



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA

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

**“IMPLEMENTACIÓN DE UN CNC CON 3 GRADOS DE
LIBERTAD APLICADO A IMPRESIÓN 3D” ***

T E S I S

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

Presenta:

Ing. Carolina Jiménez Martínez**

Directores:

Dra. Ma. Aurora D. Vargas Treviño (FCE-BUAP)

Dr. José Fernando Reyes Cortés (FCE-BUAP)

Dr. Sergio Vergara Limon (FCE-BUAP)

Puebla, México

Diciembre 2019

* Trabajo financiado por proyecto VIEP.

** Becario CONACYT.

BUAP[®]

Agradecimientos

Agradezco a Dios por la vida que me ha regalado, por las oportunidades que me ofrece cada día para seguir superándome y por darme la fortaleza necesaria para permitirme culminar una meta más en mi área profesional.

A mis padres Gloria y Martín, por el apoyo incondicional que siempre me brindan en cada paso que doy, por guiarme hacia el camino correcto a través de sus valores y sabios consejos que me han permitido ser una persona de bien y sobre todo por ser el pilar fundamental en cada aspecto de mi vida.

A mis hermanos Martín y Miguel, porque a través de ellos he aprendido valiosas lecciones de vida.

A mi novio Paco, por el apoyo moral e intelectual que siempre me brinda incluso a la distancia y por ser parte importante en esta nueva etapa de mi vida.

A mis amigos y compañeros de maestría, por compartir sus conocimientos, alegrías y tristezas durante este trayecto y que juntos logramos culminar esta etapa.

A mis asesores de tesis, Dra. Aurora y Dr. Sergio, por guiarme en este camino para lograr concluir este trabajo, por todos los conocimientos y el apoyo brindado.

A los que formaron parte de mi jurado revisor, Dr. José Eligio Moisés Gutiérrez Arias, Dra. Josefina Castañeda Camacho y Dr. César Martínez Torres, por las observaciones y comentarios que contribuyeron en el desarrollo y culminación de este proyecto.

A la coordinadora, Dra. Olga Guadalupe Félix Beltrán, por el apoyo incondicional que brinda a los alumnos para realizar los trámites debidos en tiempo y forma.

A la Benemérita Universidad Autónoma de Puebla, por hacer lo necesario para contar con posgrados de calidad y en especial a la Facultad de Ciencias de la Electrónica por darme la oportunidad de ser parte de este posgrado y que a través de su infraestructura y sus excelentes docentes me permitieron concluir mis estudios.

A CONACYT, por el apoyo económico que ofrece mediante las becas para que más personas se sigan preparando profesionalmente.

A VIEP por financiar este proyecto.

Dedicatoria

A mis padres Gloria y Martín, a mis hermanos Martín y Miguel, y a mi Paco, porque con el amor y apoyo de la familia todo es posible.

Resumen

Partiendo de un robot cartesiano de 3 grados de libertad tipo CNC desarrollado en la Maestría en Ciencias de la Electrónica de la Benemérita Universidad Autónoma de Puebla, se presenta el desarrollo de hardware, firmware y software necesarios para llevar a cabo el control de dicho robot mediante una tarjeta FPGA con microprocesador embebido, de modo tal, que pueda funcionar como impresora 3D y de acuerdo con sus componentes y estructura mecánica sea comparada con las impresoras 3D industriales. Además, para tener comunicación con el robot se utiliza una interfaz de usuario desarrollada en LabView, la cual se comunica mediante protocolo Wi-Fi, esta permite la descarga del código de control al microprocesador, así como la lectura y escritura de datos.

Se presenta el estado del arte de impresoras 3D de nivel industrial y de sistemas embebidos como tarjetas de desarrollo con FPGA. De igual manera, se describen las características y especificaciones que posee el robot, como distancias de desplazamiento, resolución y área de impresión, por mencionar algunas, y se estudia su cinemática directa y modelo dinámico.

Por otra parte, se describen los componentes mecánicos y electrónicos, así como el funcionamiento que tendrán para complementar el diseño del robot y permitirle de esta manera funcionar como impresora 3D. Con esto, se procede a la metodología implementada para la caracterización de los distintos componentes implementados, como el termistor del extrusor y la base que les permitirá calentarse a determinada temperatura, los adc para convertir las señales analógicas a digital y poder interpretarlas fácilmente; esto con el fin de desarrollar sus etapas de potencia, de modo tal, que logren drenar la corriente suficiente para mantener controlada la planta.

