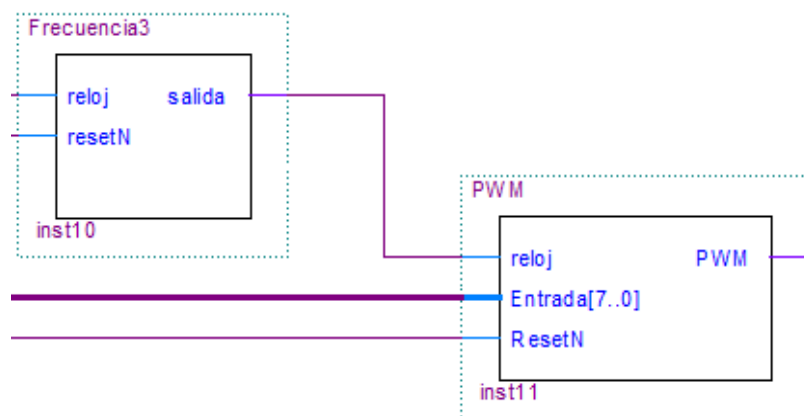


Figura 2.13: Divisor de frecuencia y bloque generador de la señal de PWM.

Cuando ya se ha programado la frecuencia de trabajo de nuestro motor, se pasa a la siguiente etapa donde se genera la señal de PWM. Para poder producir esta señal es necesario que el software de control envíe un dato de 8 bits sin signo por el puerto PCI al FPGA, el valor de este dato dependerá del control de posición que se realiza en el software.

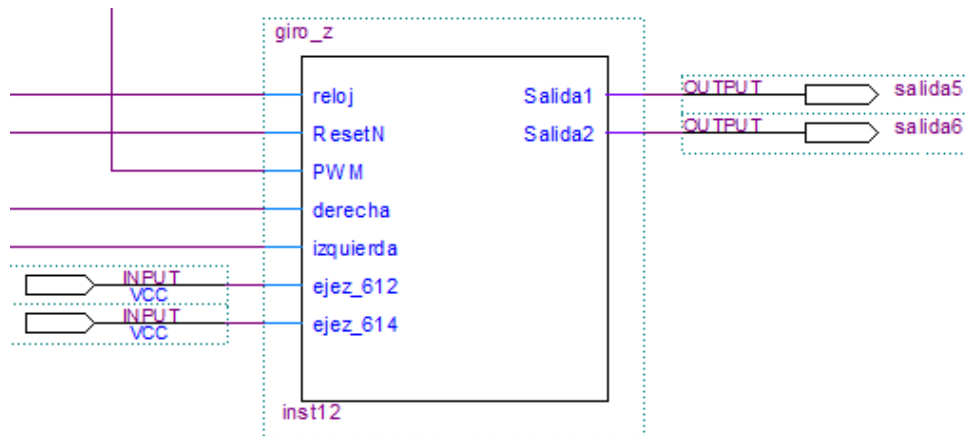


Figura 2.14: Bloque de envío de la señal de PWM a partir de la señal de giro.

Finalmente, esta señal de PWM pasa a un bloque donde se espera que el software envíe la señal del giro del motor, con esta señal el firmware determina el pin de salida de la señal de PWM. Además, se tiene control de las señales de fin de carrera, estas señales se activan solo en caso de que el robot CNC se desplace fuera del área de trabajo, evitando de esta manera colisiones.

2.5 Conclusiones

En este capítulo se menciona la importancia que tiene el protocolo de comunicación PCI, para nuestra tesis ya que, mediante este se realiza la interacción entre el software de control y el firmware. Además, se indica como es el proceso de decodificación de la señal de los encoders, sensores que nos sirven para ubicar la posición de nuestro manipulador en el espacio de trabajo, finalmente se explica brevemente cómo funciona el firmware de control de movimiento, el cual, genera la señal de PWM a partir del dato enviado por el software de control de posición, con la frecuencia ideal para cada actuador de nuestro robot CNC.

En esta pantalla nosotros podemos visualizar las posiciones deseadas, la posición actual, el error de posición, para cada eje, además del tiempo de ejecución del software y el número de línea que se está ejecutando en estos momentos.

Desde esta pantalla podemos cargar un archivo de texto “.txt”, con las posiciones deseadas, podemos también modificar el tiempo de muestreo de nuestro software. Podemos también tener acceso al botón de salida de la secuencia de control.

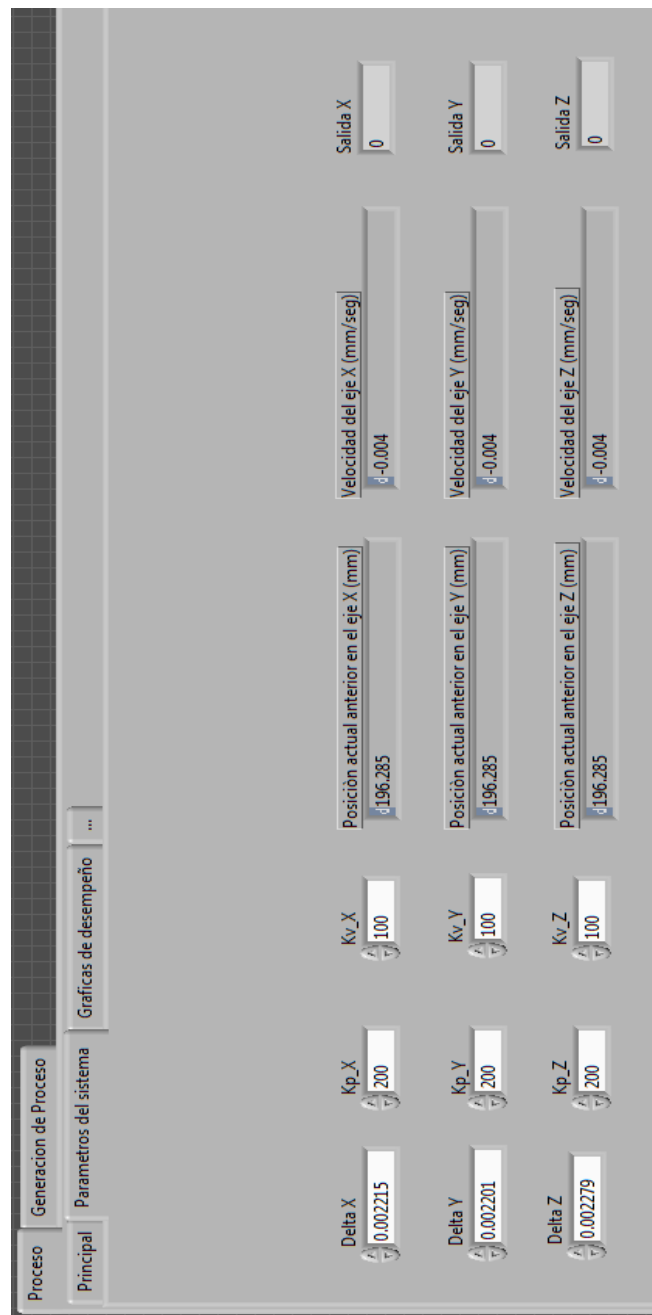


Figura 3. 1: Pantalla de los parámetros del sistema

En la pantalla de la figura anterior, se observa el entorno gráfico de labview, se ingresan datos necesarios para el sistema, como es la resolución teórica del sistema de cada grado de libertad en los recuadros que dicen “Delta”, así como los valores de las ganancias que necesitan el controlador para cada grado de libertad, como es la ganancia proporcional y la ganancia derivativa del sistema (K_p y K_v).

Así como podemos ingresar valores importantes para el sistema en esta ventana también podemos visualizar algunos datos generados por el sistema como es la posición actual anterior de cada eje del sistema, así como la visualización de las velocidades entregadas por el robot cartesiano, todos estos datos son importantes para el funcionamiento del sistema.

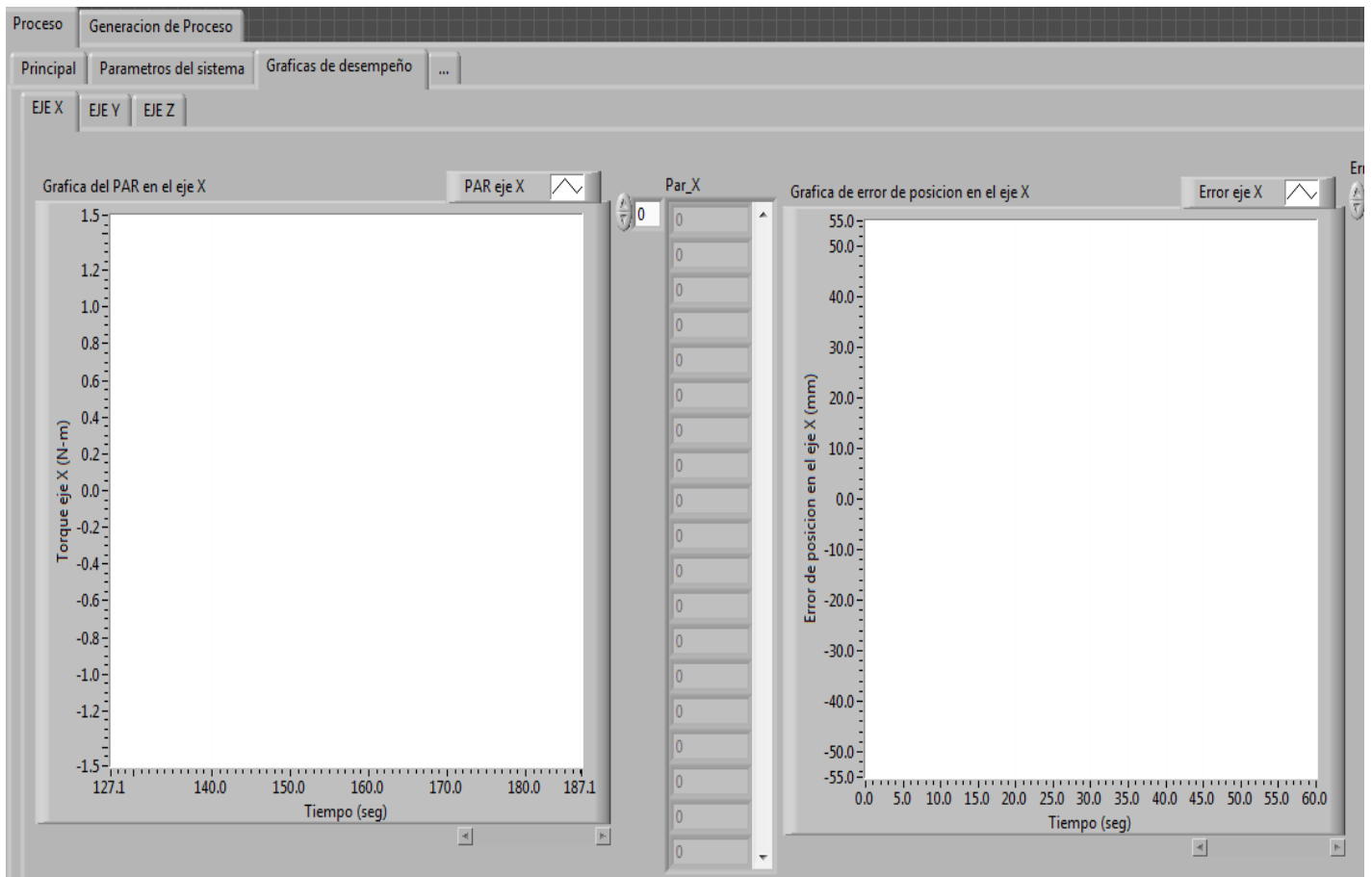


Figura 3. 3: Pantalla de gráficas de desempeño.

En esta pantalla se grafican los pares aplicados y los errores de posición de cada grado de libertad, existen sub-pestañas para poder observar los diferentes grados de libertad, estas graficas se imprimen al final de la ejecución del programa cuando se le da clic al botón de salir de la secuencia de control, solo así se puede visualizar estas gráficas.



Figura 3. 4: Pantalla de lectura de datos del controlador

En esta última pantalla se pueden visualizar los últimos parámetros del sistema, algunos de estos parámetros dependen directamente del controlador como son los cambios de giro, la cantidad de ancho de pulso que depende directamente del valor que obtiene tau del controlador, también existen parámetros de visualización para comprobar que el sistema está funcionando correctamente como son los la lecturas entregadas por el encoder de los motores y la visualización de los bits entregados a cada grado de libertad.

Diseño de trayectorias.

Para el diseño del software del control del CNC se desarrolló un programa que fuera capaz de proporcionar los puntos de trayectoria del robot por medio de funciones convencionales tal y como son constantes, rectas y círculos a partir de esas figuras se puede dibujar cualquier tipo de forma que se requiera como letras y dibujos en alto relieve, una vez que se capturan los puntos que se necesitan para poder realizar la figura estos puntos pueden ser almacenados en un archivo de texto para utilizarlos posteriormente.

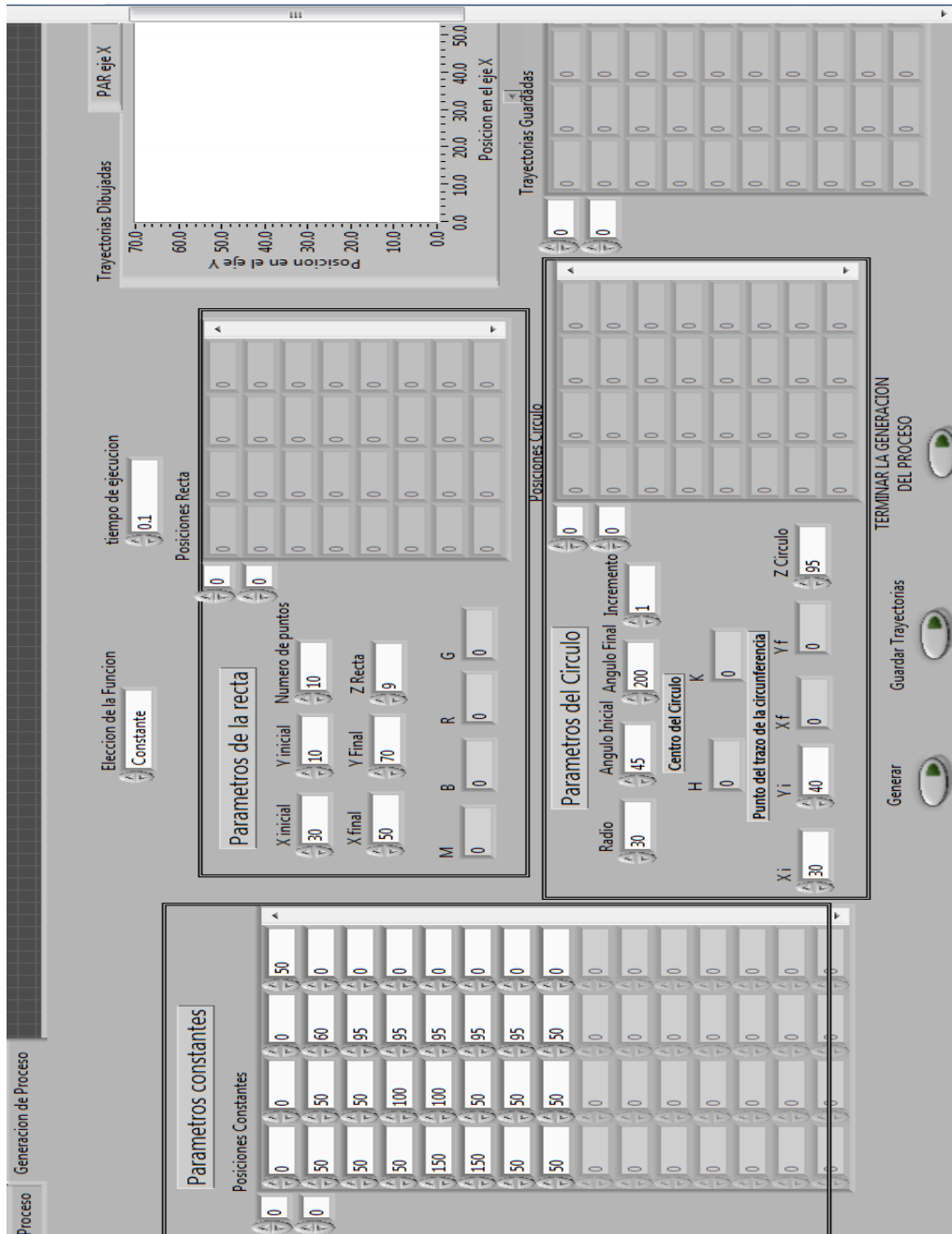


Figura 3.5: Pantalla principal para la creación de trayectorias.

Como se puede observar en la pantalla anterior hay una herramienta para elegir que función se quiere generar, ya sea constantes, círculos o rectas, con estas tres opciones se puede dibujar cualquier tipo de figura que se requiera, utilizar cada una de estas opciones es sumamente sencillo en la elección de una línea solo requiere proporcionar la posición inicial, la posición final y la cantidad de puntos que se requiere entre el inicio y el final, para el dibujo de arcos es la misma metodología el programa pide un punto de inicio un ángulo inicial y un ángulo final con la cantidad de puntos que se requieren entre ángulo y ángulo, mientras más puntos tenga una función más definición tendrá la figura a dibujar.

En la parte de las coordenadas constantes sirve para la orientación de la herramienta si se desea un punto específico ya conocido, con esta función se puede continuar la creación de la figura que se desea desde el punto final de otra función. [14]

3.2 Conclusiones

En este capítulo, se habló del software de control haciendo una breve descripción del mismo, debido que ya se contaba con él, y solo se utilizó como software base, en donde se anexó el software de control de la herramienta de impresión 3D y se realizaron algunos ajustes necesarios para la aplicación. Sin embargo, es importante resaltar que es una parte fundamental en el trabajo de tesis ya que, gracias a este programa se realiza el desplazamiento del manipulador de acuerdo a la posición deseada.

Capítulo 4. Instrumentación de la herramienta de impresión

En una definición amplia el proceso de extrusión hace referencia a cualquier operación de transformación en la que un material fundido es forzado a atravesar una boquilla para producir un artículo de sección transversal constante y, en principio, longitud indefinida. Además de los plásticos, muchos otros materiales se procesan mediante extrusión, como los metales, cerámicas o alimentos, obteniéndose productos muy variados como son marcos de ventanas de aluminio o PVC, tuberías, pastas alimenticias, entre otras cosas. Desde el punto de vista de los plásticos, la extrusión es claramente uno de los procesos más importantes de transformación.

Materiales para impresión 3D mediante extrusión.

ABS: El Acrilonitrilo Butadieno Estireno o ABS es un plástico muy resistente al impacto, utilizado especialmente en la industria automotriz, pero en ocasiones también en la doméstica.

El ABS es un polímero compuesto por tres bloques, *acrilonitrilo*, *butadieno* y *estireno* por lo que se le denomina *terpolímero*. Cada uno de los tres bloques aporta características distintas. El *acrilonitrilo* rigidez, resistencia a ataques químicos, dureza y estabilidad a las altas temperaturas. El *butadieno*, tenacidad a la temperatura cuando ésta es especialmente baja y resistencia a impacto; y el *estireno*, resistencia mecánica, rigidez, brillo, dureza.

Tiene un punto de fusión alto, se puede utilizar para fabricar contenedores de líquidos calientes, hay que extruirlo a unos 230-260 grados y hay que imprimirlo en impresoras con base de impresión caliente.

Al llegar al punto de fusión el ABS desprende gases que en concentraciones altas pueden ser nocivos. Se puede utilizar sin problemas en casa o en la oficina, pero para evitar las concentraciones altas no se recomienda tener varias impresoras funcionando en un espacio pequeño y sin ventilar.

Este material es indicado cuando se busca la fuerza en el resultado o cuando lo que se va a imprimir va a quedar expuesto a temperaturas extremas. Además, se puede mecanizar, pulir, lijar, limar, agujerear, pintar, pegar etc. con extrema facilidad, y el acabado sigue siendo bueno. Está disponible en un abanico más amplio de colores sobre todo por la buena tolerancia de éste a la pigmentación de sí mismo, aunque su precio es más elevado que el PLA.

PLA: El PLA o ácido poliláctico o poliácido láctico es un polímero constituido por moléculas de ácido láctico, con propiedades semejantes a la del Politereftalato de Etileno (PET, Tereftalato de polietileno) que se utiliza para hacer envases, pero además es biodegradable.

Se degrada fácilmente en agua y óxido de carbono. Es un termoplástico que se obtiene a partir de almidón de maíz.

Se utiliza ampliamente en la impresión 3D gracias a sus propiedades físicas y mecánicas que lo hacen muy útil en este campo.

Su tamaño es bastante estable durante el proceso de calentamiento y enfriamiento, lo que hace posible imprimir piezas sin apenas sufrir 'warping', es decir que se doble y se levante por los extremos debido a contracciones durante el proceso de enfriamiento.

Hay que tener en cuenta su mala relación con el agua, por lo que las piezas creadas con PLA no pueden estar expuestas a condiciones de alta humedad por que se deshacen, aunque de forma lenta.

Su punto de fusión se alcanza sobre los 150 °C, en las impresoras 3D se suele calentar el hot-end a unos 210 °C, dependiendo del modelo de impresora 3D. [15]



Figura 4.1: Rollos de filamento PLA Y ABS.

4.1 Extrusor de material

Para la realización de la impresora 3D, son necesarios algunos aditamentos, los cuales se acoplarán a nuestro robot CNC.

Características de Metal J-Head:

- Material: Acero inoxidable, PVC, Nylon
- Voltaje: 12V
- Potencia: 30w
- Longitud del cable: 1 m
- Tamaño del calentador: 6 x 15mm (D * L)
Tubo de PTFE
- Longitud: 1 metro
- Diámetro exterior: 6 mm
- Diámetro interior: 4 mm
- Material: teflón
- Color: blanco

- Transductor
- Termistor de 100K NTC
- Resistencia de cartucho

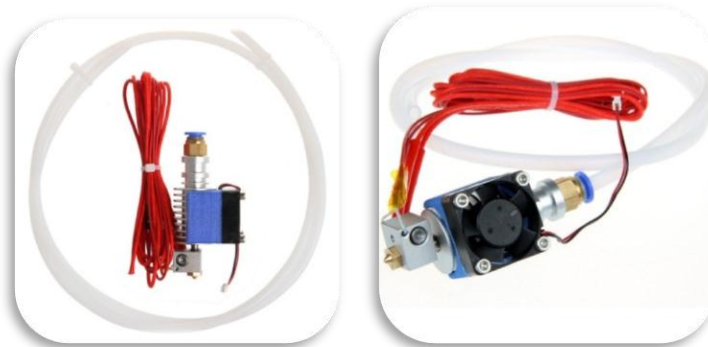


Figura 4.2: Extrusor Metal J-Head de corta distancia y tubo de PTFE (teflón)

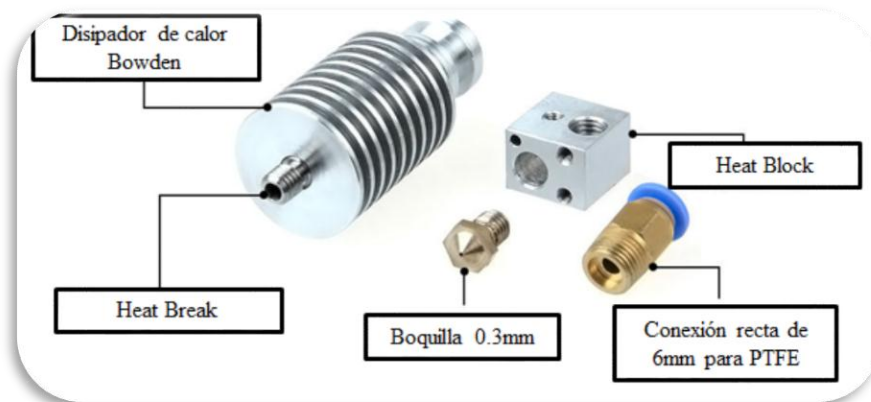


Figura 4.3: Partes del extrusor J-Head

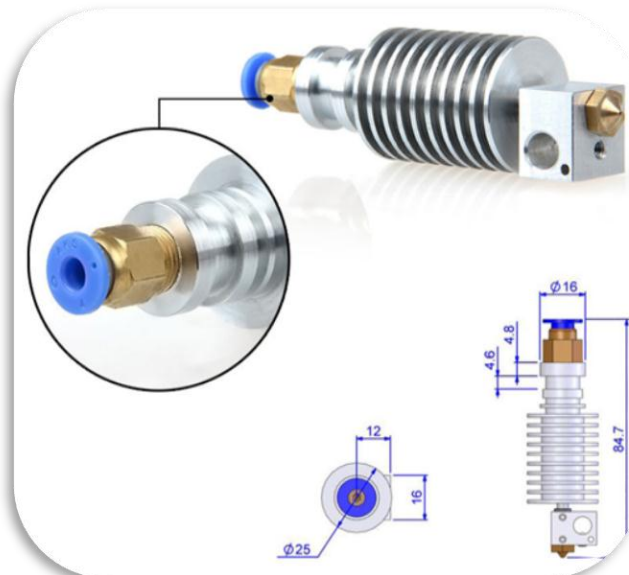


Figura 4.4: Dimensiones de J-Head, medidas en milímetros.