De igual manera, se presenta un compilador desarrollado en LabView que permitirá obtener las posiciones deseadas, las cuales se obtienen ya sea de una secuencia de posiciones asignada manualmente o del código G de una pieza 3D extraído de un software laminador, convirtiendo dichas posiciones a código hexadecimal para que el microprocesador pueda interpretarlas.

Finalmente, se presentan los algoritmos de control implementados, para lo cual se utilizó control con acción acotada como lo es la función tangente hiperbólica, ya que permitirá a los actuadores trabajar en la parte lineal. Logrando como resultados, poder alcanzar una precisión de hasta 10 micras en el eje Z.

ÍNDICE

Introducción.....	viii
Capítulo 1. Descripción del sistema mecatrónico	1
1.1 Características generales.....	1
1.2 Especificaciones.....	3
1.3 Resolución real del robot CNC.....	4
1.4 Diagrama general.....	5
1.5 Conclusiones.....	6
Capítulo 2: Modelo Cinemático y Dinámico.....	7
2.1 Cinemática directa	7
2.2 Dinámica.....	10
2.3 Conclusiones.....	12
Capítulo 3: Hardware.....	13
3.1 Componentes para el sistema mecánico	13
3.1.1 Riel de alta precisión y carro lineal de precisión	13
3.1.2 Tornillo embalado	14
3.1.3 Interruptores de fin de carrera	14
3.1.4 Cadenas porta cables	15
3.1.5 Motores DC y reductores	15
3.1.6 Encoders.....	17
3.1.7 Herramienta de extrusión	17
3.2 Electrónica de potencia.....	22
3.2.1 Etapa de potencia para los motores DC	22
3.2.2 Circuito para el acoplamiento de señales de los encoders	22
3.2.3 Etapa de potencia para el extrusor de material.....	23
3.2.3.1 Caracterización del extrusor	25
3.2.3.2 Regulador de voltaje.....	29
3.2.3.3 ADC0804 para el extrusor de material	30
3.2.4 Etapa de potencia para la cama caliente.....	32

3.2.4.1 Caracterización de la cama caliente.....	33
3.2.4.2 ADC0804 para la cama caliente	35
3.2.5 Alimentador de material.....	37
3.2.6 Interruptores de fin de carrera	38
3.2.7 Módulo Wi-Fi.....	38
3.2.8 Tarjeta de implementación.....	39
3.3 Conclusiones.....	40
Capítulo 4: Sistema de control	41
4.1 Firmware.....	41
4.1.1 Microprocesador.....	41
4.1.1.1 Administrador Wi-Fi	43
4.1.2 Administrador SDRAM	43
4.1.3 Decodificador de señales.....	43
4.1.4 Convertidor para ADC	45
4.1.5 Interruptores de proximidad.....	47
4.1.6 Generador PWM	50
4.1.7 Secuencia de pasos para el control de velocidad del motor a pasos	52
4.2 Software	57
4.2.1 Lazo de control.....	57
4.2.1.1 Control de posición.....	57
4.2.1.2 Control de velocidad.....	59
4.2.1.3 Control de temperatura	60
4.2.2 Software de usuario.....	62
4.2.2.1 Compilador de posiciones deseadas.....	63
4.2.2.2 Interfaz de usuario.....	67
4.3 Conclusiones.....	70
Capítulo 5: Resultados experimentales	71
5.1 Estructura mecánica de la impresora 3D	71
5.2 Control de posición.....	71
5.3 Control de velocidad.....	74
5.4 Control de temperatura	76

5.5 Impresión 3D	78
5.5.1 Impresión 3D con control de velocidad	80
5.5.2 Impresión 3D con control de posición	82
5.5.3 Tabla comparativa de impresoras 3D industriales	83
Conclusiones generales.....	84
Apéndices	86
Apéndice A: Constancia de Ponencia.....	86
Apéndice B: Artículo publicado	87
Apéndice C: Firmware embebido en la FPGA	94
Bibliografía.....	95

ÍNDICE DE FIGURAS

Figura 1. Tipos de articulaciones de un robot [2].....	x
Figura 2. Configuración de los robots manipuladores y su espacio de trabajo [3].....	xi
Figura 3. Máquina textil de Joseph Jacquard programada mediante tarjetas perforadas.	xii
Figura 4. Fresadora Cincinnati.	xii
Figura 5. Primera impresora 3D comercial SLA-1.....	xiii
Figura 6. Impresora 3D: DT 600+.....	xiv
Figura 7. Impresora 3D: Industry F340.....	xiv
Figura 8. Impresora 3D: BigRep Studio.....	xiv
Figura 9. Arquitectura básica de un FPGA.	xvi
Figura 10. Tarjeta de desarrollo con FPGA Cyclone III.	xvii
Figura 11. Tarjeta de desarrollo DE0-CV con FPGA Cyclone V de Intel.	xviii
Figura 12. Robot CNC de 3 gdl.....	1
Figura 13. Diagrama general del funcionamiento del sistema inicial.	3
Figura 14. Diagrama general del robot CNC con 3 gdl aplicado a impresión 3D.....	6
Figura 15. Representación del robot CNC de 3 grados de libertad.	7
Figura 16. Primera rotación Denavit-Hartenberg para obtener el sistema de referencia $\Sigma 1$. .	8
Figura 17. Rotación auxiliar $\Sigma 2a$ para alcanzar la alineación deseada en el sistema $\Sigma 2$	8
Figura 18. Sistema de referencia $\Sigma 2$, después de la última rotación realizada sobre el eje x_{2a} del sistema $\Sigma 2a$	8
Figura 19. Riel de alta precisión y carro de precisión.	13
Figura 20. Tornillo embalado.	14
Figura 21. Interruptor de fin de carrera.	14
Figura 22. Robot CNC con 3 gdl: a) Robot inicial, b) Robot recableado.	15
Figura 23. Motor Z5D120-12 con reductor 5GU9K.	15
Figura 24. Motor Z4D40-12GN-30S con reductor 4GN15K.	16
Figura 25. Encoder giratorio: a) E6B2-CWZ6C, b) Conexión para encoder con salida NPN colector abierto.	17
Figura 26. Porta carrete para rollo de filamento.....	18
Figura 27. Extrusor de material: a) Metal J-Head, b) Partes del extrusor.	18
Figura 28. Motor a pasos bipolar con impulsor de filamento y configuración de bobinas. .	20
Figura 29. Cama caliente MK3.....	21
Figura 30. Diagrama esquemático del puente H.	22
Figura 31. Diagrama esquemático para las señales de los encoders.	23
Figura 32. Señales de cuadratura de los canales A y B.	23
Figura 33. Circuito de potencia para el extrusor de material.....	24
Figura 34. Circuito divisor de voltaje para el extrusor.....	25
Figura 35. Gráfica de linealidad del extrusor.	26

Figura 36. Circuito amplificador para el termistor del extrusor de material.	27
Figura 37. Gráfica de voltaje medido contra temperatura con curva de ajuste logarítmica.	28
Figura 38. Gráfica de voltaje medido contra temperatura con curva de ajuste potencial.	28
Figura 39. Módulo regulador de voltaje LM2596.	29
Figura 40. Conexión para los LM2596: a) Voltaje de salida 5V, b) Voltaje de salida -5V.	29
Figura 41. Conexión en modo continuo para el ADC0804 del extrusor.	31
Figura 42. Gráfica de linealidad del ADC0804 del extrusor.	32
Figura 43. Circuito de potencia para la cama caliente.	32
Figura 44. Circuito divisor de voltaje para la cama caliente.	34
Figura 45. Gráfica de linealidad de la cama caliente.	35
Figura 46. Gráfica de voltaje medido contra temperatura de la cama caliente.	35
Figura 47. Conexión en modo continuo para el ADC0804 de la cama caliente.	36
Figura 48. Gráfica de linealidad del ADC0804 de la cama caliente.	36
Figura 49. Circuito integrado L298.	37
Figura 50. Arreglo de señal para motor a pasos: a) Señal sin diodos, b) Señal con diodos.	37
Figura 51. Diagrama esquemático del circuito para los interruptores de fin de carrera.	38
Figura 52. Módulo Wi-Fi.	38
Figura 53. Tarjeta de desarrollo DE0-CV: a) Vista superior, b) Vista inferior.	39
Figura 54. Diagrama general del firmware.	41
Figura 55. Combinaciones de las señales de cuadratura.	44
Figura 56. Diagrama de estados para la lectura de las señales A y B del encoder.	44
Figura 57. Bloque general para la decodificación de los encoders.	45
Figura 58. Firmware para generar la señal de reloj y el conteo para la máquina de estados.	45
Figura 59. Bloque general para el acondicionamiento de las señales de los ADC0804.	46
Figura 60. Conversión de las señales de entero a punto flotante.	46
Figura 61. Diagrama de flujo para almacenar las señales de los ADC0804.	47
Figura 62. Bloque general para el acondicionamiento de los interruptores.	48
Figura 63. Firmware para la lectura y conversión de las señales de los interruptores.	48
Figura 64. Diagrama de estados para la lectura de los interruptores.	49
Figura 65. Circuito de protección para interruptores de fin de carrera.	50
Figura 66. Bloques de control mediante PWM.	50
Figura 67. Firmware para el control de la etapa de potencia mediante PWM.	51
Figura 68. Modelo conceptual del motor a pasos bipolar.	53
Figura 69. Diagrama de flujo de la secuencia para controlar el giro del alimentador de material.	54
Figura 70. Bloque para el control del motor a pasos.	55
Figura 71. Firmware para el control del motor a pasos.	55
Figura 72. Señales de la secuencia de pasos del alimentador de material.	56
Figura 73. Lazo cerrado para el control de posición: a) Eje Z, b) Ejes X y Y.	57
Figura 74. Diagrama de flujo para el control de posición del eje X.	58