El bowden permite reducir la masa en movimiento de la extrusora y permitiendo un movimiento controlado más rápido, menos sacudidas de la máquina, un menor uso de energía y lo más importante: una impresión más rápida. Normalmente, el mecanismo que impulsa el filamento en el extremo caliente (donde se funde) es directamente en la parte superior de la extrusora. Esto crea problemas de equilibrio y oscilaciones con movimiento más rápido que se puede ver en los resultados de impresión.

Si el mecanismo de accionamiento de filamentos se coloca en una parte no móvil de la impresora 3D, puede ser empujado en un tubo. PTFE (teflón) es útil porque es resbaladizo: tiene poca fricción con los plásticos. Estos límites de desgaste y pérdida de energía. Otro extremo del tubo se conecta a hot- final de la extrusora.

El concepto de usar un cable Bowden como guía fue sugerido y promovido por Ed Sells.

El diámetro interior de la boca de alimentación de Bowden es de 4 mm para imprimir con filamento de 1,75 mm, y 6 mm para imprimir con filamento de 3 mm.

Una de las principales ventajas que presenta este J-Head, es el material del cual está hecho (acero inoxidable), ya que le permite trabajar con policarbonato de hasta 300 grados centígrados y con materiales futuros que requieren temperaturas más altas de trabajo más allá de los admitidos por extrusoras de plástico y extremo caliente. Probado con PLA, ABS, policarbonato, nylon y PVA (materiales utilizados para la impresión 3D). [16]

Termistor.

Los termistores, como RTDs, son semiconductores térmicamente sensibles cuya resistencia varía con la temperatura. Cada sensor tiene una resistencia nominal designada que varía proporcionalmente con la temperatura de acuerdo a una aproximación alineada. Los termistores tienen ya sea un coeficiente de temperatura negativo (NTC) o un coeficiente de temperatura positivo (PTC). El primero, y el más común, tienen una resistencia que disminuye al aumentar la temperatura y el segundo presenta mayor resistencia al aumentar la temperatura.

Los termistores generalmente tienen una sensibilidad muy alta ($\sim 200 \Omega/^{\circ}\text{C}$), lo cual los hace extremadamente susceptibles a los cambios de temperatura.

Sistema de conversión analógica – digital y digital - analógica.

Se requirió de una etapa de conversión digital – analógica debido a que utilizamos termistores para el sensado de temperatura, este sensor varía su resistencia con respecto a la temperatura actual, como se mencionó anteriormente, y se emplea tanto en la herramienta de extrusión como en la base caliente, por lo que tuvimos que adecuar la señal. Para esto se propuso un

divisor de voltaje, el cual nos permitió tener una variación de voltaje en lugar de una variación de resistencia siendo más fácil de manipular de manera digital.

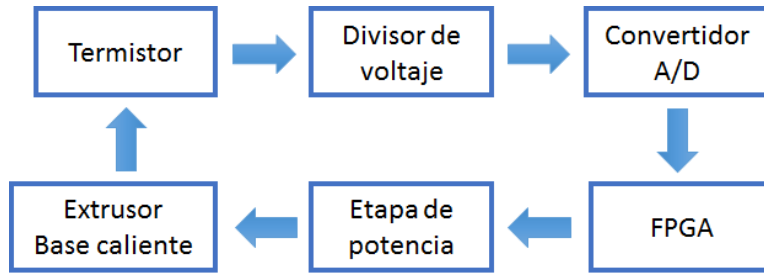


Figura 4.6: Diagrama esquemático de la conversión a/d de la señal de nuestros termistores.

Esta señal se manda a un convertidor analógico – digital, un ADC0804, puesto que 8 bits son suficientes para nuestra aplicación. Después de la digitalización, esta señal se manda por el bus PCI para poder ser leída por el FPGA, en donde se realiza el procesamiento de señal para posteriormente realizar el control dentro del firmware y mandar una señal de control a nuestra etapa de potencia. Este ciclo se repite durante todo el proceso de impresión, teniendo un periodo de precalentado antes de iniciar la impresión 3D.

Resistencia eléctrica tipo cartucho.

Para poder controlar la resistencia del extrusor mediante una señal digital, se realizó la caracterización de misma utilizando la siguiente metodología:

1. Se realizó un circuito para poder controlar la resistencia del extrusor mediante una señal de PWM a una frecuencia de trabajo de 100Hz, con una amplitud de 3.3V pico a pico. Para ello se implementaron transistores en configuración Darlington como se muestra a continuación:

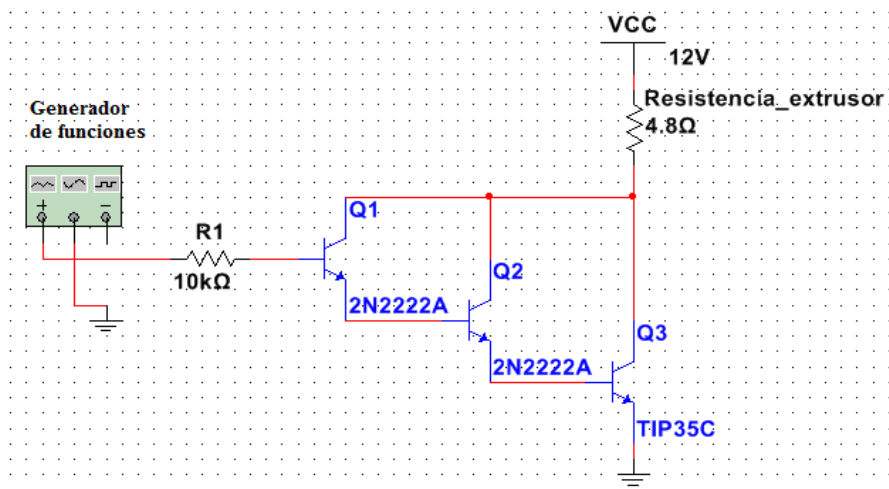


Figura.4.8: Circuito Darlington ocupado para la caracterización de la resistencia del extrusor.

2. Se conectó un sensor de temperatura y un display ajeno al termistor del extrusor para poder visualizar la temperatura actual del dispositivo.
3. Posteriormente se fue variando el ciclo de trabajo de la señal de PWM en un 10%, desde 5% hasta 100%, y después de cinco minutos de funcionamiento se tomó la lectura de la temperatura actual.
4. Finalmente se grafican los resultados obtenidos, para observar el comportamiento de la resistencia del extrusor.

Frecuencia= 100 Hz	Tiempo de prueba= 5 min. por intervalo
Temperatura inicial= 30°C para cada intervalo	
% Ciclo de trabajo	Temperatura °C
5%	32°
10%	43°
20%	63°
30%	82°
40%	104°
50%	123°
60%	143°
70%	160°
80%	174°
90%	191°
95%	200°
100%	207°

Tabla 9: Caracterización de la resistencia.

En base a los resultados obtenidos, encontramos su curva de ajuste mediante Labview, herramienta que nos proporciona la ecuación de ajuste que utilizamos para nuestro controlador.

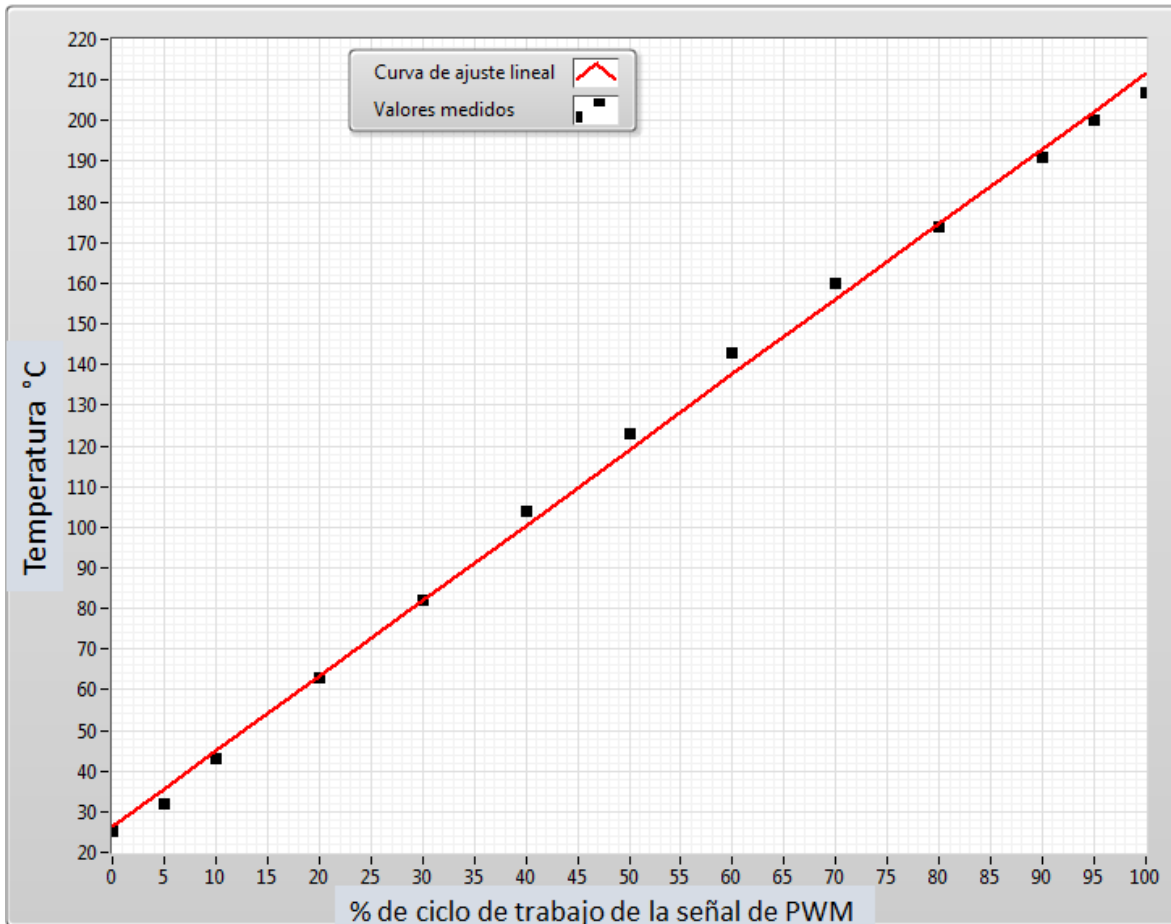


Figura 4.10: Gráfica temperatura vs % ciclo de trabajo (Frecuencia 100 Hz) y su curva de ajuste lineal.

La ecuación de ajuste será:

$$y = x * m + b$$

dónde:

$m = 1.8512$, es la pendiente.

$b = 26.4419$, es la ordenada al origen.

x representa el porcentaje de ciclo de trabajo de la señal de PWM.

y representa la temperatura del extrusor.

De la gráfica anterior podemos concluir que nuestra herramienta impresión, el extrusor, tiene un comportamiento muy cercano al lineal por lo que nuestro lazo de control tendrá buenos resultados al aplicarse.

4.1.1 Firmware de control de temperatura del extrusor de material.

En el firmware se realiza el control de temperatura del extrusor y de la base caliente, empleando un control proporcional, como se muestra a continuación.

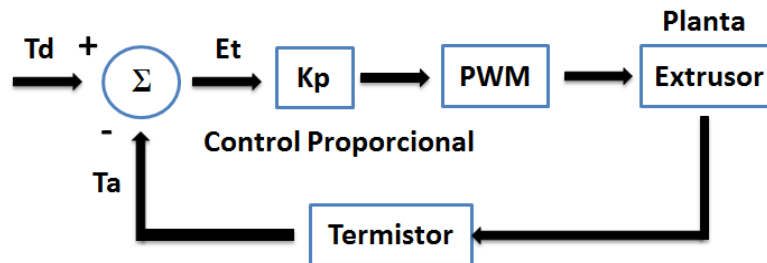


Figura 4.11: Diagrama a bloques del lazo cerrado de control del extrusor.

En donde, T_d es la temperatura deseada, K_p representa el control proporcional, E_t es el error de temperatura y T_a la temperatura actual.

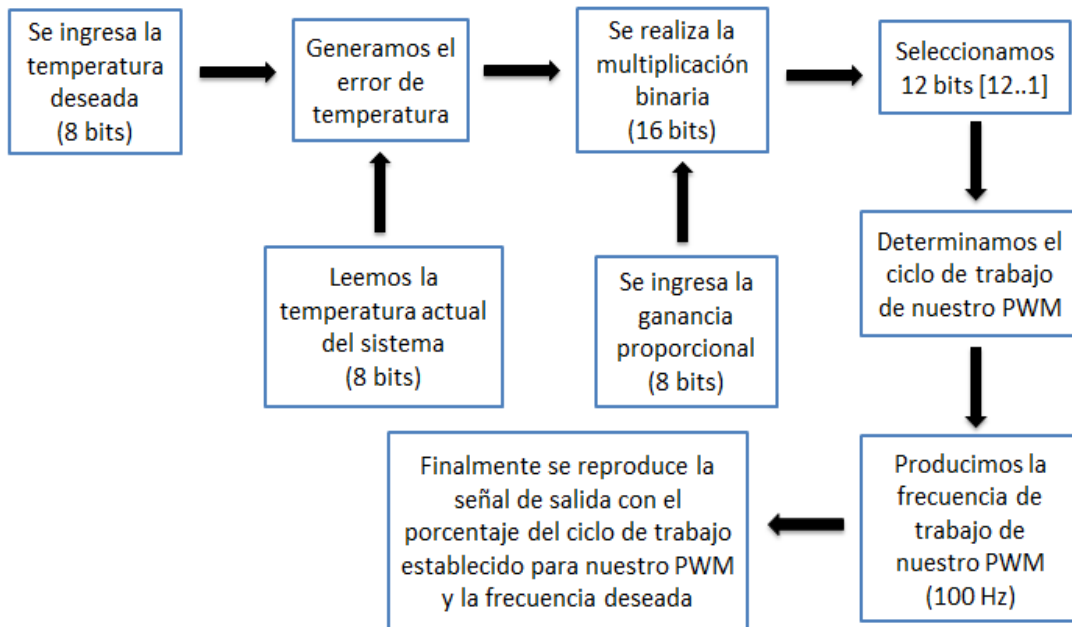


Figura 4.12: Diagrama a bloques del funcionamiento del firmware del FPGA

Utilizamos el software del sistema para que el usuario pueda indicar la temperatura deseada, que posteriormente será enviada a través del puerto PCI en un dato de 8 bits sin signo. La temperatura actual sensada por el termistor es enviada a un ADC de 8 bits, el cual envía el dato directo al FPGA, sin embargo, para que el usuario pueda visualizar esta información se manda esta información al puerto PCI, donde el software recibe el dato y lo decodifica para mostrarlo en la pantalla principal.

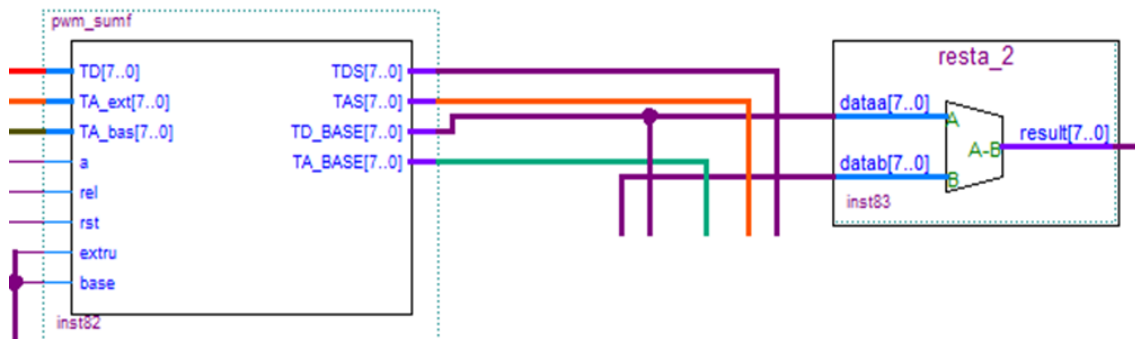


Figura 4.13: Etapa 1. Lectura de temperatura deseada y actual. Para la generación del error de temperatura.

Una vez que tenemos los datos de T_d y T_a lo que necesitamos realizar una resta para determinar el error de temperatura E_t , lo cual conseguimos empleando una resta de 8 bits sin signo.

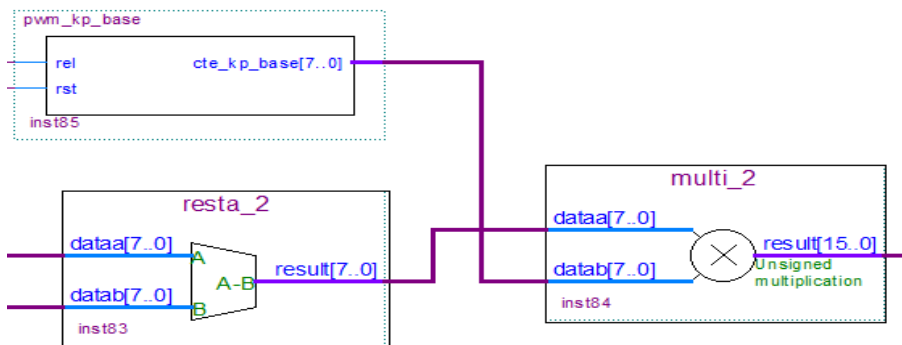


Figura 4.14: Se multiplica la constante proporcional por el error de temperatura.

Cuando ya tenemos el valor de E_t se multiplica por K_p , la cual es una constante previamente calculada y programada en el firmware. Esta multiplicación nos da como resultado un valor de 16 bits. Sin embargo, nosotros empleamos una señal de 8 bits para crear la señal de PWM que posteriormente pasa a la etapa de potencia.

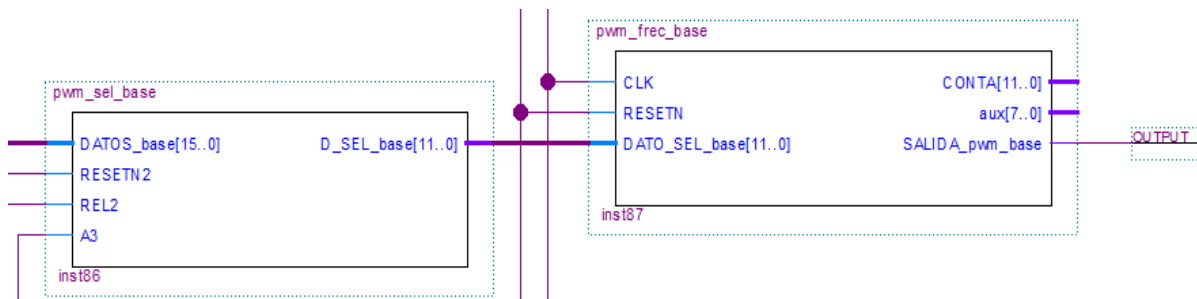


Figura 4.15: Etapa 3. Se seleccionan los 8 bits para determinar el ancho de pulso de la señal y se genera la señal de salida.

Por lo que, en el firmware de control de temperatura se realizó un bloque en donde se puedan seleccionar los 8 bits intermedios, para generar la señal de PWM a una frecuencia de 100 Hz, que es la frecuencia de trabajo seleccionada para nuestro extrusor.

4.2 Base caliente.

La base caliente que implementamos fue la modelo MK3, esta base caliente cuenta con 3 mm de núcleo de aluminio, que le permite imprimir directamente sobre ella, sin necesidad de una placa de vidrio.

Tiene doble fuente de alimentación, es decir operar a 12V o 24V. Cuando está conectada a 24V se pueden utilizar hilos más finos. [17]

Sus características son:

- Dimensiones 214mm x 214mm
- 1 capa 35µm (base 1 oz) de cobre
- Resistencia entre 1,4 y 1,6 ohmios para el 12V
- Resistencia entre 5,0 y 5,4 ohmios para el 24V
- 110 grados Celsius posible para 12V y 24V
- Ejecución de 24 V en los terminales de 12V será el Modo de calentamiento Heatbed a 100 grados en sólo 2 minutos
- 180 grados centígrados de temperatura máxima
- Planitud aprox. dentro + -0.2mm

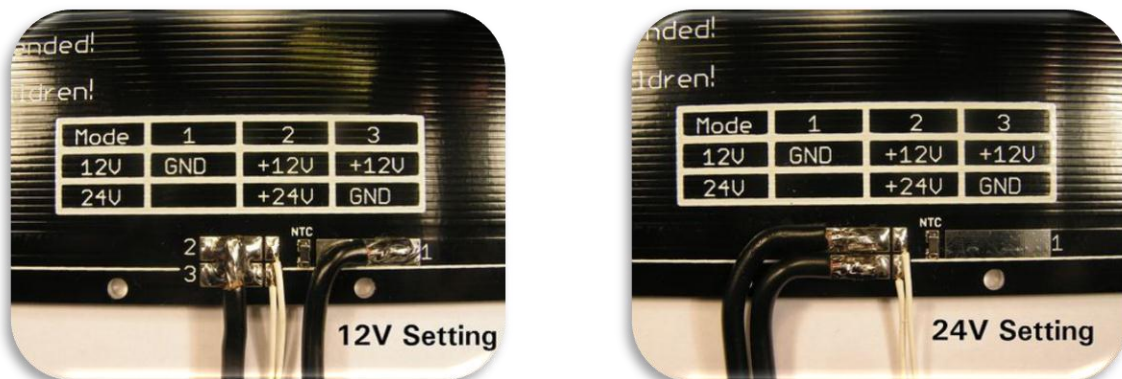


Figura 4.16.- Configuraciones de alimentación de la base caliente.

De manera análoga al extrusor de material, la base caliente emplea un control proporcional para regular la temperatura de trabajo. El lazo cerrado de control será el siguiente:

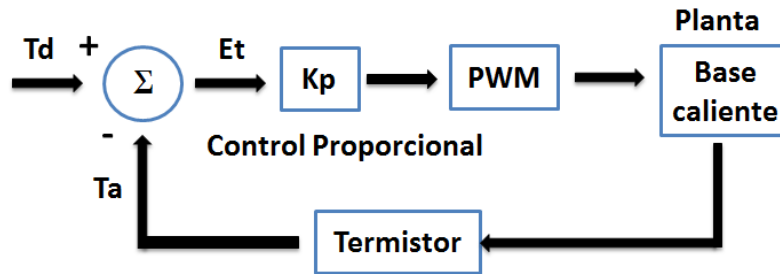


Figura 4.17: Lazo cerrado de control para la base caliente.

dónde:

T_d , es la temperatura deseada.

T_a , es la temperatura actual de la base caliente.

E_t , es el error de temperatura.

K_p , representa el control proporcional, en donde se puede modificar la ganancia proporcional.

El firmware se realizó de manera análoga al del extrusor, teniendo las mismas etapas de control con pequeñas variaciones, como el valor de la ganancia proporcional y la selección de los 8 bits para generar la señal de PWM, en este caso se utilizaron los 8 bits menos significativos.

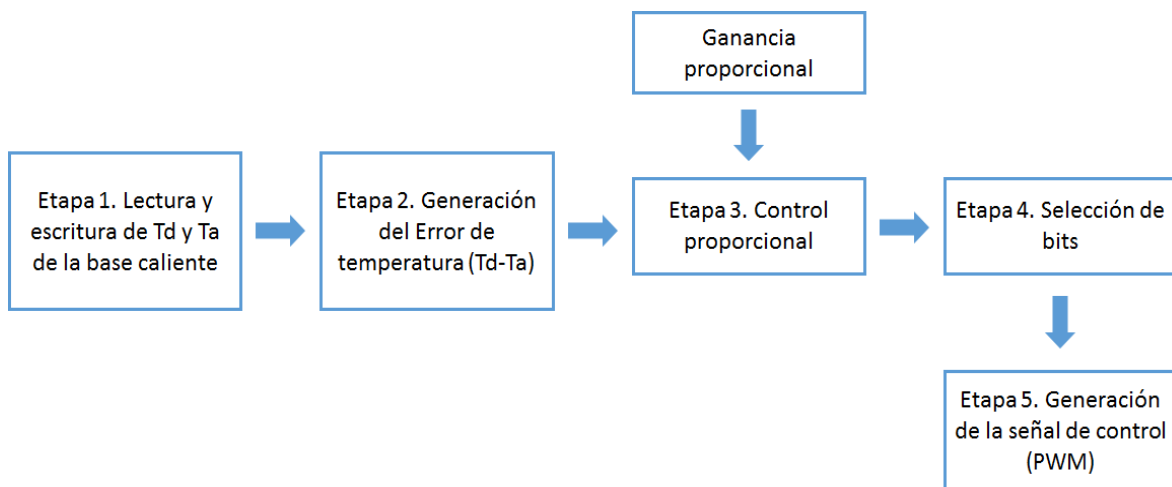


Figura 4.18: Diagrama bloques del firmware de control de temperatura.

Cabe resaltar que, en la primera etapa, se utiliza el puerto PCI, para enviar y recibir el dato que contiene la información de la temperatura deseada y la temperatura actual. Se interactúa con el software de control, el cual, multiplexea la señal para determinar cuándo se realiza la

lectura de la temperatura deseada del extrusor y cuando se realiza la lectura de la temperatura deseada de la base caliente. Además, se realiza un filtrado digital para después enviar el dato al puerto PCI para que pueda ser leído por el software de control, y el usuario lo pueda visualizar.

Para la segunda y la tercera etapa, se utilizan las megas funciones programadas en Quartus II, software empleado para realizar firmware de control del sistema. Con respecto a la cuarta etapa, como ya se mencionó se seleccionan los 8 bits menos significativos para enviarlos al siguiente bloque en el firmware, ya en la quinta etapa, se genera la señal de PWM a la frecuencia deseada en este caso también será de 100Hz.

4.3 Alimentador de material

Para la impresión 3D es necesario un alimentador, en nuestro caso es un motor a pasos engranado con cambio rápido de filamento, modelo EXTV22SMBU, que cuenta con una relación de transmisión 14:1, es decir por cada 14 vueltas del motor tendrá una vuelta a la salida. El motor es de la marca Kysan modelo 1040220. Una de las características principales de este dispositivo es que cuenta con un alto par de salida del motor con un consumo de energía menor y una buena sujeción para el agarre del filamento, debido a su engranaje interno de precisión.

Otra característica importante es el cambio rápido de filamento, el cual se reduce a girar dos veces el tornillo de sujeción de la polea de agarre, insertar el nuevo filamento y apretar. Reduciendo considerablemente el tiempo de la operación.

En la parte superior, se puede conectar un tubo (PTFE) guía para orientar el filamento a la extrusora, evitando que el filamento se enrede con otros mecanismos. [18]

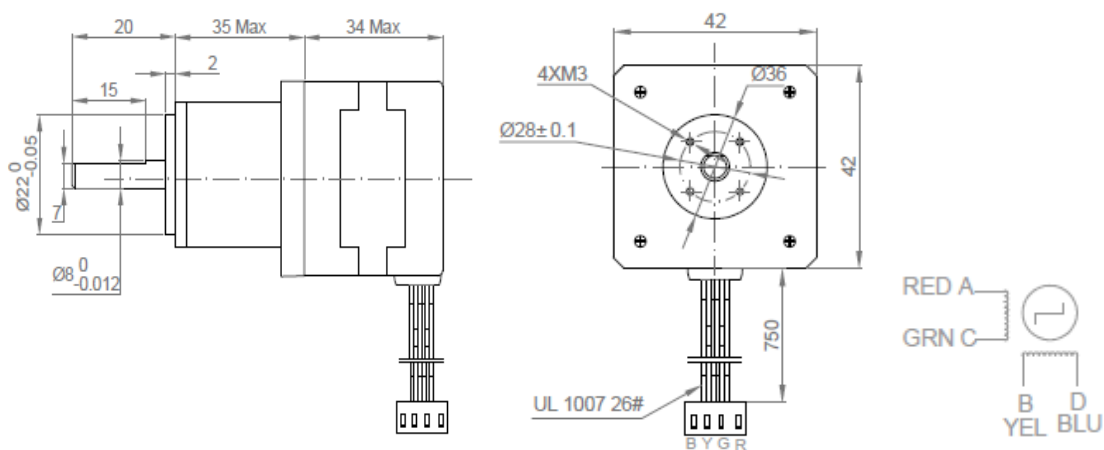


Figura 4.19: Dimensiones del motor a pasos en mm y configuración de bobinas (motor a pasos bipolar)

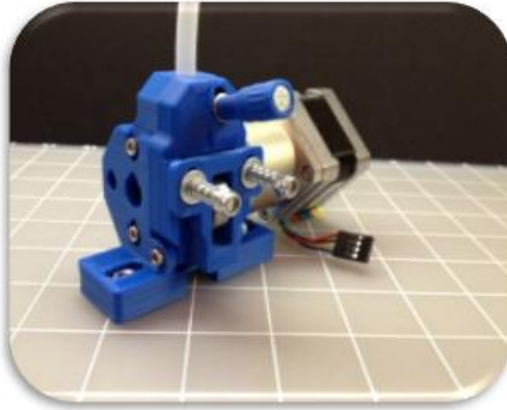


Figura 4.20: Motor a pasos engranado con cambio rápido de filamento.

Artículo	Especificación
Angulo de paso	$1.8^\circ \pm 5\%$
Voltaje Nominal	2.6V
Corriente nominal	1A
Resistencia por fase	$2.6\Omega \pm 10\%$
Numero de fases	2
Relación	14:01
Torque	24 kg*cm

Tabla 10: Características del motor a pasos Kysan

4.3.1 Firmare del control de velocidad del alimentador de material

Durante la impresión 3D es importante tener control de la velocidad de alimentación del material a extruir, en este caso, esta velocidad de alimentación es controlada por la frecuencia de la secuencia de pulsos que se manda a la etapa de potencia del alimentador. Debido a que es un lazo abierto de control no se cuenta con retroalimentación.

Para le secuencia de control se utilizó la siguiente secuencia de activación de bobinas:

Secuencia	Terminales			
	A	B	C	D
1	VCC	GND	VCC	GND
2	VCC	GND	GND	VCC
3	GND	VCC	GND	VCC
4	GND	VCC	VCC	GND

Tabla 11.- Secuencia de activación de motor a pasos bipolar.

Como se describe en el siguiente diagrama de flujo se utilizaron condicionales IF, y un contador para ir coordinando la secuencia de manera análoga a una máquina de estados. Para

nuestro alimentador no es necesario contar con una secuencia reversible lo que facilita la implementación de firmware.

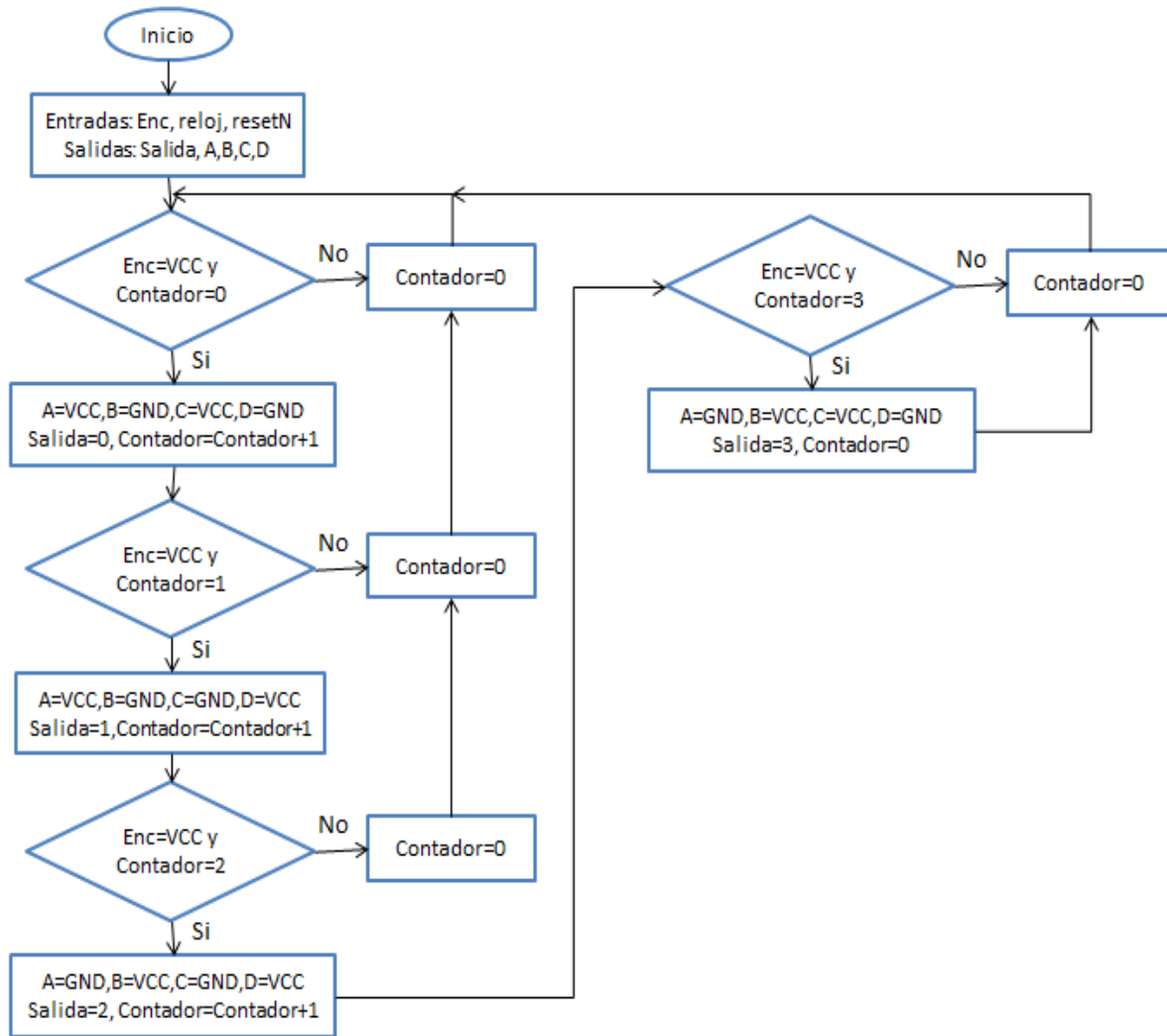


Figura 4.21: Diagrama de flujo del firmware para control del motor a pasos bipolar

Sin embargo, para esta secuencia de control necesitamos una frecuencia mucho menor a la que maneja el FPGA, por lo que se utilizó un divisor de frecuencia como el empleado en el caso anterior para el PWM.

Esta frecuencia deseada se genera a partir del reloj del FPGA por medio de la siguiente fórmula:

$$Num. de pulsos = \left(\frac{1}{Frecuencia\ deseada} \right) (Frecuencia\ del\ FPGA)$$

$$\left(\frac{1}{Valor\ máx.\ decimal\ de\ 8\ bits} \right) \tag{52}$$

Este número de pulsos en nuestro divisor de frecuencia servirá para dar un conteo. De la formula anterior obtenemos:

$$\left(\frac{1}{100\text{Hz}}\right) (33\text{MHz}) \left(\frac{1}{255}\right) \approx 1294 \text{ pulsos de reloj del FPGA} \quad (53)$$

En el bloque del divisor de frecuencia la señal de salida se mantendrá en alto durante 647 pulsos, que es la mitad de los pulsos necesarios para generar la frecuencia deseada, y durante los otros 647 pulso la señal se mantendrá en bajo, así tendremos una señal de salida con una frecuencia de 100 Hz. Sin embargo, nuestro programa cuenta con la versatilidad de poder seleccionar tres frecuencias de alimentación, en este caso, serán 100 Hz, 300Hz y 500Hz, las dos frecuencias anteriores se determinaron empleando la misma metodología utilizada para el divisor de frecuencia de 100Hz.

Por lo tanto, nuestro firmware quedo de la siguiente manera:

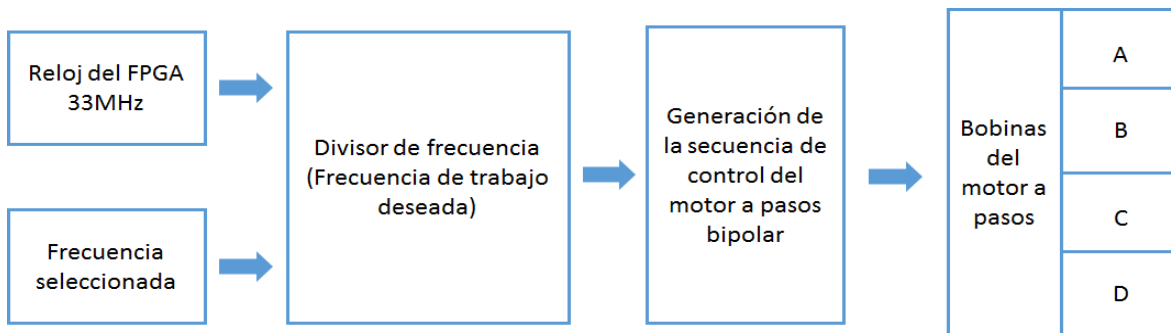


Figura 4.22: Diagrama a bloques del firmware de control de velocidad del alimentador.

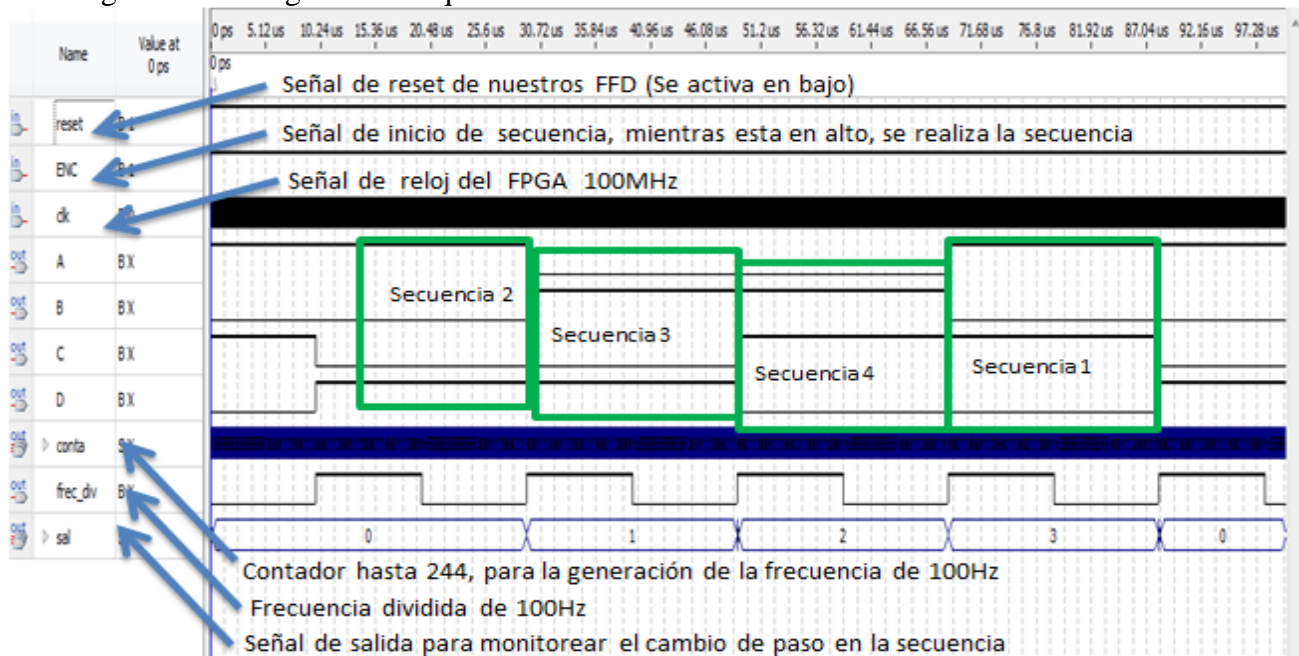


Figura 4.23: Diagrama de tiempos de la simulación del firmware.

dónde:

La señal de Enc es nuestro habilitador de inicio de secuencia.

Clk, es el reloj del FPGA 100Mhz

A, B, C y D son nuestras señales de salida, las cuales habilitan las bobinas de nuestro motor.

Conta, representa el contador interno para la generación de la frecuencia de trabajo deseada

Frec. div, simboliza la frecuencia de trabajo de nuestro motor (en este caso, utilizamos 100Hz).

Sal, es una señal que nos sirve de referencia para saber en qué paso de nuestra secuencia nos encontramos.

4.4 Conclusiones

En el capítulo anterior se hace mención del firmware implementado para el control de las herramientas necesarias para la impresión 3D. Para controlar la temperatura tanto del extrusor como de la cama caliente fue necesario un controlador sencillo debido a la respuesta de la planta, sin embargo, es funcional para la aplicación. Con respecto al alimentador de material, tenemos que es un control en lazo abierto, por lo que facilita la realización del firmware de control, pero cuenta con una característica importante para este proceso que es la selección de velocidad de alimentación, que ayudará a la hora de realizar la impresión.

Capítulo 5. Software de la impresora 3D

El software del sistema se realizó en LabView, ya que esta plataforma nos permite manipular fácilmente el puerto PCI, nuestro bus de transmisión y lectura de datos. Mediante este programa el usuario puede visualizar la información adquirida por el FPGA, de nuestros diferentes sensores, como los encoders para el caso de la posición del CNC y los termistores para la temperatura actual del sistema. Además, el usuario también puede modificar los diferentes parámetros del sistema, como la temperatura deseada tanto del extrusor, como de la cama caliente, las posiciones deseadas para cada uno de nuestros ejes y la velocidad del alimentador.

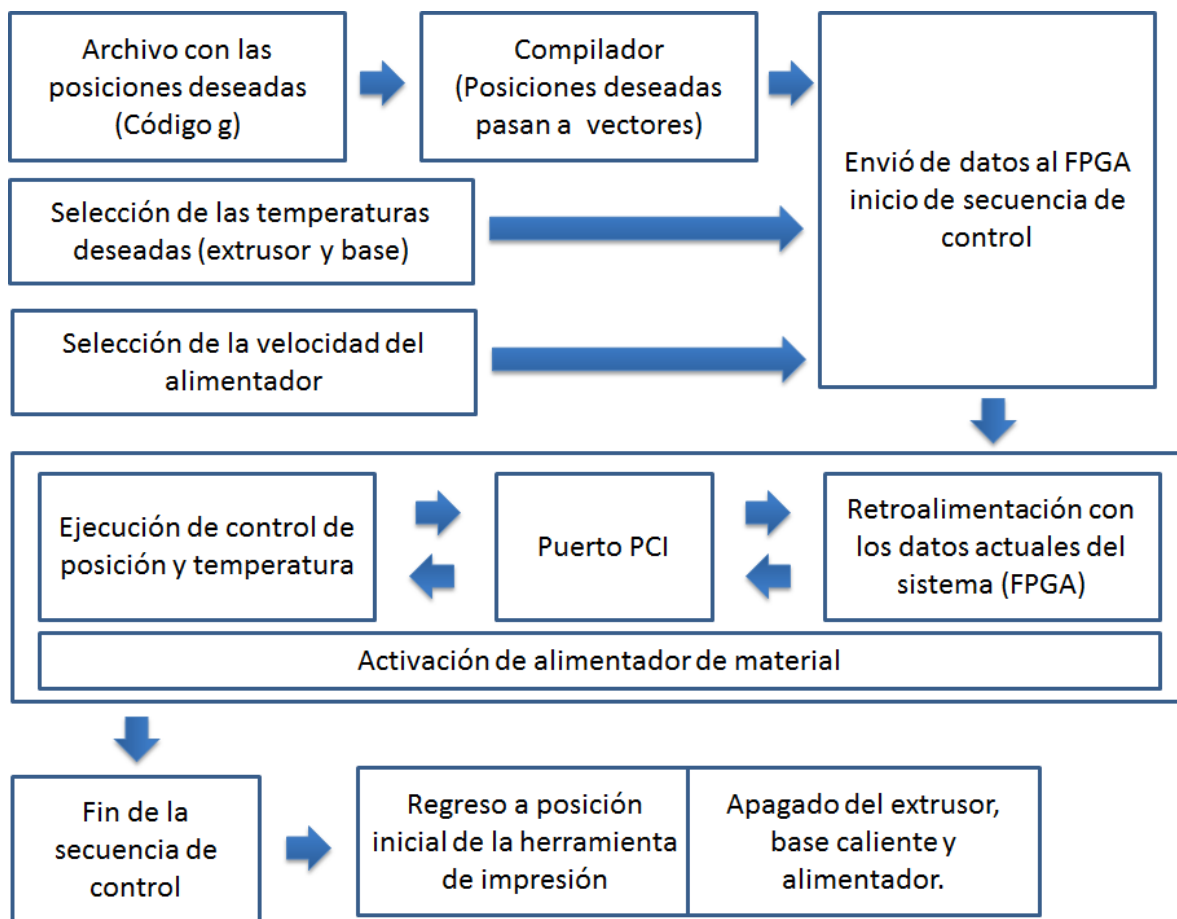


Figura 5.1: Diagrama esquemático del funcionamiento del software de control.

En el diagrama anterior se muestra brevemente el funcionamiento del software de control de la impresora 3D, empezando por un programa en código g, el cual pasa a un compilador para seleccionar las posiciones de cada eje y posteriormente ser ejecutadas por nuestro CNC, en el caso de la sección del software que controla la herramienta de impresión, se divide en dos partes, en la primera parte se indica las temperaturas deseadas para nuestra impresión, tanto

del extrusor como de la cama caliente, estas temperaturas varían acorde al material a imprimir, en nuestro caso será PLA. En la segunda parte se selecciona la velocidad con la cual se alimentará el material a nuestro extrusor, es decir, la velocidad de nuestro motor a pasos. Una vez que ya se definieron estos parámetros, se inicia la secuencia de control, durante la ejecución de la secuencia de control, nuestros algoritmos se mantendrán ejecutándose para mandar constantemente la energía que debe ser aplicada a nuestros actuadores, además, se hace una retroalimentación con el FPGA cada 5 milisegundos, para ir actualizando la información de nuestros sensores. El envío y recepción de información se realiza a través del bus PCI, por lo que constantemente se tiene que ir actualizando la lectura y escritura del puerto. El software se mantendrá enviando la señal de control al alimentador de material mientras la secuencia se encuentre activa. Cuando termina la secuencia de control el software indica al sistema apagar las señales que controlan la herramienta de impresión y el robot se posicionará en su posición de inicio.

A continuación, se presenta la primera parte de la pantalla principal del software de control.

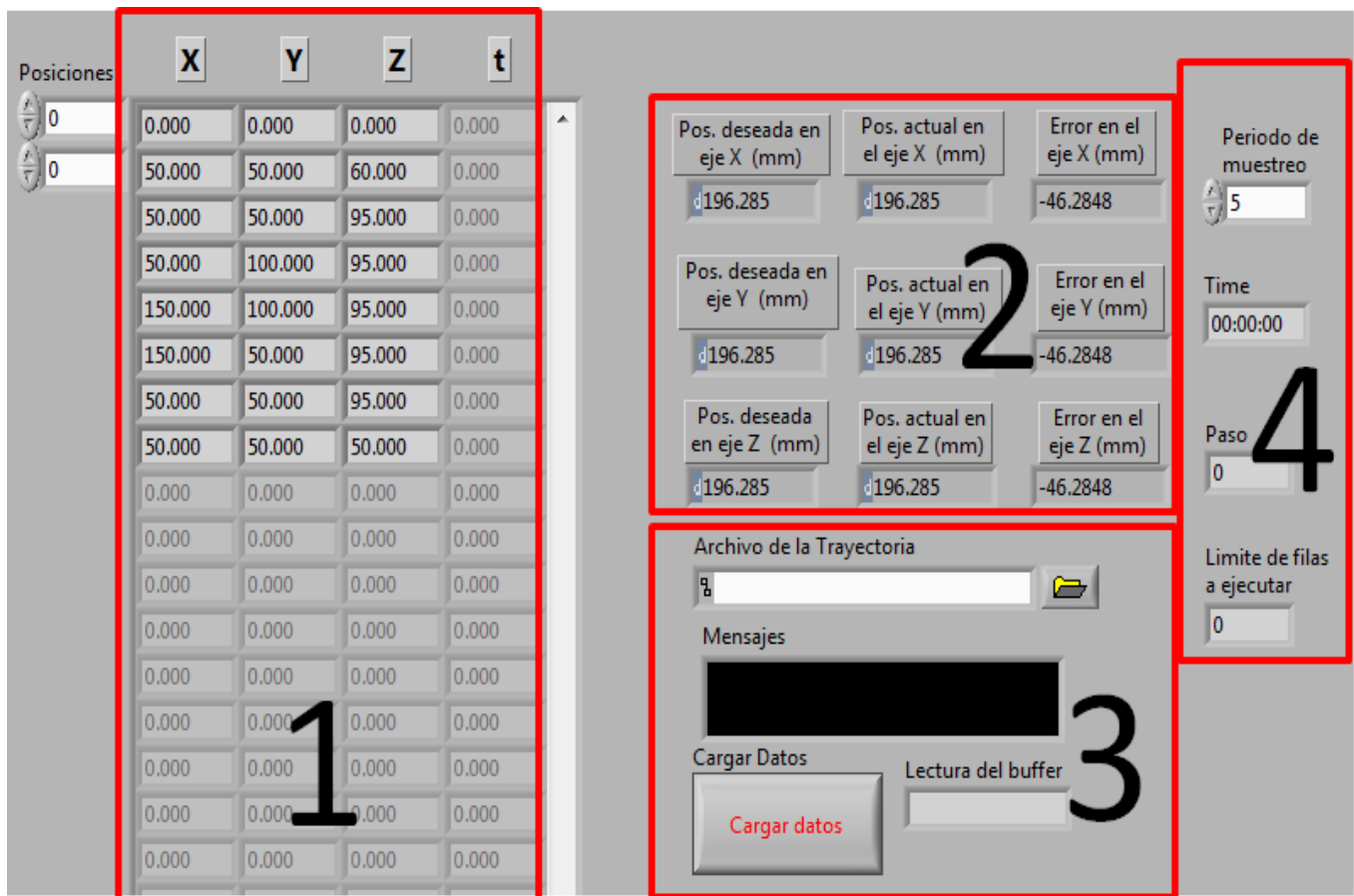


Figura 5.2: Primera parte de la pantalla principal del sistema.