Figura 75. Diagrama de flujo para el control de posición del eje Y.....	59
Figura 76. Diagrama de flujo para el control de posición del eje Z.	59
Figura 77. Lazo cerrado para el control de velocidad de los ejes X y Y.....	60
Figura 78. Lazo cerrado para el control de temperatura.....	61
Figura 79. Lazo cerrado para el control de temperatura del extrusor y la base.....	62
Figura 80. Interfaz gráfica del compilador de posiciones deseadas.	62
Figura 81. Interfaz de usuario para controlar el sistema mecatrónico.....	63
Figura 82. Proceso previo a la impresión en 3D.....	63
Figura 83. Ejemplo de modelo en 3D diseñado en SolidWorks.....	64
Figura 84. Modelo 3D en formato <i>.stl</i>	64
Figura 85. Modelo 3D laminado.	65
Figura 86. Código G del modelo 3D laminado.	65
Figura 87. Compilador para obtener posiciones deseadas.....	66
Figura 88. Ventana para seleccionar el archivo a compilar.....	66
Figura 89. Posiciones deseadas en formato hexadecimal.....	67
Figura 90. Ventana para descargar el software de control.	68
Figura 91. Ventana para descargar posiciones deseadas a la RAM.	68
Figura 92. Ventana para inicializar parámetros.....	69
Figura 93. Ventana para enviar comandos al microprocesador.....	69
Figura 94. Ventana para la lectura de datos.....	70
Figura 95. Sistema mecatrónico: a) Antes, b) Después.....	71
Figura 96. Control de posición en el eje X.	72
Figura 97. Control de posición en el eje Y.....	73
Figura 98. Control de posición en el eje Z.	74
Figura 99. Control de velocidad en el eje X.	75
Figura 100. Control de velocidad en el eje Y.....	76
Figura 101. Control de temperatura de la base a 60° C.....	77
Figura 102. Control de temperatura del extrusor a 200° C.....	77
Figura 103. Diagrama de flujo para impresión 3D.....	78
Figura 104. Secuencia 3D para leer posiciones.....	79
Figura 105. Boceto de malla como ejemplo.	80
Figura 106. Impresión de malla a 40 mm/s.	80
Figura 107. Impresión de malla a 30 mm/s.	81
Figura 108. Impresión de malla a 20 mm/s.	81
Figura 109. Impresión de malla con control de posición (2 capas).....	82
Figura 110. Impresión de malla con control de posición: a) 5 capas, b) 10 capas.	82