En la primera sección, se muestran las coordenadas deseadas que fueron cargadas desde el archivo en código g.

Posiciones	X	Y	Z	t
0	0.000	0.000	0.000	0.000
0	50.000	50.000	60.000	0.000
	50.000	50.000	95.000	0.000
	50.000	100.000	95.000	0.000
	150.000	100.000	95.000	0.000
	150.000	50.000	95.000	0.000
	50.000	50.000	95.000	0.000
	50.000	50.000	50.000	0.000
	0.000	0.000	0.000	0.000

Figura 5.3: Coordenadas deseadas cargadas en el software de control.

En este caso, se cargan las coordenadas correspondientes a los ejes X, Y y Z, con respecto al vector del t (tiempo), este vector será generado por nuestro compilador, el número de coordenadas está determinado por el archivo en código g.

Pos. deseada en eje X (mm)	Pos. actual en el eje X (mm)	Error en el eje X (mm)
196.285	196.285	-46.2848
Pos. deseada en eje Y (mm)	Pos. actual en el eje Y (mm)	Error en el eje Y (mm)
196.285	196.285	-46.2848
Pos. deseada en eje Z (mm)	Pos. actual en el eje Z (mm)	Error en el eje Z (mm)
196.285	196.285	-46.2848

Figura 5.4: Indicadores de coordenadas.

La sección dos, sirve al usuario para monitorear la posición actual del sistema y compararla con la posición deseada. También permite visualizar el error de posición en cada eje (posición deseada – posición actual), indicándole al usuario como se acerca la herramienta de impresión a la posición deseada.

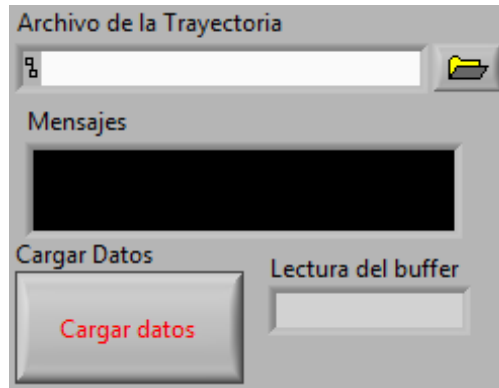


Figura 5.5: Sección para seleccionar la trayectoria del manipulador e inicial la secuencia de control.

En la figura anterior se muestra la sección 3 del programa donde el usuario puede cargar la trayectoria a realizar, cuando se pulso el icono de “Cargar datos”, la secuencia de control comienza a ejecutarse. El usuario podrá visualizar mensajes del estatus del programa en esta sección, en el apartado de “Mensajes”.

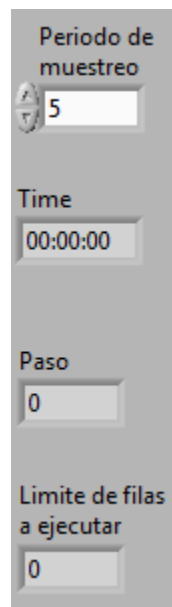


Figura 5.6: Apartado para seleccionar el tiempo de muestreo.

En la sección 4, el usuario puede seleccionar el tiempo de muestro del sistema, este valor estará multiplicada por " 10^{-3} ", es decir, para el caso anterior tendremos 5 milisegundos. Este tiempo de muestreo se carga por default con el valor de 5. Además, se puede observar el tiempo de ejecución del programa en la casilla de “Time” y el número de filas a ejecutar, este número depende de las coordenadas deseadas marcadas el código g.

En la segunda parte de la pantalla principal, tenemos:

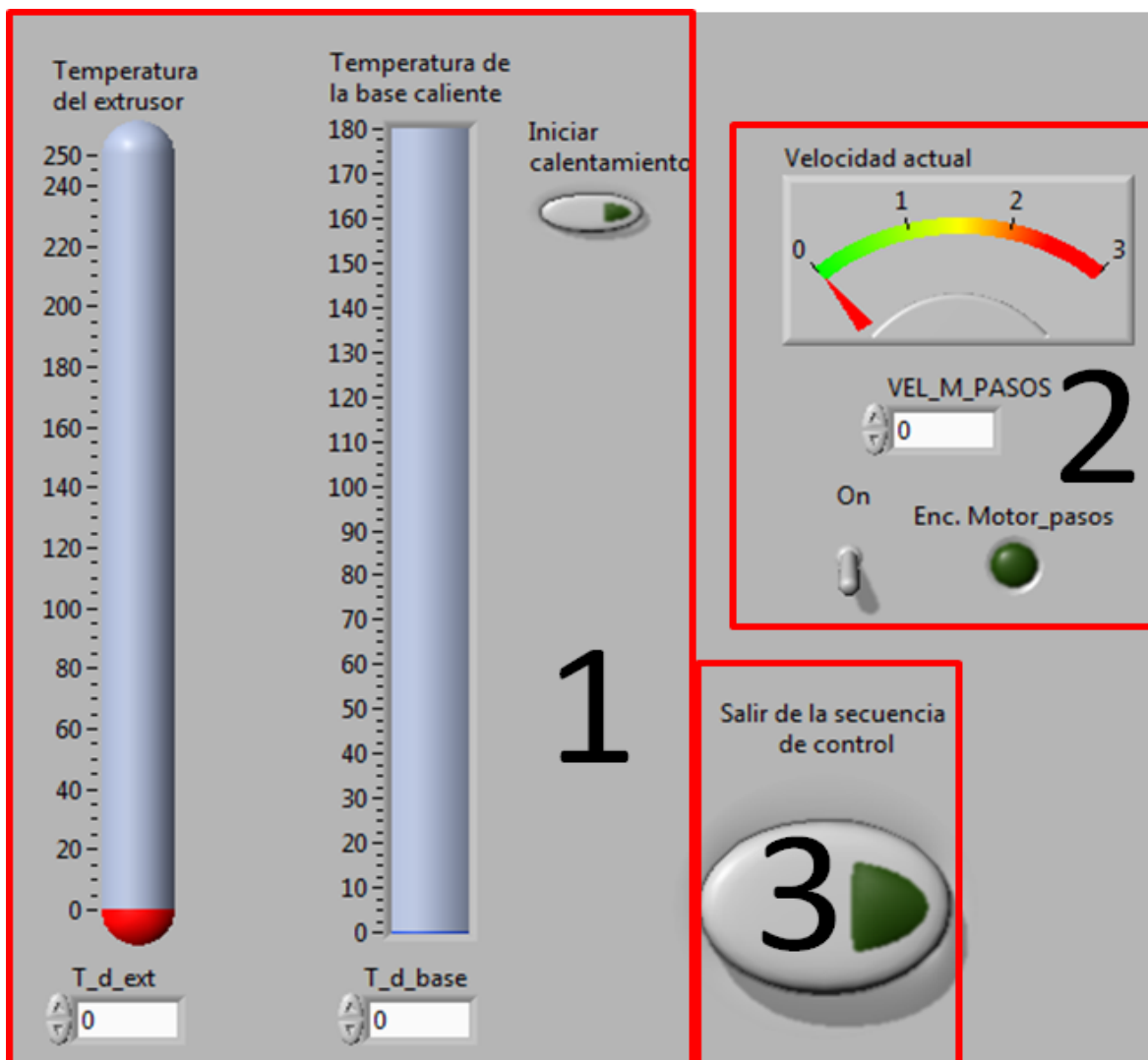


Figura 5.7: Segunda parte de la pantalla principal.

En esta segunda pantalla se muestran tres secciones, en la primera sección se enfoca principalmente a los parámetros de temperatura del extrusor y de la base, en donde se indica la temperatura deseada para cada uno de los elementos, además, se puede visualizar la temperatura actual tanto del extrusor como de la base caliente. Después de haber programado la temperatura de cada uno de estos elementos, se debe de presionar el botón de “inicio de calentamiento”, para que el software permita la lectura y escritura en el puerto PCI de los parámetros de temperatura, de lo contrario, aunque la electrónica se encuentre encendida no se hará ninguna acción.

En la segunda sección tenemos el control de velocidad de nuestro alimentador de material (motor a pasos). Como se puede ver en la figura (5.7), se pueden seleccionar tres tipos de

velocidades de alimentación, esto con el afán de que el usuario al modificar la temperatura de trabajo pueda ajustar la velocidad de alimentación de material y mantener un buen desempeño de la impresora 3D, acorde a sus necesidades. También podemos encontrar un botón con el cual se enciende el alimentador, este botón de accionamiento permite que el usuario puede definir cuándo iniciar o interrumpir el proceso de alimentación.

Finalmente tenemos una tercera sección donde podemos encontrar el botón de salida de la secuencia de control, al presionar este botón podemos salir de la secuencia de control, aunque nos encontremos a mitad del proceso, este botón funciona como botón de paro dentro del software.

5.1 Laminador y compilador.

Actualmente, el proceso a nivel software para realizar una impresión 3D con nuestro robot manipulador consiste en que el usuario realice el diseño de la pieza o prototipo que desea imprimir en un programa de CAD, lo guarde como un archivo con extensión “. stl”, para que posteriormente lo pueda abrir con un programa laminador para generar el código g de la pieza. Este nuevo archivo, lo podemos abrir desde nuestro software de control, el cual hace una depuración del archivo y adquiere todas las coordenadas por eje y las guarda en vectores, para que cuando el usuario ejecute la secuencia de control estas coordenadas pasen a ser posiciones deseadas y se pueda aplicar el algoritmo de control.

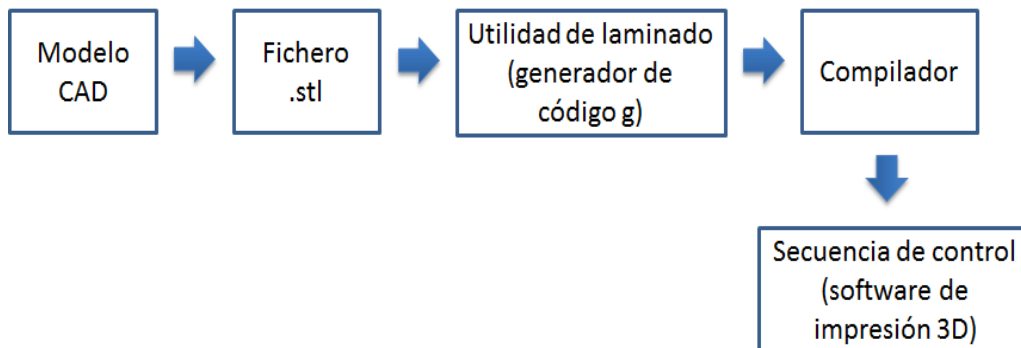


Figura 5.8: Diagrama esquemático de ciclo del software para impresión 3D.

▮ CAD (Diseño asistido por computadora)

Es el uso de programas computacionales para crear representaciones gráficas de objetos físicos, ya sea en segunda o tercera dimensión (2D o 3D).

Algunos ejemplos de software CAD:

- ✓ Solid Works

- ✓ NX
- ✓ Solid Edge
- ✓ Catia
- ✓ Open Scad

Utilizando estos programas podemos realizar nuestro modelo 3D y exportarlo al formato “.stl”.



Figura 5.9: Ejemplo de modelo 3D.

► ¿Qué es un archivo .stl?

Es un archivo digital de diseño asistido por computadora (CAD). Este archivo de extensión STL, adopta la sigla del inglés ‘STereo Lithography’ (estereolitografía). Este método de fabricación como ya mencionó anteriormente es lo que da origen a la impresión 3D. Actualmente los archivos .stl se utilizan como archivos de entrada para las impresoras 3D.

```
solid BikeOrnamentR2
facet normal 0.614857 0.788639 0.000000
  outer loop
    vertex -30.197639 27.567017 52.084904
    vertex 1.929615 2.519205 52.084896
    vertex -30.197639 27.567017 47.134903
  endloop
endfacet
facet normal 0.614857 0.788639 0.000000
  outer loop
    vertex -30.197639 27.567017 47.134903
    vertex 1.929615 2.519205 52.084896
    vertex 1.929615 2.519205 47.134899
  endloop
endfacet
facet normal 0.000000 0.000000 -1.000000
  outer loop
    vertex -19.878952 -24.914907 47.134899
    vertex -22.161694 -28.727596 47.134899
    vertex -31.466022 -17.928549 47.134899
  endloop
endfacet
facet normal 0.000000 0.000000 -1.000000
  outer loop
    vertex -31.466022 -17.928549 47.134899
    vertex -22.161694 -28.727596 47.134899
    vertex -36.494522 -20.085690 47.134899
  endloop
endfacet
```

Figura 5.10 Ejemplo de un archivo .stl

▮ Convertidor de archivos .stl a G-CODE (Utilidad de laminado)

El laminador tiene como entrada el archivo .stl previamente realizado con el software de diseño 3D y genera el fichero .gcode. Para realizar el laminado, tenemos que introducir características propias de nuestra máquina y algunos criterios de calidad.

Programas para la utilidad de laminado:

- Slic3r
- Skeinforge
- Big Brother

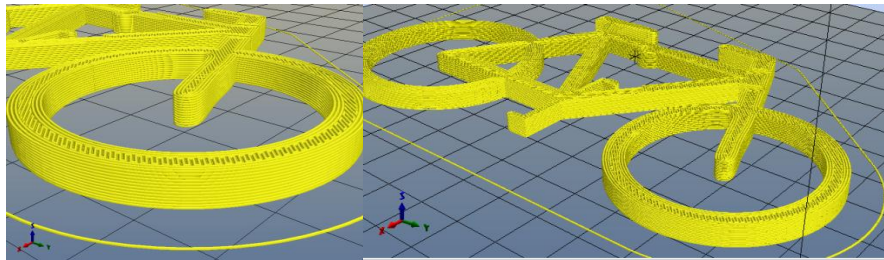


Figura 5.11: Ejemplo de pieza laminada

```
| generated by Slic3r 1.1.4 on 2014-12-03 at 11:11:23
```

```
; perimeters extrusion width = 0.50mm  
; infill extrusion width = 0.52mm  
; solid infill extrusion width = 0.52mm  
; top infill extrusion width = 0.52mm  
; support material extrusion width = 0.50mm
```

```
G21 ; set units to millimeters  
M107  
M104 S205 ; set temperature  
G28 ; home all axes  
G1 Z5 F5000 ; lift nozzle  
M109 S205 ; wait for temperature to be reached  
G90 ; use absolute coordinates  
G92 E0  
M82 ; use absolute distances for extrusion  
G1 F1800.000 E-1.00000  
G92 E0  
G1 Z0.350 F7800.000  
G1 X32.178 Y27.929 F7800.000  
G1 E1.00000 F1800.000  
G1 X33.932 Y26.442 E1.07480 F1080.000  
G1 X36.087 Y25.641 E1.14960  
G1 X37.295 Y25.531 E1.18906
```

Figura 5.12: Ejemplo de archivo .gcode

Compilador.

Debido a que el programa de laminado nos proporciona un archivo en código g, nosotros tuvimos la necesidad de realizar un software de compilación o depuración, donde le damos el formato necesario para nuestro software y adquirimos cada una de las coordenadas para el proceso de impresión 3D.

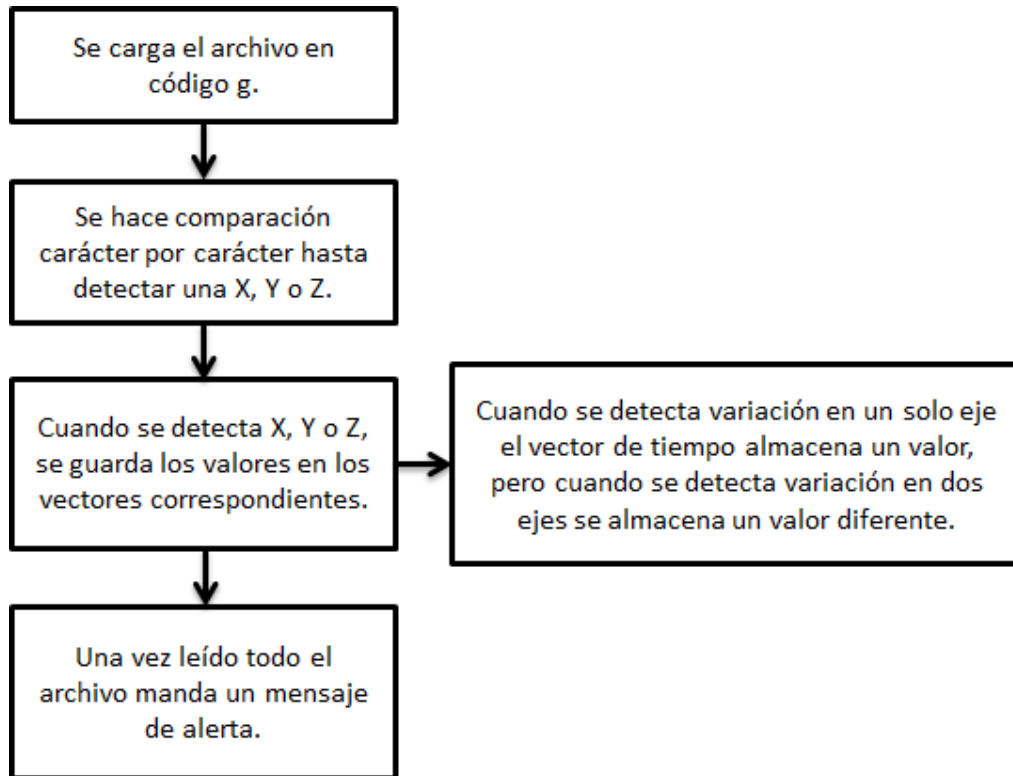


Figura 5.13: Diagrama a bloques del proceso de compilación.

En el diagrama anterior tenemos que lo primero que realizará nuestro software es la lectura de archivo en formato código g, nuestro software hace una lectura del archivo carácter a carácter buscando un símbolo igual a X, Y o Z, ya que después de esos símbolos, por el formato que tiene el código g vendrá la coordenada a la cual se debe mover nuestro robot cartesiano, por lo tanto una vez encontrado alguno de los símbolos anteriores se va almacenando la coordenada hasta encontrar un “espacio en blanco”, el cual nos indica que la coordenada ha terminado de formarse, este valor se guarda en el vector o arreglo correspondiente a cada eje.

| generated by Slic3r 1.1.4 on 2014-12-03 at 11:11:23

```
; perimeters extrusion width = 0.50mm  
; infill extrusion width = 0.52mm  
; solid infill extrusion width = 0.52mm  
; top infill extrusion width = 0.52mm  
; support material extrusion width = 0.50mm
```

```
G21 ; set units to millimeters  
M107  
M104 S205 ; set temperature  
G28 ; home all axes  
G1 Z5 F5000 ; lift nozzle  
M109 S205 ; wait for temperature to be reached  
G90 ; use absolute coordinates  
G92 E0  
M82 ; use absolute distances for extrusion  
G1 F1800.000 E-1.00000  
G92 E0  
G1 Z0.350 F7800.000  
G1 X32.178 Y27.929 F7800.000  
G1 E1.00000 F1800.000  
G1 X33.932 Y26.442 E1.07480 F1080.000  
G1 X36.087 Y25.641 E1.14960  
G1 X37.295 Y25.531 E1.18906
```



	Vector_X	Vector_Y	Vector_Z	Vector_t
131	94.117	105.883	-0.9	0.3
131	105.883	94.117	-0.9	0.3
135	105.893	94.842	-0.9	0.3
135	94.842	105.893	-0.9	0.3
135	95.578	105.893	-0.9	0.3
135	105.893	95.578	-0.9	0.3
135	105.893	96.313	-0.9	0.3
135	96.313	105.893	-0.9	0.3
135	97.048	105.893	-0.9	0.3
135	105.893	97.048	-0.9	0.3
135	105.893	97.784	-0.9	0.3
135	97.784	105.893	-0.9	0.3
135	98.519	105.893	-0.9	0.3
135	105.893	98.519	-0.9	0.3
135	105.893	99.255	-0.9	0.2
135	99.255	105.893	-0.9	0.3

Figura 5.14: Ejemplo de depuración del archivo en código g a solo vectores de coordenadas.

Para formar el arreglo o vector de tiempo, tomamos en cuenta, si el robot se mueve de forma recta o en diagonal, es decir si varía solo un eje nos indica una recta y variación en dos ejes una diagonal, dependiendo de esto será el valor que se almacene en el vector de tiempo, ya que utilizamos estos valores para generar dos velocidades una para rectas y otra para curvas o diagonales.

105.893	98.519	-0.9	0.3
105.893	99.255	-0.9	0.2
99.255	105.893	-0.9	0.3

← RECTA (VARIA SOLO UN EJE)
← DIAGONAL (VARIAN AMBOS EJES)

Figura 5.15: Ejemplo de cómo se forma el vector tiempo.

Con respecto al eje Z, tenemos que el código g solo nos proporciona un valor cuando varía, es decir, no se repite el mismo valor en todo el código, sin embargo, para nuestro software es necesario tener presente el valor, ya que, monitorea los tres ejes para realizar el movimiento solicitado, por lo tanto, en el software repetimos el valor adquirido para el eje Z hasta detectar un nuevo valor.

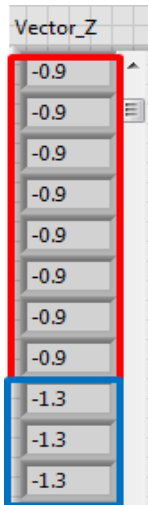


Figura 5.16: Ejemplo de cómo se forma el vector Z.

Para conseguir lo anterior se tuvo que seguir la siguiente metodología:

1. Se debe abrir el archivo que contenga el código g a realizar. Se determina el número de caracteres del archivo y se cambia el formato “.txt” a “string”, para poder manipular la información.

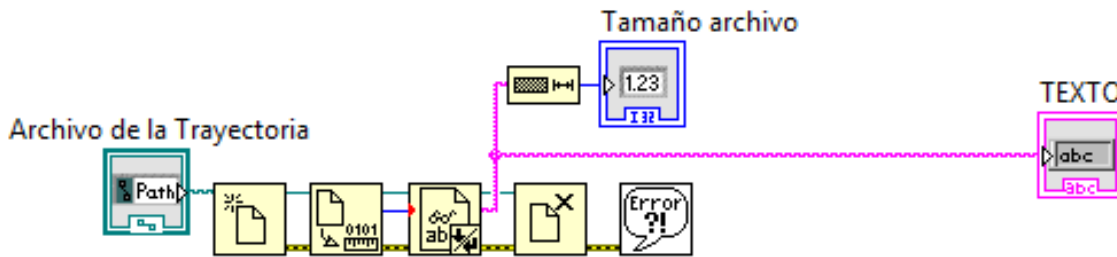


Figura 5.17: Apertura del archivo a compilar.

2. Se realiza convierte el archivo en arreglo para poder tener acceso a cada uno de los caracteres de forma independiente, posteriormente se vuelve a convertir a tipo “string”, para poder continuar con el proceso.

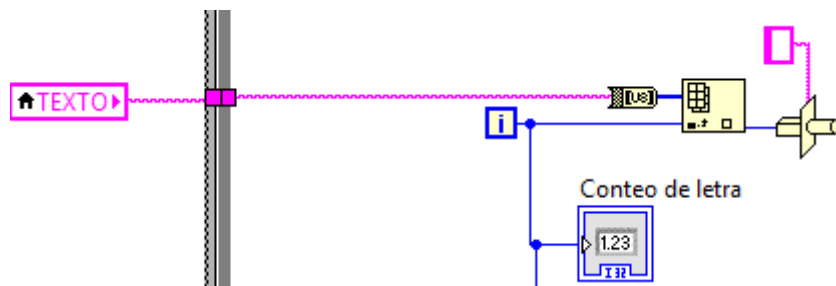


Figura 5.18: Conversión de “string” a “array”.