ÍNDICE DE TABLAS

Tabla 1. Evolución de los dispositivos electrónicos.....	xvi
Tabla 2. Características generales del robot.	2
Tabla 3. Tabla de parámetros de Denavit-Hartenberg.....	9
Tabla 4. Características del motor Z5D120-12.	16
Tabla 5. Características del reductor 5GU9K.....	16
Tabla 6. Características del motor Z4D40-12GN-30S.	16
Tabla 7. Características del reductor 4GN15K.....	16
Tabla 8. Características del encoder E6B2-CWZ6C.....	17
Tabla 9. Especificaciones del extrusor de material Metal J-Head.....	19
Tabla 10. Características del alimentador de material.....	20
Tabla 11. Características del MK3.	21
Tabla 12. Datos obtenidos al variar el ciclo de trabajo del extrusor.	26
Tabla 13. Datos del voltaje amplificado.....	27
Tabla 14. Cálculo del error mediante mínimos cuadrados.	29
Tabla 15. Función de los pines del ADC0804.....	30
Tabla 16. Datos obtenidos al variar el ciclo de trabajo de la cama caliente.....	34
Tabla 17. Características del módulo Wi-Fi.....	39
Tabla 18. Instrucciones del microprocesador en ensamblador.....	42
Tabla 19. Variables de las instrucciones MIPS.	42
Tabla 20. Frecuencia de trabajo de los actuadores.	51
Tabla 21. Secuencia para controlar el giro del motor a pasos bipolar.....	53
Tabla 22. Posiciones deseadas en X, Y, Z.....	72
Tabla 23. Ganancias para control de posición en X, Y, Z.....	72
Tabla 24. Ganancias para control de velocidad en X y Y.	75
Tabla 25. Ganancias para control de temperatura.	76
Tabla 26. Comparación de impresoras 3D industriales.....	83

Introducción

Hoy en día, la tecnología se ha vuelto parte crucial y, hasta cierto modo, se puede ver como algo común en la vida de los seres humanos. Dándole el enfoque en el área de la Robótica, esta se encuentra relacionada en sí con el deseo de sintetizar algunos aspectos de la función humana mediante el uso de mecanismos, sensores, actuadores y computadoras, de modo que puedan realizar una extensa variedad de funciones según el sector en el que se quieran implementar. Debido a esto, se le puede encontrar, por ejemplo, en robots para tareas domésticas, en centros de investigación, hospitales con quirófanos robotizados, en universidades e incluso en fábricas totalmente automatizadas mediante robots.

La robótica es un campo relativamente nuevo en la tecnología moderna. Dentro de los tipos de robots están los robots manipuladores, que se encuentran mayormente en lo que es el área industrial ya que ofrecen ventajas considerables por el hecho de que pueden realizar tareas difíciles o peligrosas para el ser humano, así como también pueden realizar dichas tareas en menor tiempo y con mayor calidad. Dentro de los robots manipuladores se encuentra el robot cartesiano, el cual es uno de los dispositivos de mayor uso en las máquinas automatizadas gracias a que puede generar trayectorias complejas en un espacio tridimensional a partir de movimientos lineales independientes. De esta forma, pueden ser empleados para manipular objetos o en máquinas-herramienta de control numérico computarizado (CNC) como las cortadoras láser, perforadoras de placas, soldadura y centros de mecanizado, para realizar trabajos de pick & place (recoger y colocar), ensamblaje, corte, carga o incluso como impresora 3D, entre otras.

Las impresoras 3D se han vuelto muy populares hoy en día, gracias a las ventajas que ofrecen al poder realizar réplicas de modelos diseñados por ordenador convirtiéndolos en piezas 3D de distintos tamaños y formas según la aplicación para la cual se requieran. Debido a la versatilidad de este tipo de máquinas se les puede emplear casi en cualquier sector, por ejemplo, en los hospitales se ha empleado esta técnica de impresión 3D para la fabricación de órganos mediante materiales especiales, de igual manera, se han desarrollado impresoras 3D para construir pequeñas casas de concreto.

Por otra parte, los sistemas embebidos son tan comunes que se encuentran disponibles casi en cualquier dispositivo a nuestro alrededor, desde teléfonos celulares hasta automóviles, así como también en equipos de audio, aviones, trenes, electrodomésticos o controladores de procesos en sistemas de producción industrial. La implementación de sistemas embebidos en el diseño de algún dispositivo, por lo general se orienta a reducir costos, reducir tamaño y maximizar la confiabilidad.

Ya que los sistemas embebidos están diseñados para realizar tareas específicas, para este proyecto de tesis se desarrollará un sistema de control para un robot cartesiano que tendrá la función de una máquina CNC para realizar impresión en 3D mediante un sistema embebido; de modo que con este sistema se logren controlar los 3 grados de libertad (gdl) y la temperatura, tanto de la cama caliente como del extrusor de material, al igual que hacer monitoreo y diagnósticos del sistema en tiempo real.