- Después se realizan una serie de comparaciones para identificar cuando el carácter es X, Y o Z, ya que extraeremos el valor siguiente y se ordenará en el arreglo correspondiente a su eje.

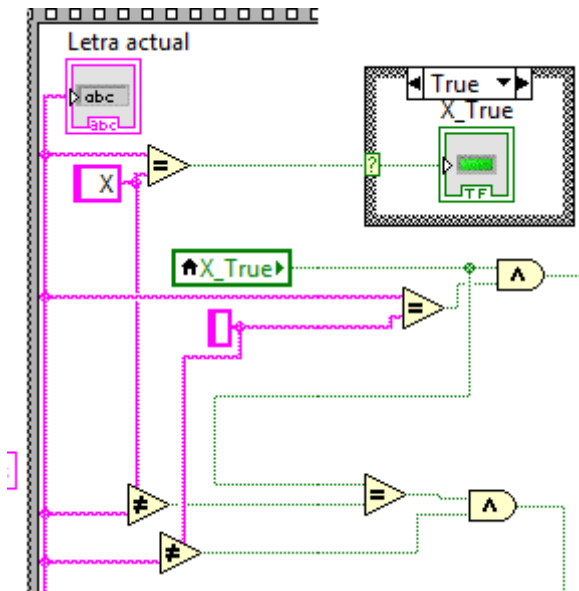


Figura 5.19: Comparaciones para iniciar la extracción de coordenadas.

- Como se adquiere carácter por carácter, se debe realizar una concatenación para acomodar el valor de la posición deseada.

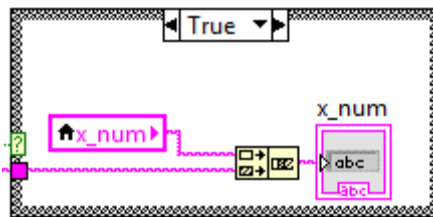


Figura 5.20: Sección donde se realiza la concatenación de la coordenada.

- Una vez que se tiene el valor completo de la coordenada, se debe realizar nuevamente una conversión de tipo de dato a "doble", para poderlo ordenarlo posteriormente en el arreglo final. Además de inicializa la variable que contiene el dato, para evitar información errónea.

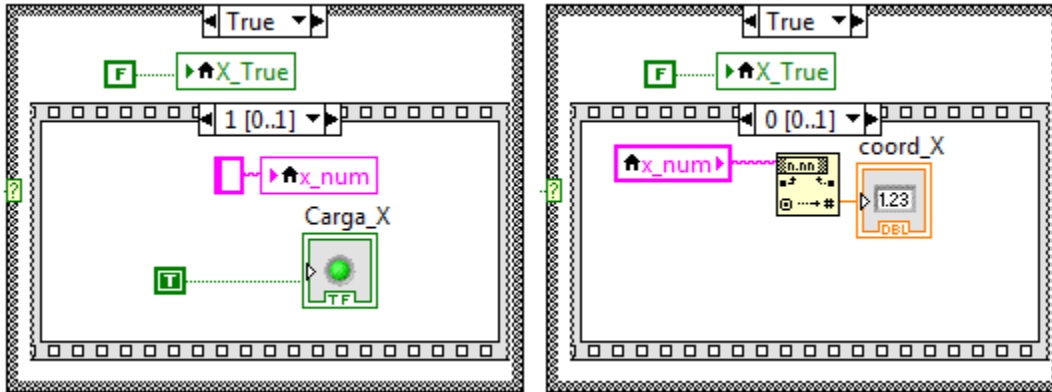


Figura 5.21: Conversión de “string” a “dbl” e inicialización de variables.

6. Cuando el valor ya tiene el formato deseado se empieza a formar el arreglo por eje.



Figura 5.22: Formación del arreglo del eje X.

7. Para determinar si el robot manipulador está haciendo una recta o una diagonal, se debe primero tener un antecedente del punto anterior por lo que se asigna el valor obtenido a otra variable para poder realizar este proceso.

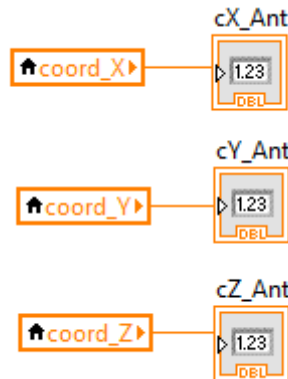


Figura 5.23: Asignación de valor a coordenada anterior.

8. Se realiza una comparación para determinar si tiene una recta en el eje X o Y, si ambas se mueven tendremos una diagonal. Con respecto al eje Z no tenemos problemas porque este eje se mueve cada vez que termina una capa de la impresión, por lo que no aplica el caso.



Figura 5.24: Determinación de rectas o diagonal.

- Lo anterior se realiza para poder obtener el valor del vector de tiempo, el cual nos servirá para regular la velocidad a la hora de implementar nuestro algoritmo de control.

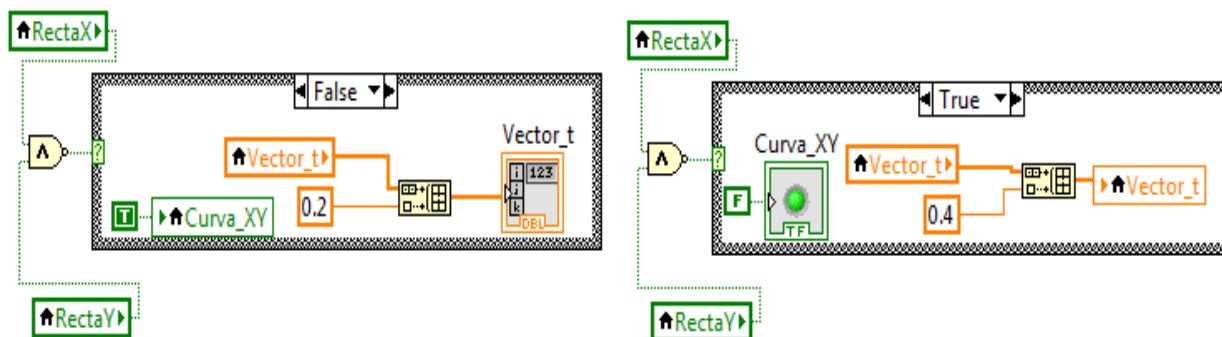


Figura 5.25: Formación del arreglo de tiempo.

- Con respecto al eje Z, ya se había mencionado que se tiene otra situación, en la cual solo se tiene un valor, cada cierto número de líneas por lo que se debe de repetir este valor. Sin embargo, también se tiene ajustar la altura, ya que los valores de este eje tienen otra orientación.

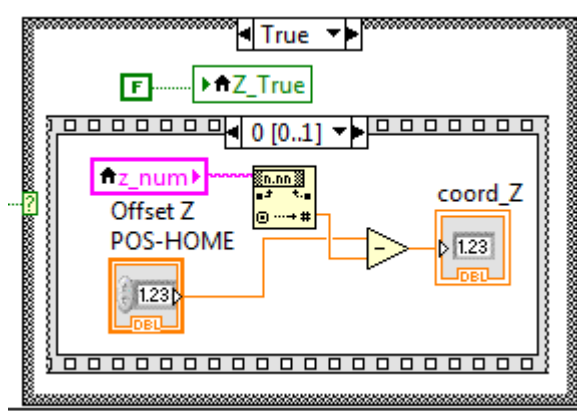


Figura 5.26: Formación de la coordenada Z.

- Finalmente, una vez generados todos los arreglos se debe generar el arreglo principal que se utilizará en la secuencia de control del software.

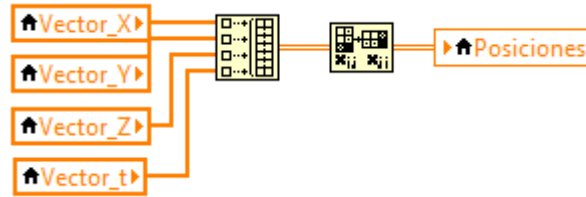


Figura 5.27: Arreglo principal.

Con este proceso obtendremos las posiciones deseadas para cada uno de los ejes de nuestro robot manipulador, a partir del archivo que nos brinda el laminador, para que posteriormente se pueda aplicar la secuencia de control.

5.2 Parámetros de la herramienta de impresión 3D

Como hemos mencionado, trabajamos sobre el software de control de posición, por lo que faltaba agregar una parte que nos permitiera enviar por el puerto PCI la temperatura deseada tanto del extrusor como de la base caliente, al firmware de control de temperatura, además, de realizar el software que nos permitiera visualizar la temperatura actual que envía el FPGA por el puerto PCI.

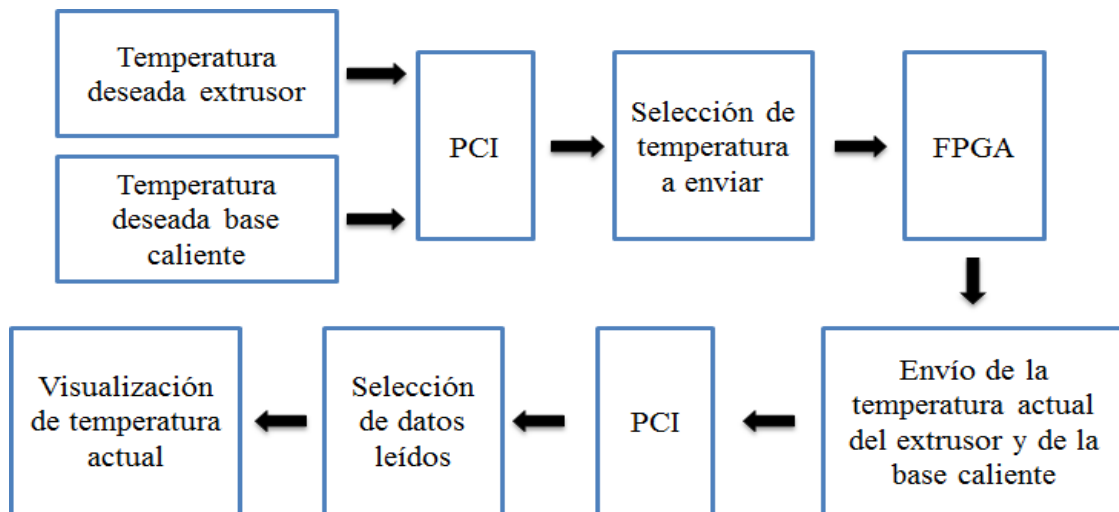


Figura 5.28: Diagrama a bloques del envío de parámetros.

Para el envío de la temperatura deseada por el puerto PCI, primero se definieron los pines del FPGA que se iban a utilizar para este proceso, posteriormente se define si la temperatura deseada es del extrusor o de la base caliente, enviando una combinación de bits 01 para la temperatura del extrusor y 10 para la temperatura deseada de la base caliente, este bit de control se envía por el puerto PCI a los pines 20 y 21.

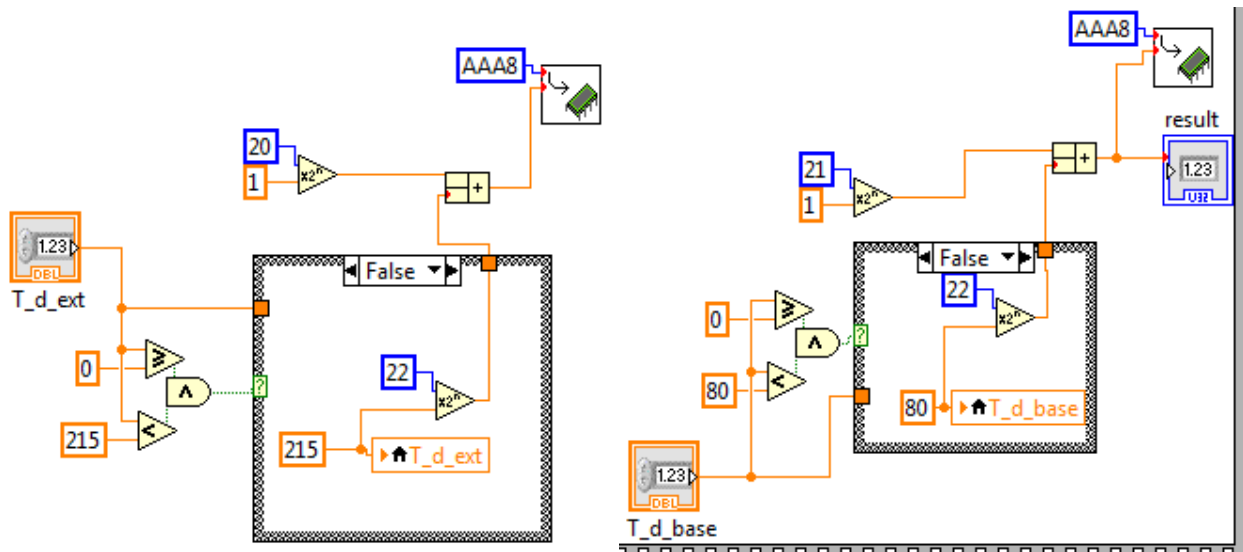


Figura 5.29: Envío de la temperatura deseada del extrusor y de la base caliente, por el bus PCI.

En la figura anterior se puede observar que además de enviar el dato de la temperatura deseada del extrusor y de la base caliente, estás también se delimitan para evitar problemas de funcionamiento.

Para iniciar el envío de datos se tiene un botón que nos permite controlar el proceso, ya durante el envío se realiza un multiplexeo entre el envío del dato para el extrusor y para la base caliente, ya que, no se cuentan con suficientes pines en el puerto PCI, para poder enviar ambos al mismo tiempo.

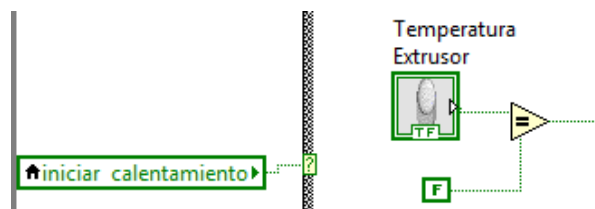


Figura 5.30: Control del envío de datos

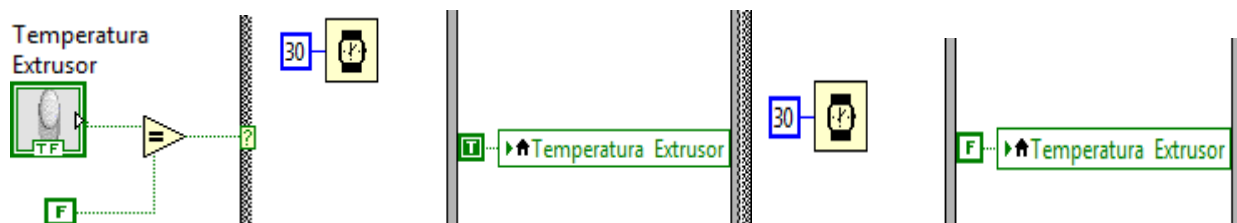


Figura 5.31: Parte del software que realiza el multiplexeo del envío de la temperatura deseada.

Anteriormente ya se habló del envío de la temperatura deseada, pero falta la sección de lectura de la temperatura actual tanto de la base caliente como del extrusor.

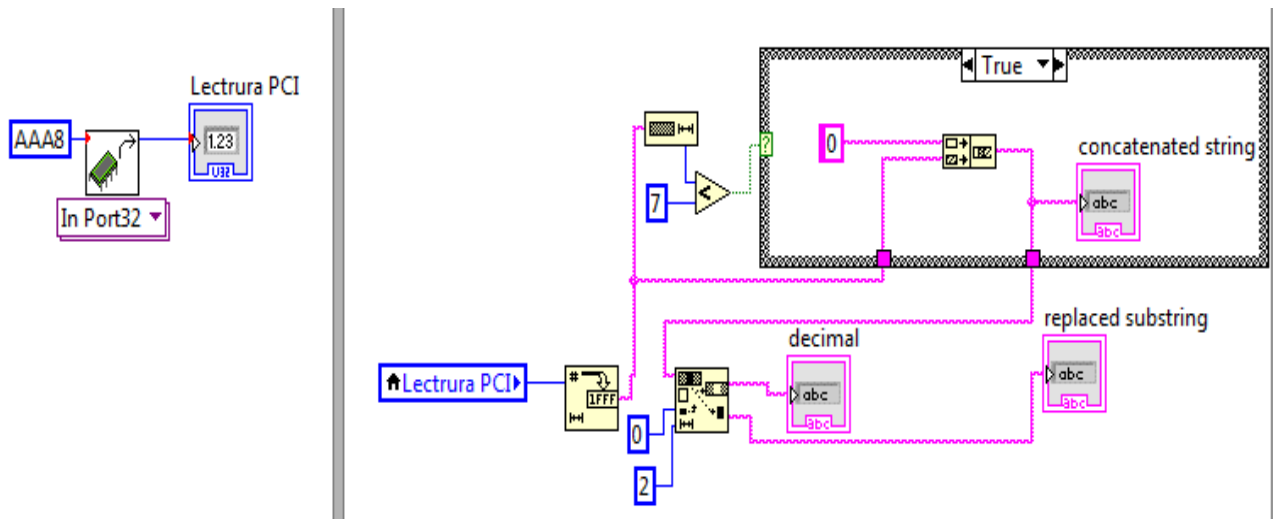


Figura 5.32: Lectura del puerto PCI y selección de datos.

Para ello, primero se debe leer el puerto PCI, se le asigna el valor del puerto a una variable, esta variable se encuentra en formato “unsigned de 32 bits”, por lo que se cambia su formato a “string tipo hexadecimal”, ya que es más fácil seleccionar los bits que corresponden a la temperatura del extrusor y los que corresponden a la temperatura de la base caliente. Posteriormente, se vuelve a realizar un cambio de formato a “dbl”, para poderlo visualizar en la pantalla principal del software del control.

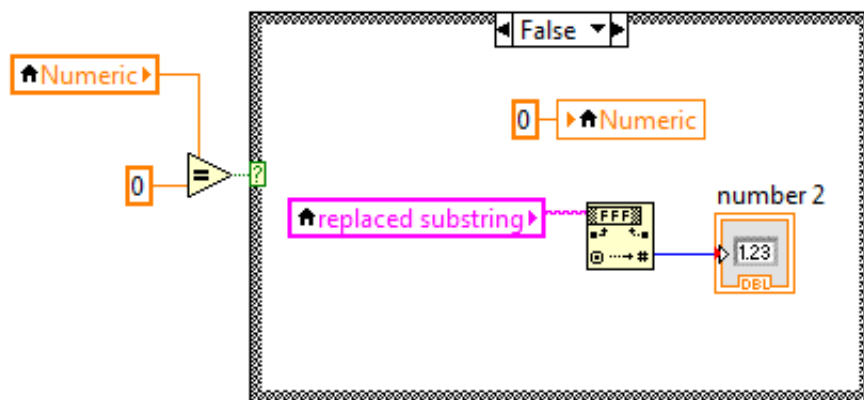


Figura 5.33: Conversión de dato seccionado “string” a “dbl”.

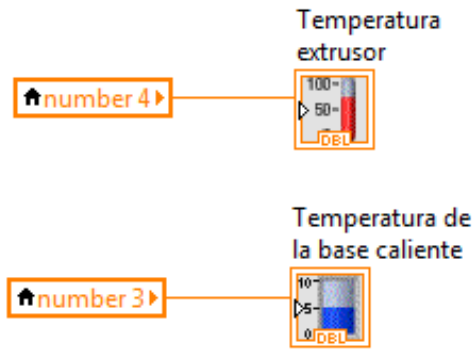


Figura 5.34: Envío de dato para visualización del usuario.

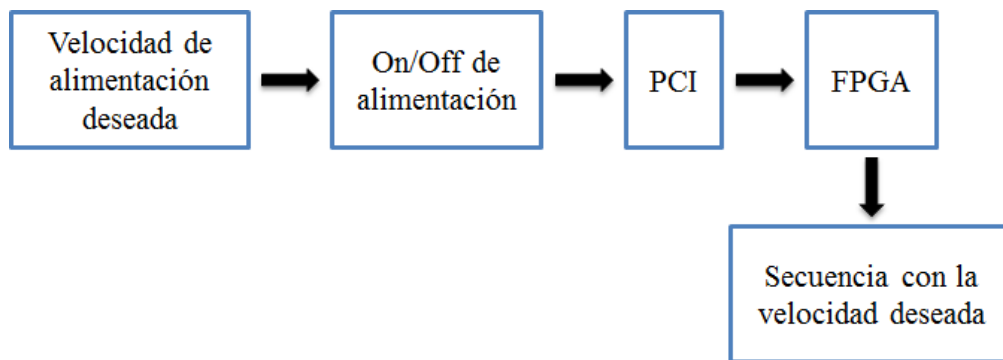


Figura 5.35: Diagrama a bloques del envío de la velocidad de alimentación.

Por otro lado, también es necesario realizar le envío por el puerto PCI del dato que contenga la velocidad de alimentación de material, en el capítulo anterior se mencionó que se tienen tres diferentes frecuencias de trabajo, solo se tiene que indicar al bloque de control del FPGA cuál de estas frecuencias será la que se emplee durante la impresión 3D. Además, se tiene que enviar un bit que active la secuencia del motor a pasos o alimentador de material.

Los datos anteriores se tienen que estar mandando a escribir cada vez que el software envíe información al puerto PCI, ya que se formatea cada vez que se repite este proceso. Lo mismo sucede para la lectura del puerto, ya que los datos que envíe el FPGA se van actualizando constantemente.

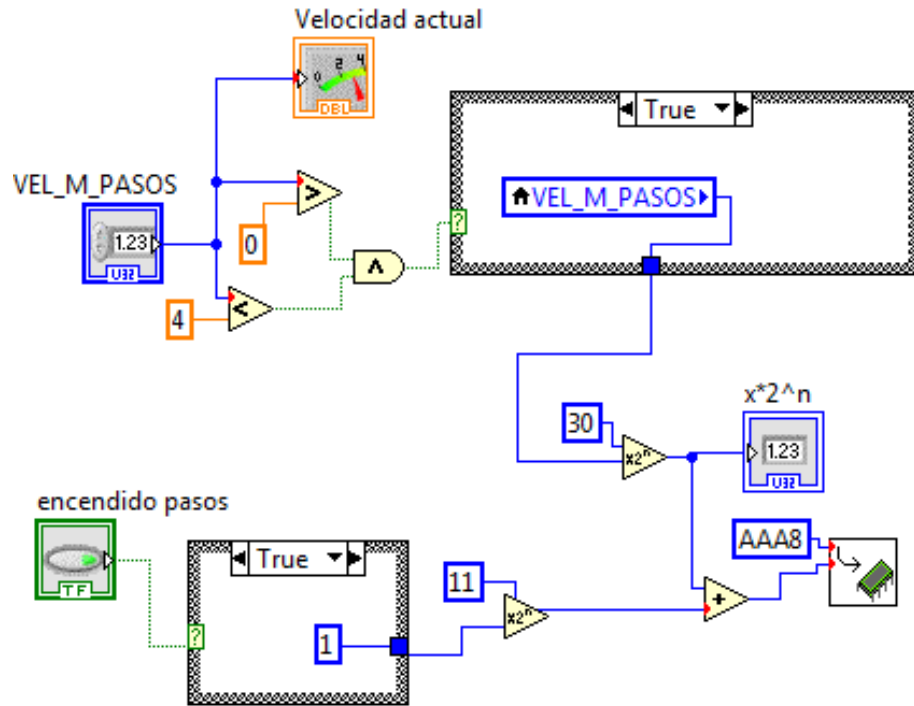


Figura 5.36: Envío de la velocidad de alimentación por el puerto PCI.

5.3 Conclusiones

Este capítulo aborda como se realiza el proceso para obtener las posiciones deseadas a partir de un diseño CAD, donde el software realiza una depuración del archivo producido por el laminador. Gracias a este proceso de compilado no es necesario involucrar más software para realizar nuestra impresión 3D. Por otro lado, se menciona como el software ocupa el puerto PCI para enviar los parámetros necesarios para realizar el proceso de control de temperatura en nuestro firmware, además, de realizar la lectura de datos para que el usuario pueda visualizar la temperatura actual de la herramienta de impresión en tiempo real.

Capítulo 6. Resultados experimentales

Para esta sección se hicieron pruebas eje por eje, para confirmar la efectividad de nuestro control de posición, para ello se realizaron las gráficas de error de posición, par aplicado y velocidad de desplazamiento.

6.1 Resultados de control del CNC

En esta prueba se solicitaron dos posiciones -50mm/s y -30mm/s, y finalmente regresar a la posición de casa.

Grafica de error de posicion en el eje Z

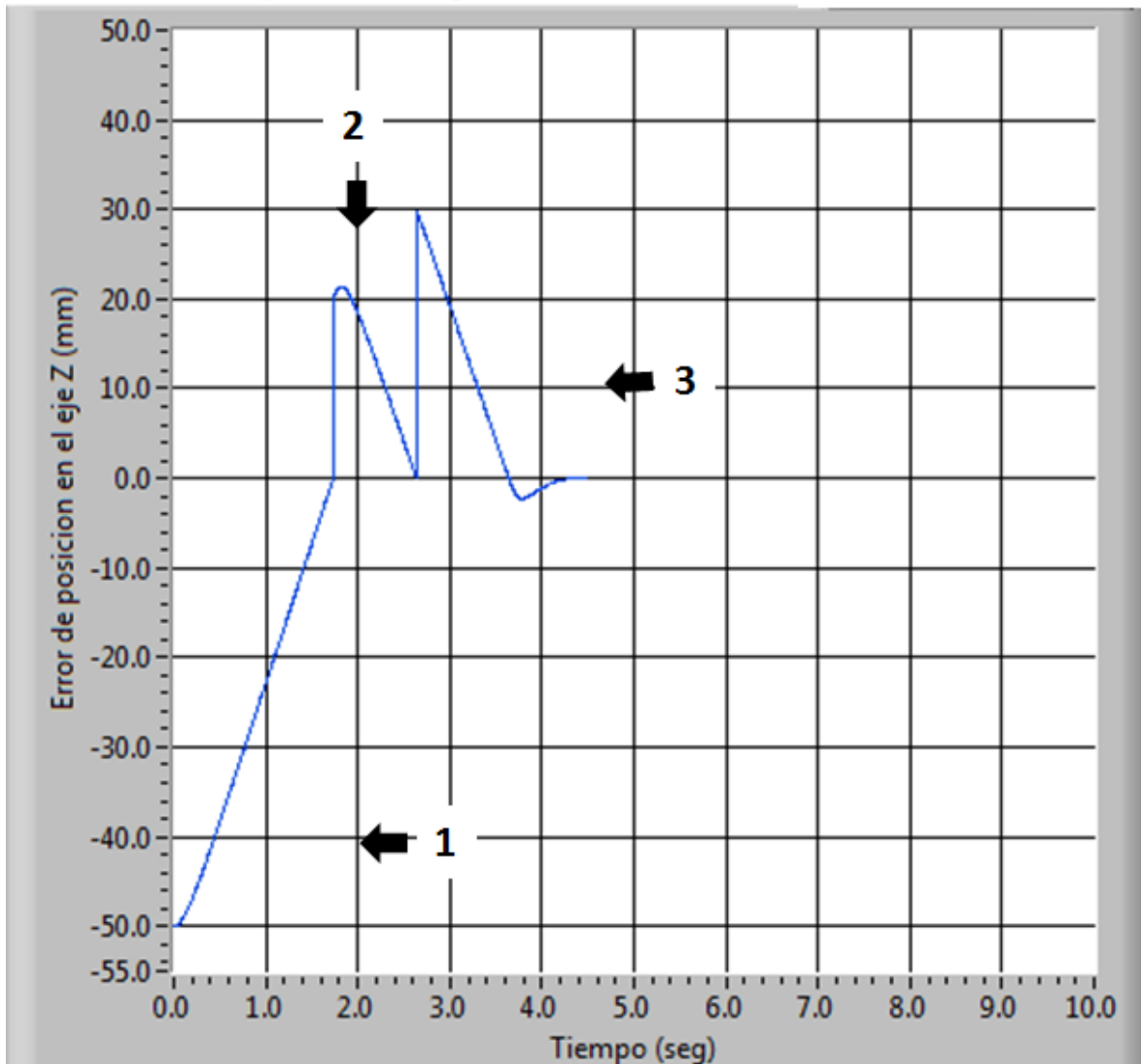


Figura 6.1: Gráfica de error de posición del eje Z.

Como se puede observar en la gráfica anterior, en la parte “1” en el primer instante de tiempo el error de posición será de -50mm, conforme el robot manipulador se va desplazando a la posición deseada, el error va disminuyendo, y al llegar al segundo 1.8, nuestro eje llega a la posición deseada por lo tanto el error de posición es igual a cero, en ese momento se actualiza la posición deseada a -30mm, en la parte “2”, la diferencia entre ellos será de 20mm [posición deseada – posición actual= error de posición $\rightarrow -30\text{mm}-(-50\text{mm})=20\text{mm}$], que será nuestro nuevo valor para nuestro error de posición, que será alcanzado al segundo 2.6, finalmente, en la parte “3” se desea regresar a posición de casa, por lo que el error será de 30mm que serán alcanzado al segundo 4.4. Con lo anterior se comprueba la eficiencia de nuestro algoritmo de control para alcanzar la posición deseada con una buena velocidad de respuesta.

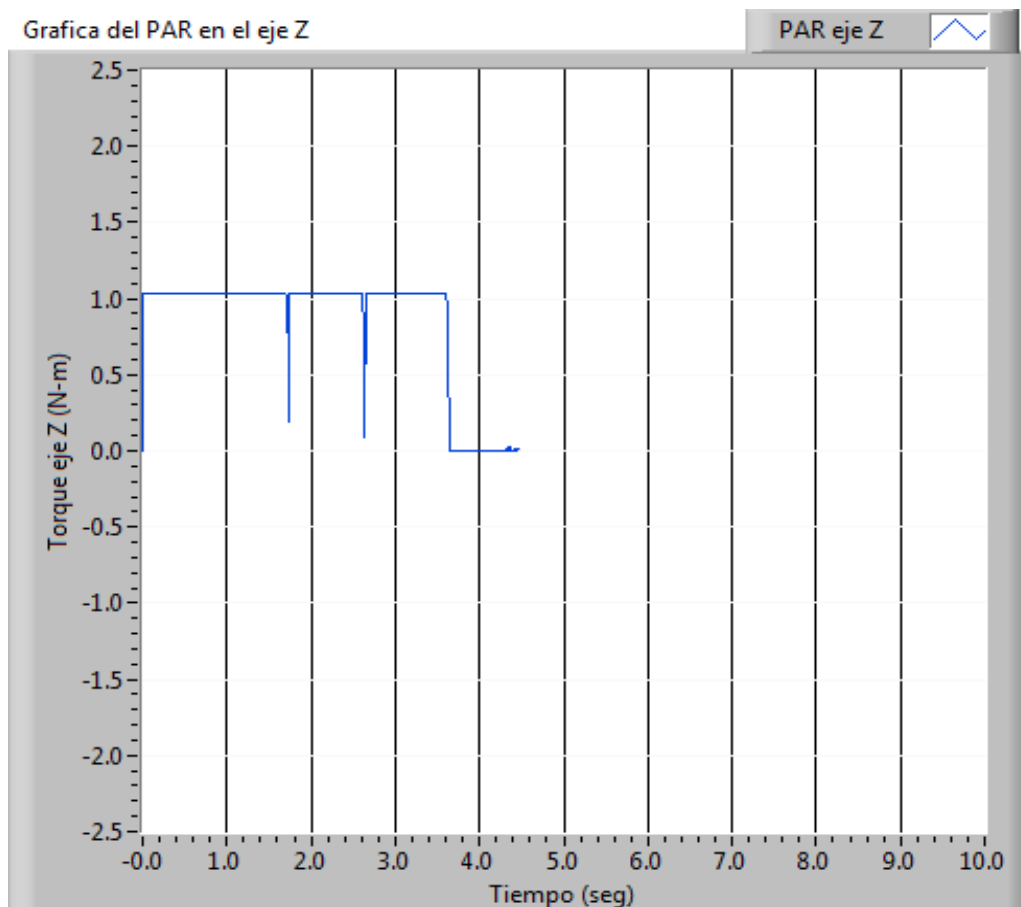


Figura 6.2: Gráfica de par eje Z.

Por otro lado, con respecto al par aplicado tenemos que el máximo par solicitado a nuestro direct drive fue de 1.05 Nm, otra cosa que podemos resaltar, es que el par no deja de ser aplicado durante el proceso, debido a que le estamos solicitando movimiento constante, hasta que llega a su posición deseada.

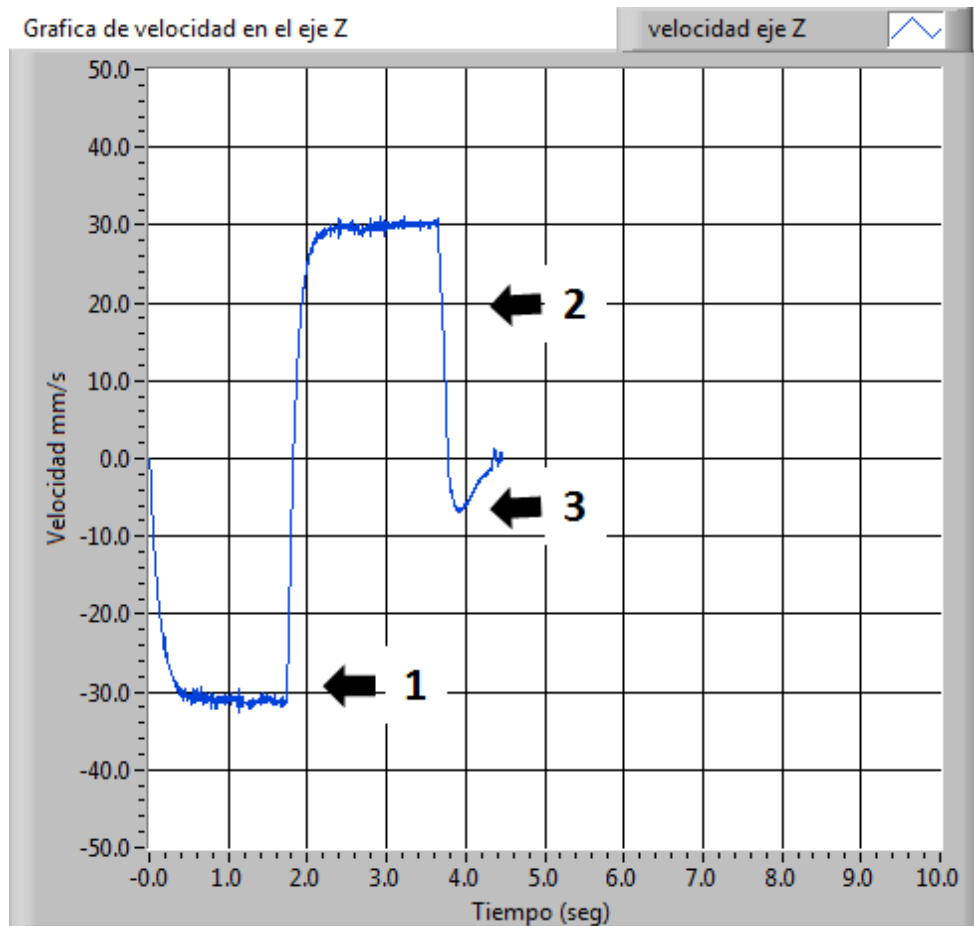


Figura 6.3: Gráfica de velocidad en el eje Z.

En la gráfica anterior tenemos que la velocidad máxima que alcanza nuestro eje Z, para este caso tenemos que la velocidad máxima será de 3cm/s, con una aceleración de 0.4 segundos, para la parte “1”, en la gráfica se describe con el signo de menos debido a la dirección, para la parte “2”, vemos que la velocidad alcanzada será de 3cm/s, pero en sentido a las manecillas del reloj, por lo que se maneja el signo positivo y para la parte “3”, tenemos una pequeña medición de velocidad esto debido a que nuestro algoritmo de control tiene un pequeño sobretiro al llegar a la posición deseada, y realiza el regreso al punto deseado.

Ahora probaremos el eje Y, para ello solicitaremos que llegue a los puntos -40mm y 50mm, con respecto a nuestra primera posición.

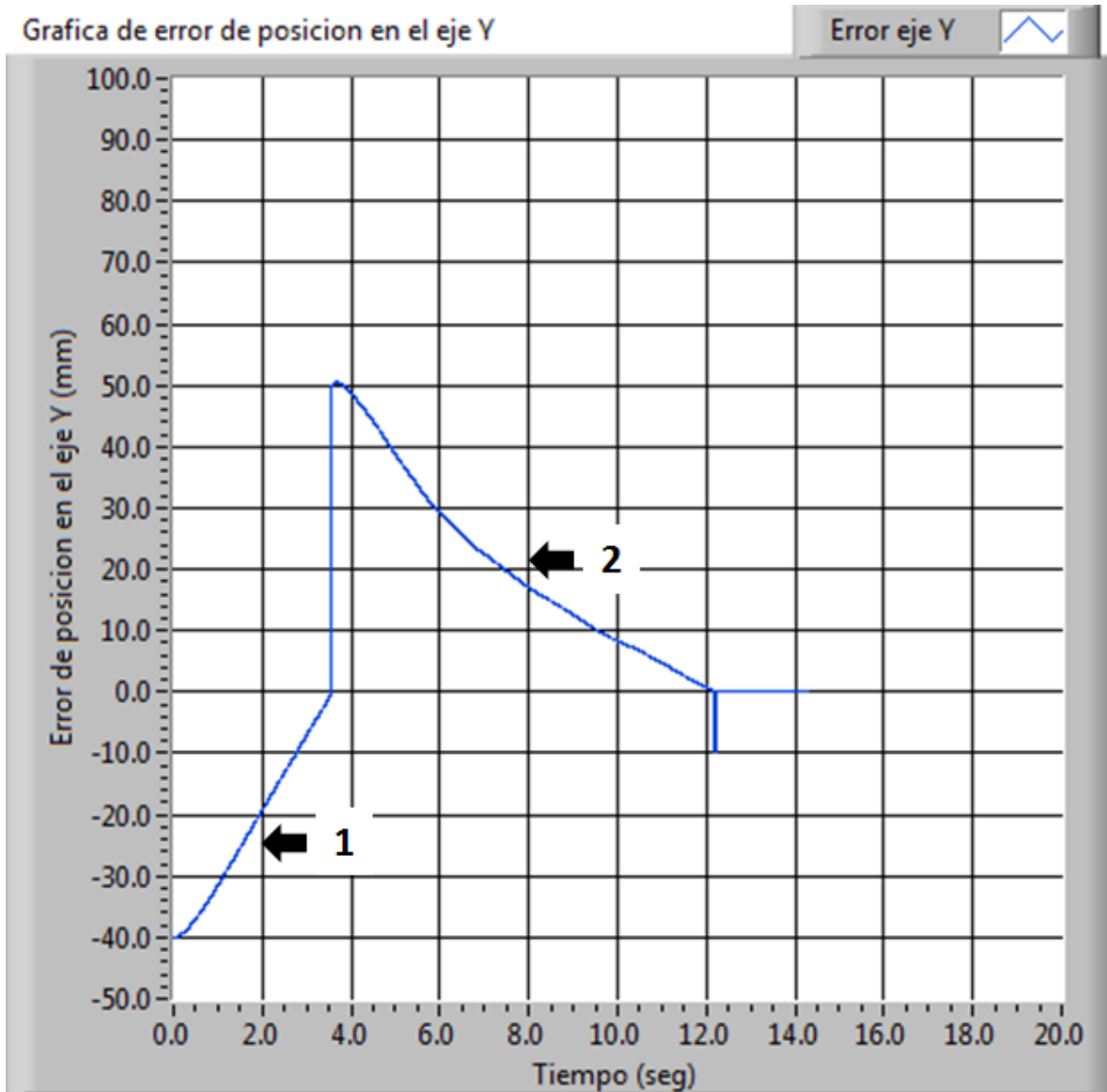


Figura 6.4: Gráfica de error de posición en el eje Y.

En la gráfica anterior tenemos que la parte “1” nuestro error empieza en -40mm, pero en el segundo 2.8 se llega a la posición deseada, por lo que en ese instante de tiempo el error de posición cambia a 50mm y va disminuyendo gradualmente hasta el segundo 12.1 cuando alcanza la última posición deseada.

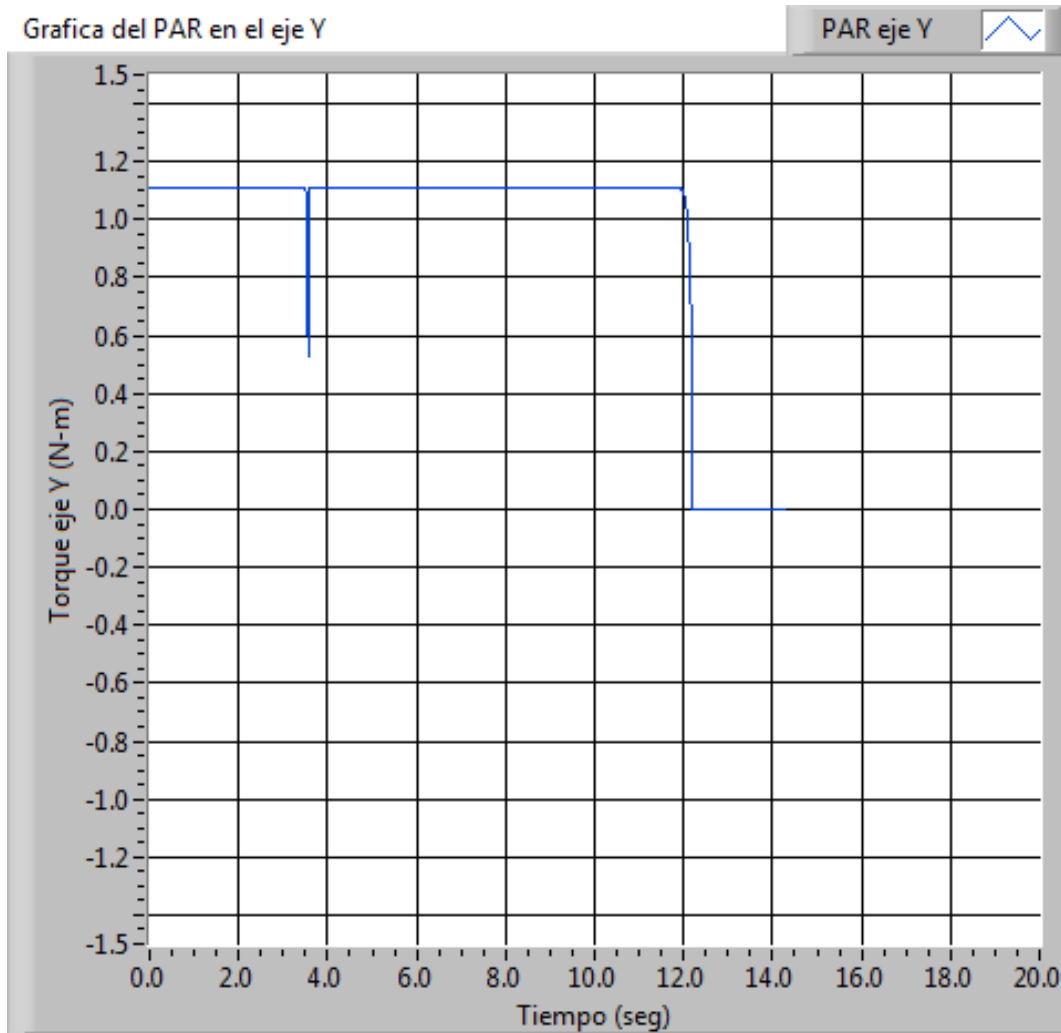


Figura 6.5: Gráfica de par aplicado.

En la gráfica anterior tenemos que el máximo par aplicado es de 1.1Nm, debido a nuestro algoritmo de control tenemos que el par aplicado cambia rápidamente del par máximo a cero una vez alcanzada la posición deseada, de forma análoga en un breve instante de tiempo alcanza el par máximo una vez solicitada una nueva posición deseada.

Conclusiones generales

Para efectuar esta tesis se realizó un estudio del funcionamiento de las impresoras 3D, que existían en el mercado, para determinar cómo era el proceso que se tenía que desarrollar con el robot manipulador que ya se contaba, de acuerdo a ello se procedió con la adquisición de la herramienta que se implementaría para la impresión 3D.

Además, se estudió la cinemática y dinámica del robot CNC que nos ayudó a entender mejor el comportamiento de nuestro sistema, como saber que elementos intervienen en la determinación del par que se aplicará a nuestros actuadores. Por otro lado, también se estudió la mecánica de nuestro manipulador, para comprender como se realiza la transmisión mecánica y el desplazamiento de precisión.

Durante la implementación de la herramienta 3D, se aprovechó para el cambio de nuestros motorreductores, ya que se tiene la idea de hacer prototipos rápidos tanto impresos como maquinados, para ello se realizó la caracterización de los mismos con la finalidad de utilizarlos como motores de transmisión directa, para lo que se utilizó una metodología desarrollada en la maestría, también se realizó la caracterización del extrusor, herramienta importante para nuestra impresión. Además, se actualizó la electrónica y los drivers del sistema que pudieran soportar estos nuevos actuadores.

Con respecto al firmware, se incluyó el control de nuestra herramienta de control, mediante un control proporcional, con el cual se controla la temperatura de la base caliente como del extrusor, también se realiza el control de la alimentación de material siendo pasos importantes para la impresión 3D. También se incorporó la comunicación con el puerto PCI que nos permitiera visualizar parámetros del sistema en nuestro software de control.

En el software de control se realizaron las modificaciones pertinentes para poder decodificar los archivos que nos proporciona el software de laminado, y de esta manera utilizar las coordenadas deseadas para poder realizar nuestro control. En este software se realiza el control de posición que maneja nuestros actuadores mediante una tangente hiperbólica como ley de control, además de permitir al usuario visualizar y modificar los parámetros del sistema durante el proceso de impresión, como la temperatura deseada o la velocidad de alimentación.

Durante el proceso se tuvo un aprendizaje completamente mecatrónico, lo cual sirvió para reafirmar lo aprendido en la maestría. Y proponer mejoras al sistema actual, como actualizar la interfaz de PCI a Wi-fi, poder implementar otras herramientas para desarrollar otros procesos como desbaste de material, o palpado de piezas, debido a que el sistema es muy versátil. Para proyectos más complejos se podría agregar otros dos grados de libertad, con los cuales se podría realizar procesos más complejos y de mayores prestaciones.

Referencias

- [1] “Robótica: Estado del arte”, noviembre 2016, disponible en: www.academia.edu/913608
- [2] “Impresoras 3D: historia, cómo trabajan y algunas de las impresoras más populares”, noviembre 2016, disponible en: geeksroom.com/2014/02/impresoras-3d/82881
- [3] Impresoras 3D, noviembre 2016, disponible en: www.areatecnologia.com/informatica/impresoras-3d.html
- [4] Cube, noviembre 2016, disponible en: <http://cubify.com/en/Cube>
- [5] Makerbot, noviembre 2016, disponible en: store.makerbot.com/replicator-mini
- [6] Impresora 3D, noviembre 2016, disponible en: <http://reprap.org/>
- [7] Kelly Rafael, “Control de Movimiento de Robots Manipuladores”, Prentice Hall, 2003
- [8] Fernando Reyes Cortes, “Robótica control de robots manipuladores”, alfa omega
- [9] Riel de alta precisión, noviembre 2016, disponible en: <http://www.preciolandia.com/mx/cnc-riel-guia-lineal-20mm-para-routers-c-1bc6ap-a.html>.
- [10] Concepto de pick and place, noviembre 2016, disponible en: <http://www.motoman.lt/es/Solutions/Applications/Pick--place/>.
- [11] Información motor, noviembre 2016, disponible en: <http://longkang-motor.en.made-in-china.com/product/lozxPcJbOKYg/China-24V-DC-Gear-Motor-Z5D120-.html>
- [12] Información motor, noviembre 2016, disponible en: http://www.crestmi.com.au/DC_precision_gearmotors.php
- [13] Información encoder, noviembre 2016, disponible en: http://www.ebay.com/itm/231160065963?_trksid=p2060353.m2749.l2649&ssPageName=STRK%3AMEBIDX%3AIT
- [14] Bezares Colmenares Gibram Humberto, “Robot CNC Multiherramienta”, 2014
- [15] ABS y PLA: diferencias, ventajas y desventajas, noviembre 2016, disponible en: <http://www.impresoras3d.com/abs-y-pla-diferencias-ventajas-y-desventajas/>
- [16] Metal J-head short-distance, noviembre 2016, disponible en: www.geeetech.com/wiki/index.php/Metal_J-head_short-distance
- [17] PCB Heatbed, noviembre 2016, disponible en: http://reprap.org/wiki/PCB_Heatbed
- [18] Reprap 3D Printer Geared Stepper Extruder V2.2 w/ Quick-Change Feature- Blue ABS, noviembre 2014, disponible en: <http://www.ebay.ca/itm/Reprap-3D-Printer-Geared-Stepper-Extruder-V2-2-w-Quick-Change-Feature-Blue-ABS-/291159224780>

Apéndice A



IEEE
MORELOS SECTION

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS

CONSTANCIA

Artículo: "*Design and construction of an acquisition system to control a Cartesian robot as a 3D printer.*"

Autores: Alfredo Lopez Beltran, Sergio Vergara, Aurora Vargas, Fernando Reyes and Amparo Palomino

Id. artículo: 48

El Comité Técnico del XIII Congreso Internacional Sobre Innovación y Desarrollo Tecnológico CIINDET 2016, que se llevó a cabo en la Ciudad de Cuernavaca, Morelos, México, del 7 al 9 de septiembre de 2016, hace constar que el artículo citado fue presentado de acuerdo con el programa técnico del congreso e incluido en las memorias del mismo.