La robótica es una disciplina científica que aborda la investigación y desarrollo de una clase particular de sistemas mecánicos, denominados robots, diseñados para realizar una amplia variedad de aplicaciones industriales, científicas, domésticas y comerciales. La naturaleza multidisciplinaria de la robótica permite involucrar una gran cantidad de áreas del conocimiento tales como matemáticas, física, electrónica, computación, visión e inteligencia artificial, entre otras [1].

El término **robot** fue introducido por primera vez por el escritor checo Karel Capek en el año 1920 cuando estrenó su obra *Rossum's Universal Robots*, y proviene de la palabra checa **robot** que significa *esclavo* o *trabajo forzado*. A partir de entonces, el término se ha aplicado a casi cualquier sistema mecánico, es decir, prácticamente a todo lo que puede operar con cierto grado de autonomía se le ha llamado *robot*.

Una mejor definición asignada para el término *robot* es la definición oficial aportada por el *Robot Institute of America (RIA)* el cual lo define de la siguiente manera:

“Un robot es un manipulador multifuncional reprogramable diseñado para mover material, partes, herramientas, o dispositivos especializados a través de movimientos programados para el desempeño de una variedad de tareas”.

Esta definición, ligeramente modificada, ha sido adoptada por la Organización Internacional de Estándares (ISO) que define al robot industrial como:

“Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas” [2].

Sin embargo, fue el escritor Isaac Asimov el mayor impulsor de la palabra *robot*, al escribir varias novelas tratando sobre este tema y una en especial donde por primera vez enuncia las famosas *Tres Leyes de la Robótica*:

1. *Un robot no puede hacer daño a un ser humano o, por inacción permitir que un ser humano sufra daño.*
2. *Un robot debe obedecer las órdenes recibidas por un ser humano, excepto si tales órdenes entran en conflicto con la primera ley.*
3. *Un robot debe proteger su propia existencia mientras tal protección no entre en conflicto con la primera o la segunda ley.*

Entre los robots considerados de mayor utilidad en la actualidad se encuentran los robots industriales o manipuladores. Estos están destinados a realizar de forma automática determinados procesos de fabricación o manipulación, por lo que se utilizan principalmente en la fabricación industrial.

Los principales factores que determinaron la difusión de la tecnología robótica en una gama cada vez más amplia de aplicaciones en la fabricación industrial son la reducción de los costos de fabricación, el aumento de la productividad, la mejora de normas de calidad del producto y, por último, pero no menos importante, la posibilidad de eliminar tareas perjudiciales o desagradables para el operador humano en un sistema de fabricación.

Por su significado habitual, el término *automatización* denota una tecnología destinada a reemplazar a los seres humanos con máquinas en un proceso de fabricación, en

lo que respecta a no solo la ejecución de operaciones físicas sino también el procesamiento inteligente de información sobre el estado del proceso [3].

El robot industrial es una máquina con características significativas de versatilidad y flexibilidad. La estructura mecánica de un robot manipulador consta de una secuencia de cuerpos rígidos interconectados por medio de articulaciones; un manipulador se caracteriza por un brazo que asegura la movilidad, una muñeca que confiere destreza, y un efector terminal que realiza la tarea requerida del robot.

Los manipuladores consisten en vínculos casi rígidos, los cuales están conectados por articulaciones que permiten el movimiento relativo de los vínculos adyacentes. Estas articulaciones generalmente se instrumentan con sensores de posición, los cuales permiten medir la posición relativa de los vínculos adyacentes. En el caso de las articulaciones giratorias o angulares, estos desplazamientos se conocen como *ángulos articulados*. Algunos manipuladores contienen articulaciones deslizantes (o prismáticas), en las que el desplazamiento relativo entre los vínculos es una traslación, algunas veces llamada *desplazamiento de articulación* [4].

El movimiento de cada articulación puede ser de desplazamiento, de giro o de una combinación de ambas. De este modo, son posibles 6 tipos diferentes de articulaciones con sus distintos grados de libertad, como se muestra en la figura 1.

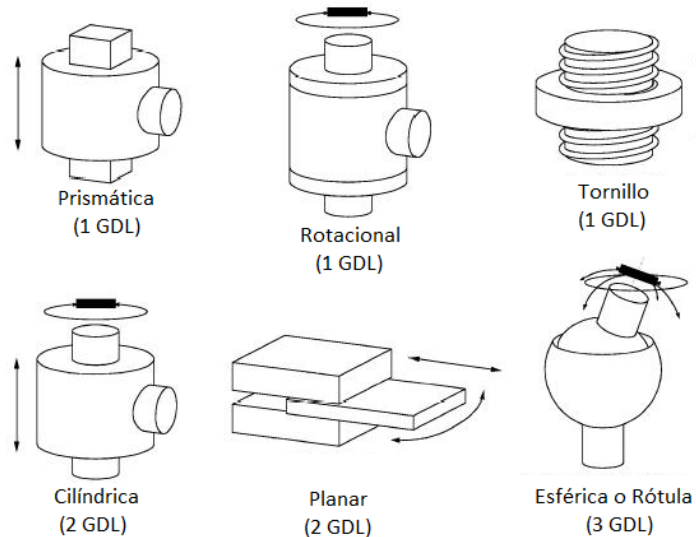


Figura 1. Tipos de articulaciones de un robot [2].

Los robots manipuladores se pueden clasificar según varios criterios, como por su geometría o configuración, su estructura cinemática, el tipo de aplicación para la cual están diseñados, la forma en que están controlados, etc. Tal clasificación es útil principalmente para determinar qué robot es el adecuado para una tarea dada. Estos manipuladores generalmente se clasifican cinemáticamente sobre la base del brazo o las primeras tres articulaciones, con la muñeca descrita por separado [5].

El empleo de diferentes combinaciones de articulaciones en un robot da lugar a diferentes configuraciones con características a tener en cuenta tanto en el diseño y construcción del robot como en su aplicación. De acuerdo a la configuración del robot, se

pueden clasificar en cinco tipos: *cartesiano (PPP)*, *cilíndrico (RPP)*, *esférico (RRP)*, *SCARA (RRP)* y *rotacional o antropomórfico (RRR)*, como se muestra en la figura 2.

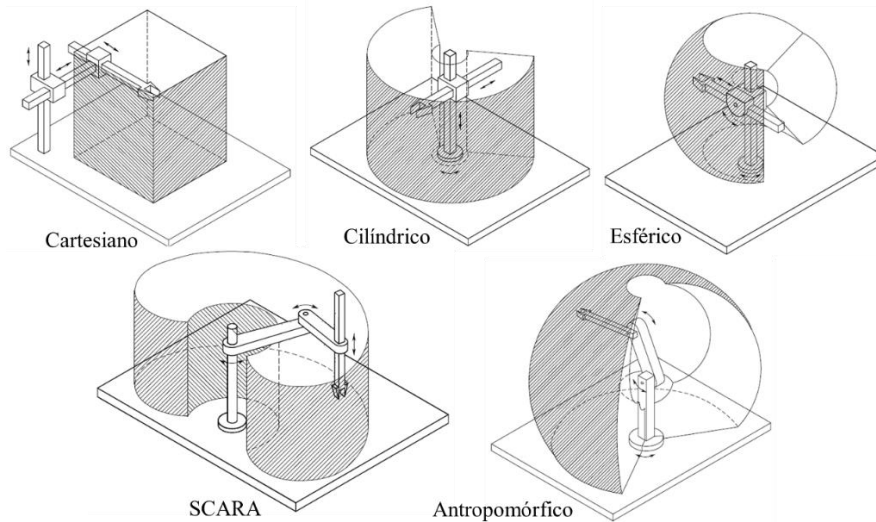


Figura 2. Configuración de los robots manipuladores y su espacio de trabajo [3].

Los robots cartesianos son máquinas automatizadas empleadas en distintos sectores, principalmente en la industria de maquinaria y manufactura, con el propósito de aumentar la velocidad, repetitividad y precisión en los distintos procesos que se llevan a cabo. La cinemática de este tipo de manipulador es la más simple de todas las configuraciones existentes. Su posicionamiento en el espacio se lleva a cabo mediante articulaciones lineales, ya que posee tres movimientos lineales, es decir, tiene tres grados de libertad, que utilizan el sistema de coordenadas cartesianas, los cuales corresponden a los movimientos localizados en los ejes X, Y y Z.

Estos robots por lo general se emplean para realizar trabajos de pick & place (recoger y colocar), ensamblaje, corte, manipulación de herramientas, transferencia de material o carga, soldadura, impresión en 3D, entre otras.