La presente constancia se expide para los fines legales que a los autores convengan.
Cuernavaca, Morelos, México a 9 de septiembre de 2016.

Atentamente
Dr. Jorge Guillermo Calderón Guizar
Presidente del Comité Técnico CIINDET 2016





CIINDET 2016
HACIA UN DESARROLLO SUSTENTABLE
CON TECNOLOGÍA PROPIA



Sección Morelos

La Sección Morelos del Instituto de Ingenieros
en Electricidad y en Electrónica

Otorga el presente

Reconocimiento

A: ALFREDO LÓPEZ BELTRÁN

Por su participación como

Congresista

Durante el XIII Congreso Internacional sobre Innovación y Desarrollo Tecnológico,
realizado del 7 al 9 de Septiembre del 2016 en la ciudad de Cuernavaca, Morelos, México.

Alfredo López Beltrán

Dr. Francisco Ponce Maldonado
Presidente del Cindet 2016

Design and construction of an acquisition system to control a Cartesian robot as a 3D printer.

A. López, S. Vergara, M.A. Vargas, F. Reyes, A. Palomino.

Abstract: This project focuses on the design and construction of an acquisition system to control a Cartesian robot, which will be used to make 3D prints, being controlled by an FPGA (Field Programmable Gate Array) unlike conventional 3D printers; and using as PCI interface port on your computer. It will be mentioned issues such as the dynamic model of the Cartesian robot, using the methodology of Euler-Lagrange, the design of acquisition system implemented data, and the software made in LabVIEW be addressed, which we use to implement the control algorithm and platform for user interface. It will also be a description of the firmware loaded into the FPGA of the Altera Cyclone III brand family, which has the function to communicate our Cartesian robot with computer and run the temperature control of our printing tool.

Keywords: Cartesian Robot, FPGA, port PCI and 3D printers.

Resumen: Este trabajo se enfoca en el diseño y construcción de un sistema de adquisición para controlar un robot cartesiano, el cual será utilizado para realizar impresiones 3D, siendo controlado por un FPGA (Field Programmable Gate Array) a diferencia de las impresoras 3D convencionales y utilizando como interfaz el puerto PCI de la computadora. Se abordaran temas como, el modelo dinámico del robot cartesiano, utilizando la metodología de Euler-Lagrange, el diseño del sistema de adquisición de datos implementado, además del software realizado en LabVIEW, el cual utilizamos para implementar el algoritmo de control y plataforma para la interfaz usuario. También se mencionará una descripción del firmware cargado en el FPGA de la marca Altera de la familia Cyclone III, el cual tiene la función de comunicar nuestro robot cartesiano con la computadora y ejecutar el control de temperatura de nuestra herramienta de impresión.

Palabras clave: Robot cartesiano, FPGA, puerto PCI e impresión 3D.

INTRODUCCIÓN.

Una de las actividades más relevantes para un país, es el desarrollo de tecnología, ya que influye directamente en todas las áreas importantes, como la economía, el medio ambiente, pero principalmente en la sociedad. Los países desarrollados que cuentan con alta tecnología en sus industrias, les permite aumentar sus índices de producción y calidad a costos reducidos, favoreciéndolos en la competencia comercial tanto nacional como internacional, generando una cinergia económica-social.

Actualmente los robots manipuladores y CNC son maquinarias muy importantes en diferentes áreas, tanto productivas, como de la salud e investigación, ya que tienen la versatilidad de cumplir con diversas actividades mientras se les controle apropiadamente. En este trabajo abordaremos una de las aplicaciones que se le puede dar a los robots CNC o también llamados robots cartesianos con 3 grados de libertad, utilizándolo para la implementación de una impresora 3D, teniendo un FPGA, lo cual la vuelve única en su tipo permitiéndonos irle enriqueciendo y modificando con otros aditamentos.

Las impresoras 3D son dispositivos que han ido ganando mucha fama a nivel industrial y de diseño, debido a las ventajas que presenta para el diseño de prototipos de cualquier índole, teniendo un bajo costo de inversión y poco tiempo de fabricación, sin embargo también han sido utilizadas para el área de la salud, ya que se ha conseguido "imprimir" algunos tipos de órganos y prótesis, que se adecuan a cada persona ya que son "diseños especiales" para ellos.

La finalidad de este proyecto es realizar impresiones de alta calidad, esto lo conseguiremos mediante la precisión del robot CNC, que es de 2 micras, y a través de realizar estudios de comportamiento del material y de los dispositivos involucrados en el proceso, caracterizando cada uno de ellos para conseguir su mejor desempeño de trabajo.

II. DISEÑO DEL SISTEMA DE ADQUISICIÓN

Nuestra impresora 3D, está constituida principalmente por dos partes como se muestra en la figura (1), la primera parte involucra a nuestro robot cartesiano de 3 grados de libertad, en el cual, se encuentra acoplada la herramienta de impresión, este robot está conformado por tres servomotores. Las señales de los encoders son leídas por medio de la tarjeta de adquisición PCI, e interpretadas por nuestro software y el usuario puede ver en tiempo real el cambio de posición del robot cartesiano. Por otro lado, la segunda parte principal de nuestro sistema, es la herramienta de extrusión, que está compuesta por el alimentador de material, que es un motor a pasos acoplado a un sistema mecánico para la sujeción del filamento (material a extruir), y su segundo componente es una pieza metálica que

Alfredo López, Benemérita Universidad Autónoma de Puebla, Av. san Claudio y 18 Sur, Ciudad Universitaria, México, fredal2014@gmail.com

Sergio Vergara, Benemérita Universidad Autónoma de Puebla, Av. san Claudio y 18 Sur, Ciudad Universitaria, México, sergio.vergara@correo.buap.mx

M. Aurora Vargas, Benemérita Universidad Autónoma de Puebla, Av. san Claudio y 18 Sur, Ciudad Universitaria, México, aurora.vargas@correo.buap.mx

Fernando Reyes, Benemérita Universidad Autónoma de Puebla, Av. san Claudio y 18 Sur, Ciudad Universitaria, México, recf52@gmail.com

Ampara D. Palomino, Benemérita Universidad Autónoma de Puebla, Av. san Claudio y 18 Sur, Ciudad Universitaria, México, palomino@ece.buap.mx

tiene adaptada una resistencia eléctrica para calentarse y poder derretir el material a extruir. Este dispositivo cuenta con un termistor como sensor de temperatura, la señal del sensor pasa por una etapa de acoplamiento para ser digitalizada, y posteriormente ser recibida por nuestra tarjeta de adquisición, y dentro del FPGA se realice la generación de la señal de control. Pero también es recibida por nuestro software, de esta manera el usuario puede manipular la temperatura deseada y visualizar la temperatura actual de la herramienta de impresión.

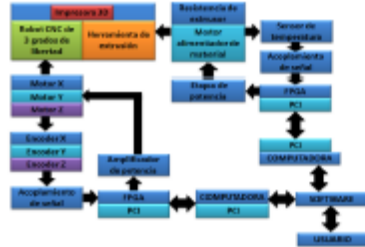


Fig. 1. Diagrama a Bloques del sistema.

Para la adquisición de datos se utilizó un FPGA Cyclone III de Altera como tarjeta hija, acoplado a una tarjeta de adquisición PCI, la cual se conecta al puerto PCI de la computadora, esto nos sirve como interfaz entre nuestro sistema físico y el software de control de nuestra impresora 3D.

Algunas características de la tarjeta de adquisición PCI son las siguientes:

- Conector header de 25 pines I/O en el banco 2.
- Puerto PCI de 32 bits en los bancos 3 y 4
- Voltaje de trabajo de 3.3V. [1]

Esta tarjeta de adquisición es una parte fundamental en la electrónica de nuestro sistema, ya que por medio de sus I/O disponibles se reciben las señales de los encoders incrementales de cuadratura, estas señales se interpretan para proporcionarnos la posición actual del robot cartesiano en cada uno de los ejes. También se reciben las señales de los sensores de temperatura tanto del extrusor como de la base caliente que son herramientas necesarias para la realización de la impresión 3D.

Por medio de esta tarjeta también se envían las señales de PWM (Pulse Width Modulation) a la etapa de potencia, que son generadas mediante el software de control y el firmware programado en el FPGA.



Fig. 2. Tarjeta Foton_1.

Para la etapa de potencia tenemos que manejar dos tipos de voltajes 3.3v que es el voltaje de las señales de control y 12v

para la alimentación de los motores. Para esto se realizó el diseño del puente H, que se muestran a continuación:



Fig. 3. Puentes H implementados.

Como se puede observar en el puente H, se utilizan transistores en configuración Darlington para garantizar la saturación de los transistores de potencia, en este caso se utilizaron los TIPs 35c y 36c, ya que pueden drenar hasta 25A, y para nuestros motores se llega a manejar hasta 15A, teniendo un buen rango de tolerancia. Además, se utilizan compuestas lógicas para el control de giro mediante la señal de PWM generada por el FPGA, también cuenta con un capacitor y una resistencia en serie, conectados en paralelo al motor para eliminar el efecto inductivo que se tiene cuando el motor es pulsado.

Como el extrusor de material y la base caliente tiene un termistor como sensor de temperatura, se debe realizar una conversión de la señal analógica a digital, para que nuestra tarjeta de adquisición pueda realizar la recepción de la señal, esto se consigue mediante un ADC0803, que nos proporciona una salida digital de 8 bits, que es suficiente para realizar el control.

En nuestra impresora 3D, utilizaremos el método de extrusión en caliente de un filamento de plástico. El proceso consiste en empujar un hilo de plástico a través de un dispositivo, denominado extrusor (Metal J-Head), que se calienta hasta una temperatura capaz de fundir ligeramente el material (actualmente se ocupa plástico ABS o PLA) que al llegar a la punta del dispositivo, es mucho más fina que el filamento original. Se va expulsando como un hilo muy fino y delgado que se va depositando en una base caliente, para que el hilo extruido se vaya quedando adherido, y así la impresora pueda ir creando la pieza capa por capa.

Las características del Metal J-Head utilizado son:

- ✓ Material : Acero inoxidable , PVC , Nylon.
- ✓ Voltaje: 12V.
- ✓ Potencia: 30w.
- ✓ Tamaño del calentador: 6 x 15mm (D * L).
- ✓ Diámetro exterior : 6 mm.
- ✓ Diámetro interior : 4 mm.
- ✓ Material: teflón.
- ✓ Termistor de 100K NTC.
- ✓ Resistencia de cartucho.



Fig. 4. Extrusor Metal J-Head de corta distancia y tubo de PTFE (teflón)

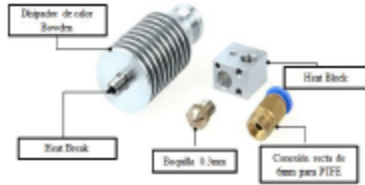


Fig. 5. Partes del extrusor J-Head

El bowden permite reducir la masa en movimiento de la extrusora, teniendo un desplazamiento controlado más rápido, menos sacudidas de la máquina, un menor uso de energía y lo más importante: una impresión más rápida.[2]

Una de las principales ventajas que presenta este J-Head, es el material del cual esta hecho (acero inoxidable), ya que le permite trabajar con policarbonato de hasta 300 grados centígrados y con materiales finuros que requieren temperaturas más alta de trabajo.

Por otro parte, se necesita un mecanismo para alimentar el extrusor, este mecanismo se coloca en una parte no móvil de la impresora 3D, el cual empuja el filamento por un tubo flexible de PTFE (teflón), que es útil por ser resbaladizo, ya que tiene poca fricción con los plásticos. Nuestro alimentador de material cuenta con un motor a pasos engranado de cambio rápido de filamento, modelo EXTV22SMBÚ marca Kysan. [3]

Tabla 1. TABLA DE CARACTERÍSTICAS DEL ALIMENTADOR.

Aditamento.	Especificación.
Angulo de paso.	1.8° ± 5%
Consumo de corriente.	1 A.
Torque	24 Kg*cm.
Relación de transmisión.	14:1
Resistencia por fase.	2.6Ω±10%

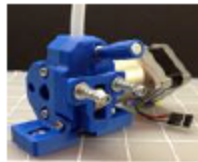


Fig. 6. Alimentador de material.

Otro dispositivo necesario para la impresión 3D es la base caliente, que permite mantener el material caliente durante la impresión y evitar el pandeo de la pieza. El modelo de nuestra base caliente es MK3, y cuenta con las siguientes características:

- > Dimensiones 214mm x 214mm.
- > 1 capa 35µm (base 1 oz) de cobre.
- > Resistencia entre 1,4 y 1,6 ohmios para el 12V.
- > Resistencia entre 5,0 y 5,4 ohmios para el 24V.
- > Calentamiento Heatbed a 100 grados en sólo 2 minutos.
- > 180 grados centígrados de temperatura máxima.
- > Planitud aprox. dentro + -0.2mm.[4]

III. DISEÑO DEL ROBOT CARTESIANO COMO IMPRESORA 3D.

Para poder realizar un control apropiado para nuestro robot, es necesario tener en cuenta el modelo dinámico del mismo, esto nos servirá para analizar su comportamiento, al utilizar diferentes algoritmos de control y de esta manera poder utilizar el que presente un mejor desempeño.

Para determinar el modelo dinámico, utilizaremos la metodología de Euler-Lagrange, la cual está ampliamente documentada.

Sabemos que, la energía total $E(q, \dot{q})$ (Hamiltoniano) de un robot manipulador de n grados de libertad es la suma de sus energías, cinética $K(q, \dot{q})$ y potencial $U(q)$.

$$E(q, \dot{q}) = K(q, \dot{q}) + U(q) \quad (1)$$

Donde $q, \dot{q} \in \mathbb{R}^n$ representan los vectores de posición, en coordenadas generalizadas y velocidad articular.

Además, el lagrangiano $L(q, \dot{q})$ de un robot manipulador de n grados de libertad se define como la diferencia entre su energía cinética $K(q, \dot{q})$ y su energía potencial $U(q)$:

$$L(q, \dot{q}) = K(q, \dot{q}) - U(q) \quad (2)$$

Por otra parte, las ecuaciones de movimiento de Euler-Lagrange para un robot manipulador de n grados de libertad, estan dadas por:

$$\frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}_i} \right] - \frac{\partial L(q, \dot{q})}{\partial q_i} = \tau_i - v(q_i, f_{e_i}) \quad (3)$$

Donde $\tau \in \mathbb{R}^n$ es el vector de pares aplicados y $v(q, f_e) \in \mathbb{R}^n$ es el vector de fuerzas o pares de fricción que depende de la velocidad articular \dot{q} y de la fricción estática f_e , que se encuentra presente en las articulaciones del mismo.

De forma equivalente:

$$\tau_i = \frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}_i} \right] - \frac{\partial L(q, \dot{q})}{\partial q_i} + b_i \dot{q}_i + f_{e_i} \text{signo}(\dot{q}_i) + f_{e_i} [1 - |\text{signo}(\dot{q}_i)|] \quad i = 1, \dots, n \quad (4)$$

Donde b, f_v son coeficientes de fricción viscosa y de Coulomb, respectivamente, f_e representa los coeficientes de fricción estática de cada una de los servomotores del robot manipulador. [5]

De esto, podemos decir que dependiendo del número grados de libertad, será el número de ecuaciones dinámicas del robot manipulador.

A continuación, se muestra la representación geométrica del robot cartesiano de 3 grados de libertad, la cual nos ayuda a visualizar el comportamiento del manipulador.

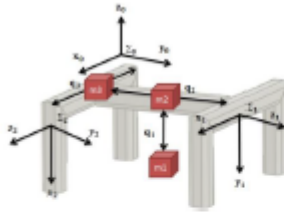


Fig. 7. Representación del robot cartesiano de 3 grados de libertad.

Para determinar el modelo de cinemática directa utilizaremos la metodología de Denavit-Hartenberg, la cual ofrece un procedimiento sencillo y cuya estructura queda en representación de transformaciones homogéneas.

La representación de Denavit-Hartenberg, se puede expresar por el producto de cuatro transformaciones básicas:

$$H_{i-1}^i = H_{R_{x_{i-1}}}(\theta_i) H_{T_{z_{i-1}}}(d_i) H_{R_{x_{i-1}}}(l_i) H_{R_{x_{i-1}}}(\alpha_i) \quad (4)$$

La transformación homogénea total se obtiene como $H_0^n = H_0^1 H_1^2 \dots H_{n-1}^n$.

De la figura 1, para obtener la tabla de parámetros de Denavit-Hartenberg, es necesario realizar una rotación auxiliar debido a que los ejes del sistema son ortogonales, a continuación, se presenta proceso realizado para abordar esta problemática. Paso 1, se realiza una rotación de $-\frac{\pi}{2}$ sobre el eje x_0 :

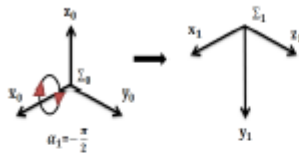


Fig. 8. Representación de la primera rotación de Denavit-Hartenberg, para obtener el sistema de referencia Σ_1 .

Paso 2, como no hay una rotación posible para que el eje z_1 se alinee al eje z_2 del sistema de referencia Σ_2 , se hace una rotación previa sobre el eje z_1 , lo que nos lleva:

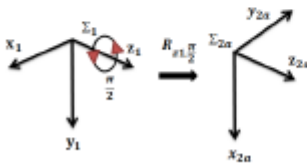


Fig. 9. Representación del sistema Σ_{2a} , después de la rotación del eje z_1 , el cual nos sirve como un sistema auxiliar para poder alcanzar la alineación deseada en el sistema Σ_2 .

Paso 3, finalmente se realiza una rotación sobre el eje x_{2a} , alcanzando la alineación del eje z_1 , con respecto al eje z_2 .

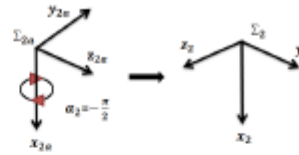


Fig. 10. Representación del sistema referenciado Σ_2 , después de la última rotación realizada sobre el eje x_{2a} , del sistema Σ_{2a} .

Con la información anterior podemos adquirir los parámetros necesarios para llenar la tabla de Denavit-Hartenberg, que nos queda de la siguiente manera:

Tabla II. TABLA DE PARÁMETROS DE DENAVIT-HARTENBERG.

Eslabón	l_i	α_i	d_i	θ_i
1	0	$-\frac{\pi}{2}$	d_1	0
2	0	$-\frac{\pi}{2}$	d_2	0
3	0	0	d_3	0

Por lo que la transformación homogénea resultante será:

$$H_0^3 = H_0^1 R_{x_0}^T H_1^2 H_2^3 = \begin{bmatrix} 0 & 0 & 1 & d_3 \\ 0 & -1 & 0 & d_2 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

De la ecuación 5, obtendremos que la cinemática directa queda de la siguiente manera:

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} d_3 \\ d_2 \\ d_1 \end{bmatrix} \quad (6)$$

Tomando en cuenta que las coordenadas $[d_3 \ d_2 \ d_1]^T$, son coordenadas cartesianas, las expresaremos en coordenadas generalizadas como $[q_3 \ q_2 \ q_1]^T$, para fines prácticos.

Del modelo de cinemática directa podemos obtener la cinemática diferencial, que es la derivada con respecto al tiempo de la cinemática directa, es decir:

$$v_i = \frac{d}{dt} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \frac{\partial f_i(l_i, q_i)}{\partial q_i} = J(q) \dot{q} \quad (7)$$

$i = 1, \dots, n$

Dónde:

v_i . Simboliza la velocidad lineal

$\partial f_i(l_i, q_i)$. Función vectorial que relaciona las coordenadas articulares q_i y las propiedades geométricas del sistema mecánico l_i , con las coordenadas cartesianas.

$J(q)$. Representa el jacobiano del robot o jacobiano analítico.

Para nuestro robot manipulador esta ecuación queda de la siguiente manera, para cada eslabón:

$$v_1 = \frac{d}{dt} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ q_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ q_1 \end{bmatrix} \quad (8)$$

$$v_2 = \frac{d}{dt} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ q_2 \\ q_1 \end{bmatrix} = \begin{bmatrix} 0 \\ q_2 \\ q_1 \end{bmatrix} \quad (9)$$

$$v_3 = \frac{d}{dt} \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_3 \\ q_2 \\ q_1 \end{bmatrix} = \begin{bmatrix} q_3 \\ q_2 \\ q_1 \end{bmatrix} \quad (10)$$

Para resolver la ecuación (2), es necesario determinar la energía cinética, que incluye el movimiento de traslación y rotación, está definida como:

$$K(\mathbf{q}, \dot{\mathbf{q}}) = \frac{m_i v_i^T v_i}{2} + \frac{I_i (\sum_j \dot{q}_j)^2}{2} = \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} \quad (11)$$

Dónde:

m_i . Simboliza el vector de las masas del sistema.

$v_i^T v_i$. Representa la rapidez lineal del centro de masa de cada eslabón.

$$v^2 = \|v\|^2 = v^T v \quad (12)$$

I_i . Es el momento de inercia del i-ésimo eslabón.

$M(\mathbf{q})$. Es la matriz de inercia del manipulador, sin embargo, para nuestro caso esta matriz es una matriz de masas constantes, ya que no tiene efecto inercial, la cual denominaremos como M.

Para nuestro caso, v^2 queda representado como:

$$v_1^2 = [0 \quad 0 \quad q_1] \begin{bmatrix} 0 \\ 0 \\ q_1 \end{bmatrix} = q_1^2 \quad (13)$$

$$v_2^2 = [0 \quad q_2 \quad q_1] \begin{bmatrix} 0 \\ q_2 \\ q_1 \end{bmatrix} = q_2^2 + q_1^2 \quad (14)$$

$$v_3^2 = [q_3 \quad q_2 \quad q_1] \begin{bmatrix} q_3 \\ q_2 \\ q_1 \end{bmatrix} = q_3^2 + q_2^2 + q_1^2 \quad (15)$$

Por otro lado, en nuestro robot cartesiano solo contamos con movimiento de traslación sobre los ejes x, y, z, de esto, podemos decir que la energía de rotación para estos ejes es igual a cero, por lo que nuestra ecuación de energía cinética quedará de la siguiente manera:

$$K(\mathbf{q}, \dot{\mathbf{q}}) = \frac{m_i v_i^T v_i}{2} = \frac{1}{2} \dot{\mathbf{q}}^T M \dot{\mathbf{q}} \quad (16)$$

$i = 1, \dots, n$

Sustituyendo los valores de las ecuaciones (13), (14) y (15), en la ecuación (16) tendremos:

$$K(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} [m_1 q_1^2 + m_2 (q_2^2 + q_1^2) + m_3 (q_3^2 + q_2^2 + q_1^2)] = \frac{1}{2} [q_1^2 (m_1 + m_2 + m_3) + q_2^2 (m_2 + m_3) + q_3^2 m_3] \quad (17)$$

Como sabemos, la energía potencial $U(\mathbf{q})$ no tiene una forma específica como en el caso de la energía cinética, depende de la geometría del robot en general, sin embargo, tiene una dependencia exclusivamente del vector de posición (\mathbf{q}) , ya que se considera su presencia a campos conservativos como la fuerza de gravedad, por lo tanto, para nuestro robot manipulador, la energía potencial solo afectará el primer eslabón, puesto que para el segundo y tercer eslabón se mantiene constante. La energía potencial se encuentra representada de la siguiente manera:

$$U(\mathbf{q}) = m_1 g q_1 \quad (18)$$

Donde, g es la constante gravitacional.

Retomando la ecuación 2, y sustituyendo los valores de la energía cinética y la energía potencial obtendremos:

$$L(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} [q_1^2 (m_1 + m_2 + m_3) + q_2^2 (m_2 + m_3) + q_3^2 m_3] - m_1 g q_1 \quad (19)$$

Como ya hemos obtenido el lagrangiano, podemos encontrar los valores de la ecuación 3, lo que nos lleva:

$$\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_1} = (m_1 + m_2 + m_3) q_1 \quad (20)$$

$$\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_2} = (m_2 + m_3) q_2 \quad (21)$$

$$\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_3} = m_3 q_3 \quad (22)$$

$$\frac{d}{dt} \left[\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_1} \right] = (m_1 + m_2 + m_3) \dot{q}_1 \quad (23)$$

$$\frac{d}{dt} \left[\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_2} \right] = (m_2 + m_3) \dot{q}_2 \quad (24)$$

$$\frac{d}{dt} \left[\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_3} \right] = m_3 \dot{q}_3 \quad (25)$$

$$\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_1} = m_1 g \quad (26)$$

$$\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_2} = 0 \quad (27)$$

$$\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_3} = 0 \quad (28)$$

Sabemos que el modelo dinámico cartesiano se encuentra dado por:

$$f_x = M_x \ddot{x} + C_x \dot{x} + g_x + f_{e_x} \quad (29)$$

Donde, $x = [x \quad y \quad z]^T$, son coordenadas cartesianas.

f_x representa la dinámica no lineal de un robot manipulador en el espacio de variables cartesianas.

M_x simboliza la matriz de inercia cartesiana.

C_x representa la matriz de fuerzas centripetas y de Coriolis cartesiana.

g_x es el vector de pares gravitacionales cartesiano.

f_{ex} simboliza el fenómeno de fricción cartesiano.

Conocemos que la cinemática diferencial y la aceleración articular se encuentran relacionadas por:

$$\dot{X} = J(q)\dot{q} \Rightarrow \dot{q} = J^{-1}(q)\dot{X} \quad (30)$$

$$\ddot{q} = J^{-1}(q)\ddot{X} + \left[\frac{d}{dt}J^{-1}(q)\right]\dot{X} \quad (31)$$

$$\ddot{X} = J^{-1}(q)\ddot{X} - J^{-1}(q)\dot{J}(q)\dot{X}J^{-1}(q)\dot{X} \quad (32)$$

En nuestro caso no se cuenta con la matriz de fuerzas centripetas y de Coriolis, ya que es igual a cero, con respecto a la matriz de inercia cartesiana nosotros contamos con una matriz de masas, mencionada con anterioridad.

Por lo tanto, el modelo dinámico cartesiano está dado por:

$$f_x = \begin{bmatrix} m_1 + m_2 + m_3 & 0 & 0 \\ 0 & m_2 + m_3 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} m_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} f_{c1} & 0 & 0 \\ 0 & f_{c2} & 0 \\ 0 & 0 & f_{c3} \end{bmatrix} \begin{bmatrix} \text{signo}(\dot{q}_1) \\ \text{signo}(\dot{q}_2) \\ \text{signo}(\dot{q}_3) \end{bmatrix} + \begin{bmatrix} f_{e1}[1 - |\text{signo}(\dot{q}_1)|] \\ f_{e2}[1 - |\text{signo}(\dot{q}_2)|] \\ f_{e3}[1 - |\text{signo}(\dot{q}_3)|] \end{bmatrix} \quad (33)$$

Este modelo puede ser igualado a τ gracias a la siguiente relación:

$$\tau = J^T(q)f_x \quad (34)$$

Sin embargo, para realizar un acoplamiento de unidades entre $\tau(Nm)$ y $f_x(N)$ se debe agregar una constante α la cual tendrá como valor 1 metro, por lo tanto, la ecuación 46 quedará de la siguiente manera:

$$\tau = \alpha J^T(q)f_x = \alpha f_x \quad (35)$$

Obteniendo el resultado de pares aplicados de la metodología de Euler-Lagrange. Siendo estos los pares que se apliquen en cada eslabon de nuestro robot.

IV. FIRMWARE DE ADQUISICIÓN.

Para la programación de nuestro FPGA EP3C10F256C6 de la familia Cyclone III, ocupamos el software Quartus II versión 13.0 web edition, realizando toda la programación en lenguaje AHDL. El firmware elaborado, tiene como parte principal el protocolo de comunicación del puerto PCI (lectura y escritura), ya que, utilizamos el FPGA para manejar el bus PCI, por medio de la tarjeta de adquisición de datos, utilizando principalmente sus 32 señales de dirección y datos. Por otro lado, el firmware también cuenta con un decoder, que nos sirve para interpretar las señales de los encoders, generando un reloj a partir de las señales A y B de cada encoder, que posteriormente será el reloj de una máquina de estados con la

que se realizamos el conteo de los pulsos de los encoders, determinando con esto la posición actual de nuestro robot cartesiano.

Nuestro robot cartesiano tenemos una frecuencia aproximada de 250KHz, de las señales de nuestros encoders, sin embargo, para poder realizar la decodificación de nuestros pulsos necesitamos una frecuencia 8 veces más rápida, por lo que manejamos frecuencias de 2 MHz, pero por medio del PCI podemos manejar velocidades de procesamiento de hasta 33MHz, por lo que no tenemos problemas con la lectura de nuestras señales, teniendo un margen para poder aumentar la velocidad de nuestros motores en futuras modificaciones.

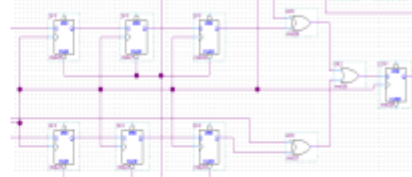


Fig. 11. Etapa de generación del pulso de reloj a partir de las señales A y B del encoder incremental de cuadratura.

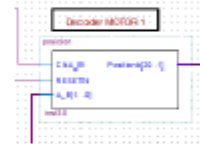


Fig. 12. Decoder.

Este decoder también determina el sentido de giro de cada una de nuestros motores, es decir, cuando se realiza un conteo positivo, el motor gira en sentido contrario de las manecillas del reloj y viceversa, teniendo la posición de casa como punto de referencia.

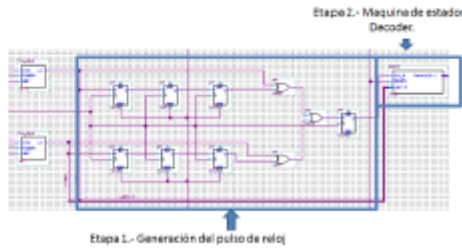


Fig. 13. Firmware de la etapa de decodificación de las señales del encoder.

También se tiene un firmware para generar la señal de PWM que se envía a nuestra etapa de potencia, con una frecuencia especial para cada motor.

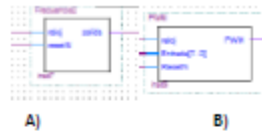


Fig. 14. A) Divisor de frecuencia. B) Generador de pulso de PWM.

El ancho de pulso dependerá de los datos enviados por el software, en el cual se realiza el control, esta señal es de 8 bits y el dato se maneja en decimal sin signo. Además, el software envía el sentido de giro mediante una señal la cual es interpretada por el FPGA para determinar por cual salida será enviada la señal de salida de PWM, de cada motor.

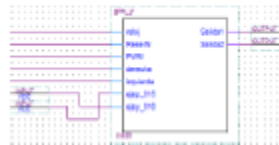


Fig. 15. Etapa donde se determina el giro y se realiza la programación para el control de los sensores de fin de carrera.



Fig. 16. Firmware para crear la señal de PWM.

Se realiza también el control de temperatura del extrusor y de la base caliente, mediante un control proporcional, y de manera análoga al control de los servomotores se utiliza una señal de PWM, para controlar la temperatura, todo esto dentro del FPGA.



Fig. 17. Diagrama a bloques del lazo cerrado de control del extrusor, que se realiza en el firmware.

Donde T_d es la temperatura deseada, T_a es la temperatura actual y E_t es el error de temperatura.

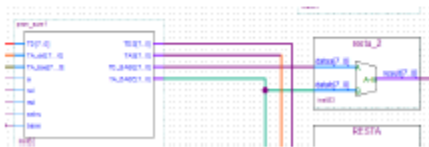


Fig. 18. Etapa 1. Lectura de temperatura deseada y actual. Para la generación del error de temperatura.

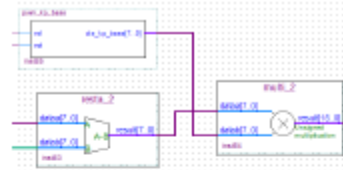


Fig. 19. Etapa 2. Se multiplica la constante proporcional por el error de temperatura.

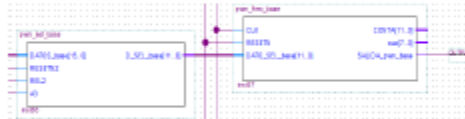


Fig. 20. Etapa 3. Se seleccionan los 8 bits más significativos para determinar el ancho de pulso de la señal y se genera la señal de salida.

Asimismo, mediante el firmware se lleva a cabo la secuencia de activación del motor a pasos del alimentador de material. Teniendo una señal de encendido para iniciar la secuencia.

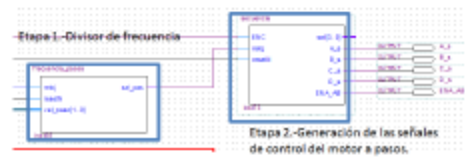


Fig. 21. Firmware para realizar la señal de control para el motor a pasos.

La frecuencia del alimentador se genera a partir de un divisor de frecuencia, en nuestro caso se propone manejar 4 frecuencias distintas de alimentación.

V. SOFTWARE DE ADQUISICIÓN.

El software del sistema se realizó en LabView. Con este programa el usuario puede escribir en el puerto PCI, los datos que necesita manejar el firmware, en este caso se manejan los 32 bits de datos del puerto. Mediante el programa el usuario puede indicar la trayectoria a seguir del robot cartesiano, pudiendo visualizar la posición actual, la posición deseada y el torque aplicado.



Fig. 22. Diagrama a bloques del funcionamiento del software.

El proceso consiste en que el usuario realice el modelo CAD de la pieza que desea imprimir, lo guarde como un archivo .stl, el cual servirá para que un programa laminador genera el código g de la pieza, posteriormente el programa de impresión que utilizamos ocupe las coordenadas generadas, como posiciones deseadas, para aplicar el control de posición y enviarlas por el bus PCI al FPGA, que producirá la señal de PWM, para realizar el control de la etapa de potencia y manipular el robot cartesiano.

Para el control de posición utilizamos el siguiente lazo de control:

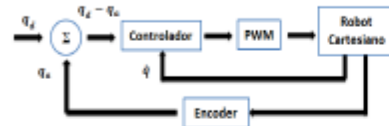


Fig. 23. Esquema del lazo de control implementado para control de posición.

Donde:

q_d es la posición deseada.

q_a es la posición actual.

$q_d - q_a$ es el error de posición.

q es la velocidad del manipulador en mm/s.

Nuestro control está dado por la siguiente ecuación:

$$\tau = Kp * \tanh(q_d - q_a) - Kv * \tanh(q) \quad (36)$$

Donde:

Kp es la ganancia proporcional del sistema.

Kv es la ganancia derivativa del sistema.

τ es la energía aplicada por el controlador, en este caso es el toque aplicado, que se encuentra en Nm.

En la siguiente figura se muestra parte de la caratula principal donde se carga el archivo a ejecutar, se visualizan las posiciones deseadas y las posiciones actuales, además de la sección de ajuste del tiempo de muestreo.



Fig. 24. Primera parte de la pantalla principal del software.

El archivo que se carga es leído y vaciado en vectores para ser utilizados por el software, el cual va tomando dato por dato para ir ejecutando el control.

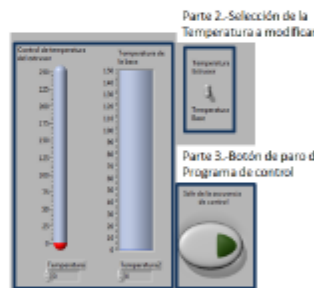
Para el control de posición son necesarios algunos datos adicionales que el usuario puede ajustar a sus necesidades. En nuestro caso, podemos modificar las ganancias proporcional y derivativa, para cada motor.



Fig. 25. Sección de parámetros del sistema.

También se pueden modificar los valores de delta para cada motor, estos parámetros son constantes que se calculan para ajustar el desplazamiento lineal de cada eje.

Con respecto, al control de temperatura tanto del extrusor, como de la cama caliente, se envía la temperatura deseada por el bus PCI para que el FPGA adquiera esa información y genere la señal de PWM.



Parte 1.-Temperaturas deseadas

Fig. 26. Segunda parte de la pantalla principal del software.

La selección de la temperatura a modificar se utiliza para realizar un multiplexeo, entre la temperatura del extrusor y la temperatura de la base, ya que solo se ocupan 8 bits del bus PCI, y para manejar ambos datos se requiere de 16 bits.

Finalmente, en el software, se pueden visualizar las gráficas de error de posición y el par aplicado, de acuerdo a los datos que se van adquiriendo del sistema y generando del control implementado.

Actualmente se está trabajando en modificaciones al algoritmo de control para manejar dos tipos de controladores uno por control de posición y otro por control de velocidad, sin embargo, aún no se tiene un algoritmo de control satisfactorio para el control de velocidad, se han implementado técnicas para optimizar el software como el método de Runge-Kutta, para obtener mejores resultados.

VI RESULTADOS EXPERIMENTALES

Para tener mejores resultados en nuestra impresión se realizó la caracterización del sensor de temperatura del extrusor, analizando su comportamiento de voltaje contra temperatura (termistor 100K NTC), de lo que obtuvimos:

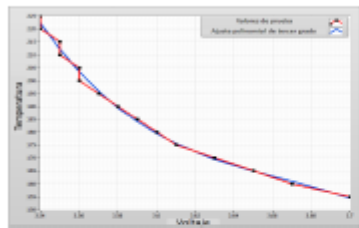


Fig. 27. Gráfica de comportamiento del termistor 100K NTC de nuestro extrusor

También se realizó la caracterización de los motores dc, para convertirlos en pseudo-servomotores de transmisión directa. Siguiendo la siguiente metodología, primero se determina el voltaje nominal de cada motor, en nuestro caso fue de 12v, después se hace una búsqueda de la frecuencia de trabajo de cada motor, la cual nos proporcione la mitad del torque máximo con un ciclo de trabajo de 50%, una vez localizada esta frecuencia se hace un barrido cambiando el porcentaje de ciclo de trabajo para poder analizar su comportamiento, que tiene que ser muy cercano al lineal. De este proceso se obtuvieron los siguientes resultados.

Tabla III. DATOS ADQUIRIDOS DEL PRIMER MOTOR A UNA FRECUENCIA DE 2.85 KHz.

Porcentaje de ciclo de trabajo de PWM	Kilogramos	Torque (Nm)
0	0	0
10	1.495	2.9302
12	1.97	3.8612
14	2.39	4.6844
16	2.665	5.2234
18	2.94	5.7624
20	3.305	6.4778
22	3.78	7.4088
24	4.03	7.8988
26	4.56	8.9376
28	4.89	9.5844
30	5.09	9.9764



Fig. 28. Gráfica de resultados del primer motor.

Se caracterizó hasta el 30% de ciclo de trabajo, ya que el servomotor nos proporciona el torque necesario para nuestra aplicación.

Para el segundo motor se obtuvieron los siguientes resultados a una frecuencia de 2.5KHz:

Tabla IV. DATOS DEL SEGUNDO MOTOR.

Ciclo de trabajo de PWM	Kilogramos	Torque (Nm)
0	0	0
10	1.69	3.3124
12	2.19	4.2924
14	2.605	5.1058
16	2.96	5.8016
18	3.235	6.3406
20	3.48	6.8208
22	3.92	7.6832
24	4.325	8.477
26	4.75	9.31
28	5.1	9.996
30	5.62	11.0152
32	6.02	11.7992
34	6.3	12.348



Fig. 29. Gráfica de resultados del segundo motor.

Con respecto, a los resultados anteriores se observa que el servomotor cumple con el torque suficiente para desplazar el eje z.

Finalmente, para el tercer motor se encontró que la mejor frecuencia de trabajo fue de 75Hz, obteniendo los siguientes resultados:

Tabla V. DATOS OBTENIDOS DEL TERCER MOTOR A 75Hz.

Ciclo de trabajo de PWM	Kilogramos	Torque (Nm)
0	0	0
10	0.165	0.0616
20	0.92	0.3435
30	1.76	0.6571
40	2.25	0.8401
50	3.35	1.2508
60	3.85	1.4375
70	5.07	1.8930
80	5.55	2.0722
90	6.88	2.5688

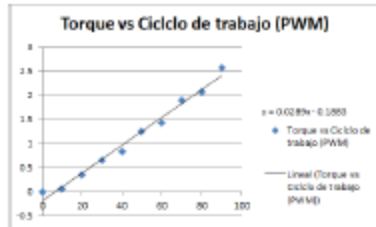


Fig. 30. Gráfica de resultados del tercer motor.

Con los resultados anteriores, se comprueba que el servomotor tendrá el torque suficiente para desplazar la herramienta sobre el eje z.

Con respecto, al algoritmo de control se realizaron pruebas para comprobar su funcionamiento, en este caso le indicamos al motor del eje "y" que se moviera 80mm, graficando los errores de posición, de lo cual se obtuvo:

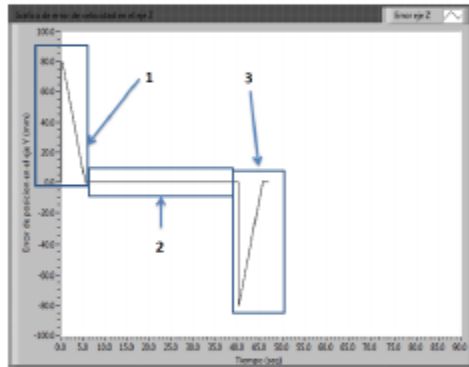


Fig. 31. Gráfica del error de posición del eje y

1. Se puede observar en la gráfica anterior como el robot cartesiano se va desplazando a la posición deseada que son 80mm, llegando en un periodo aproximando de 5.5s.
2. Se mantiene en la posición deseada alrededor de 30s por lo que el error de posición es igual a cero en ese periodo de tiempo.
3. Finalmente regresa a la posición inicial por lo que el error es negativo, este movimiento se realiza en un tiempo de 5.5s, al igual que el primero.

Nuestro algoritmo de control al tener una tangente hipérbolica nos permite proporcionar todo el torque rápidamente para llegar a la posición deseada lo más pronto posible, con ayuda de una buena sintonización de nuestras ganancias.

Por otro lado, se comprobó que el firmware del control del motor a pasos que se ocupa como alimentador realice la secuencia apropiadamente a la frecuencia deseada. A continuación se muestra la simulación de firmware implementado.

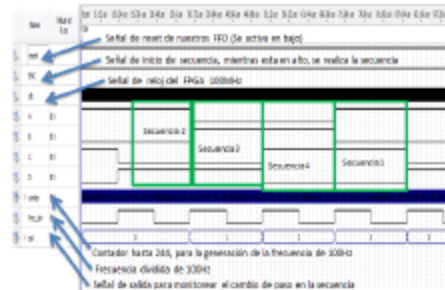


Fig. 32. Diagrama de tiempos del firmware de control del alimentador.

También se realizaron pruebas de funcionamiento del firmware del control de temperatura, monitoreando la señal de PWM que se generaba a la frecuencia deseada.

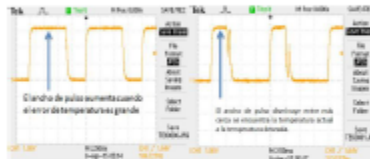


Fig. 33. Señal de PWM, generada por firmware del control de temperatura.

Como se muestra en las figuras anteriores, se muestra el funcionamiento del control de temperatura, ya que al tener un error grande en la temperatura el ancho de pulso aumenta, y disminuyendo gradualmente el ancho de pulso al reducirse el error de temperatura.

VII. CONCLUSIONES.

A diferencia de los robots cartesianos comerciales, este manipulador utiliza el puerto puerto PCI como interfaz, por medio de la tarjeta de adquisición que tiene un FPGA, lo cual le proporciona grandes beneficios como velocidad de procesamiento de 100MHz, 168 I/O digitales y arquitectura abierta, en comparación con otros sistemas empujados como Arduino Mega que manejan 54 I/O digitales y una velocidad de reloj de 16MHz [6]. Además, la programación del FPGA, es muy versátil y nos permite realizar varios procesos de control en tiempo real, que es muy importante para cualquier aplicación.

Al utilizar LabVIEW para realizar nuestro programa, fue más sencillo poder manipular el puerto PCI de la computadora, sin embargo, cabe resaltar, que hay que tener cuidado con las direcciones que se ocupan y la información que se envía porque podemos causar problemas con el sistema. Este programa nos permite también modificar fácilmente los algoritmos de control y visualizar los datos que deseamos monitorear.

Teniendo el modelo dinámico de nuestro robot manipulador, pudimos realizar simulaciones para comprobar el comportamiento del sistema empleando nuestro algoritmo de control, el cual tiene un alto desempeño. Por otro lado, si igualamos la ecuación (33), con la ecuación (36), y

desarrollando la matemática pertinente, proponiendo una función candidata de Lyapunov podemos demostrar que el punto de equilibrio es único y global. Lo cual es muy importante para el control de nuestro manipulador.

Con los resultados experimentales obtenidos podemos decir que con las frecuencias propuestas para el manejo de los servomotores tendremos un comportamiento similar al de un servomotor de transmisión directa, sirviendo esto de base para nuestra aplicación. De la caracterización de la resistencia del extrusor, podemos concluir que tendremos un comportamiento lineal con la frecuencia seleccionada, siendo importante para el proceso de impresión 3D, ya que se tienen diferentes tipos de materiales para la impresión y temperaturas de extrusión.

Como trabajos a corto plazo, una vez resuelto lo del algoritmo de control de velocidad, tenemos que realizar pruebas de tirado de material, para encontrar la temperatura óptima de trabajo y la velocidad más apropiada para realizar una buena impresión 3D.

VIII. REFERENCIAS.

- [1] G.H. Bezaros, "Robot CNC multiherramienta", (Tesis de maestría inédita). Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla.
- [2] Extrusor J-Head, Noviembre del 2014, disponible en: <http://www.ebay.com/itm/231225959977?ru=http%3A%2F%2F>
- [3] Motorreductor Kyan, Noviembre del 2014, disponible en: <http://www.kysanelectronics.com/Products/Detail.php?recordID=8072>.
- [4] Heatbed, Noviembre del 2014, disponible en: http://reprap.org/wiki/PCB_Heatbed
- [5] F. Reyes, "Robótica Control de robots manipuladores", Ed. Alfa omega, Primera edición.
- [6] <https://www.arduino.cc/en/Main/arduinoBoardMega>

Currículo corto de los autores.

A. López. Ingeniero Mecatrónico egresado de la Benemérita Universidad Autónoma de Puebla en 2010. Actualmente en la Maestría en Ciencias de la Electrónica Op. Automatización.

S. Vergara. Licenciado en Electrónica egresado de la Benemérita Universidad Autónoma de Puebla en 1994. Obtuvo el grado de Maestría en Ciencias con Especialidad en Optoelectrónica en 1997 y Doctorado en el año 2000, en la Benemérita Universidad Autónoma de Puebla, actualmente es Profesor Investigador, Titular A en la Facultad de Ciencias de la Electrónica de la BUAP, Miembro del SNI nivel I.

M.A. Vargas. Licenciada en Electrónica egresada de la Benemérita Universidad Autónoma de Puebla en 1994. Obtuvo el grado de Maestría en Ciencias con Especialidad en Optoelectrónica en 1997 y Doctorado en el año 2000, en la Benemérita Universidad Autónoma de Puebla, actualmente es Profesor Investigador, Titular A en la Facultad de Ciencias de la Electrónica de la BUAP, Miembro del SNI nivel I.

F. Reyes. Licenciado en Electrónica egresado de la Benemérita Universidad Autónoma de Puebla en 1984. Obtuvo el grado de Maestría en Ciencias con Especialidad en Electrónica, en el Instituto Nacional de Astrofísica, Óptica y Electrónica en 1990. En el Centro de Investigación Científica y de Educación Superior de Ensenada, realizó el Doctorado en

Ciencias con Especialidad en Electrónica y Telecomunicaciones en 1997.

A. D. Palomino. Doctor en Tecnologías de la Información y Sistemas, especialidad Sistemas de Control. Profesor Investigador en la Facultad de Ciencias de la Electrónica. BUAP

Apéndice C

TOEFL ITP Score Report

Name of Institution:	FACULTAD DE LENGUAS DE LA BUAP	Student Number:	214470677				
Name:	LOPEZ A. FREDO	Sex:	M	Degree:	None	Times Taken TOEFL:	None
DOB:	04/23/1987	Listening Comprehension:	45	Test Date:	05/20/2016		
Native Country:	Mexico	Structure & Written Expression:	39	Form:	TOEFL ITP		
Native Language:	Spanish	Reading Comprehension:	50	Total Score:	447		
Sealed Scores:							

ETS TOEFL ITP

The face of this document has a security background. The back contains a watermark. Hold at an angle to view.
The TOEFL ITP Assessment Series is designed to be used for placement, progress monitoring, and exit purposes. TOEFL ITP scores can also be used for admissions to programs and institutions where English is not the dominant language of instruction for content courses. Learn more at www.ets.org/toefl_ip/usa.

10616-10271 - 09/17/09-000 - Printed in U.S.A. I.N. 770462

**Student's File Copy
Do Not Copy**

Copyright © 2013 by Educational Testing Service.