La aparición de los primeros robots industriales se puede decir que se remontan en la industria textil del siglo XVIII, cuando Joseph Jacquard inventa en 1801 una máquina textil con la que utilizaba tarjetas perforadas para programar las acciones de la máquina (figura 3). Con esto se da inicio a la automatización industrial mediante la implementación de dispositivos automáticos en la producción. La revolución industrial impulsó el desarrollo de estos agentes mecánicos, entre los cuales se destacaron el torno mecánico motorizado de Babbitt (1892) y el mecanismo programable para pintar con spray de Pollard y Roselund (1939).



Figura 3. Máquina textil de Joseph Jacquard programada mediante tarjetas perforadas.

Los teleoperadores, o dispositivos maestro-esclavo, eran desarrollados durante la segunda guerra mundial para manejar materiales radiactivos. El control numérico computarizado (CNC) se desarrolló debido a la alta precisión requerida en el mecanizado de ciertos elementos, como componentes de aviones de alto rendimiento. Los primeros robots combinaban esencialmente los enlaces mecánicos del teleoperador con la autonomía y la programabilidad de las máquinas CNC [5].

La primera máquina CNC, financiada por la Fuerza Aérea Norteamericana, fue creada en los laboratorios del Instituto Tecnológico de Massachusetts (MIT) en 1952. Era una fresadora *Cincinnati* modificada (figura 4) y tenía la capacidad de coordinar los movimientos de sus ejes de coordenadas (X, Y, Z) para mecanizar una superficie compleja. Estas primeras máquinas CNC no fueron comerciales sino hasta que se presentaron en la feria *Nacional Machine Tool Show* de 1955. Mientras tanto, fueron utilizadas para elaborar piezas que formarían parte de aviones militares principalmente e incluso otro tipo de componentes similares [6].

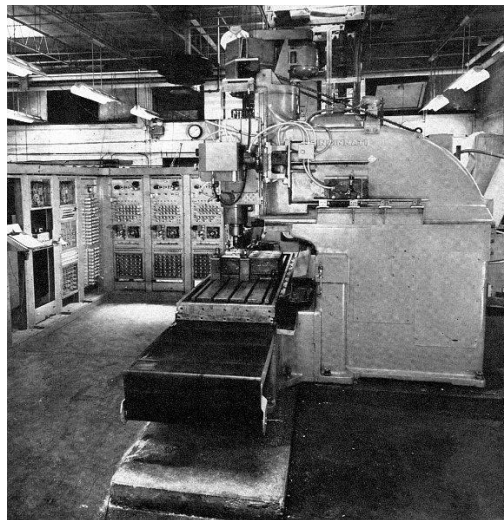


Figura 4. Fresadora Cincinnati.

Una de las aplicaciones más populares que se le ha dado al robot cartesiano en la actualidad es el de utilizarlo como impresora 3D. La impresión 3D se basa en tecnologías de procesos aditivos o por inyección de polímeros, es decir, a partir de un modelo digital, por lo general diseñado en un programa CAD (por sus siglas en inglés, *Computer Aided Design*), la fabricación aditiva permite manipular de manera automática distintos materiales y agregarlos capa a capa de forma muy precisa para construir un objeto en tres dimensiones.

El inicio de la impresión 3D se remonta a 1976, cuando se inventó la impresora de inyección de tinta. En 1984, algunas adaptaciones y avances sobre el concepto de la inyección de tinta transformaron la tecnología de impresión con tinta a impresión con materiales cuando Charles Hull inventa la *estereolitografía (SLA)*. Este método de impresión 3D se basa en la sintetización por capas usando para ello un láser. Unos años más tarde, en 1986, Hull patenta su invento y funda *3D Systems*, la primera compañía del sector. *3D Systems* también será la primera empresa en ofrecer una impresora 3D SLA comercial en 1987, en concreto su modelo SLA-1 (figura 5) [7].



Figura 5. Primera impresora 3D comercial SLA-1.

Actualmente, en el mercado se encuentran disponibles robots cartesianos más sofisticados como los que se utilizan para impresión 3D, tal es el caso de la *DT 600+* (figura 6) desarrollada por la empresa española *Dynamical Tools*. Cuenta con un doble cabezal independiente que permite trabajar con diferentes materiales en una misma pieza. Esta misma característica permite también fabricar dos piezas de forma simultánea, agilizando un proceso productivo.



Figura 6. Impresora 3D: DT 600+.

También se encuentra la impresora 3D *Industry F340* (figura 7) desarrollado por la empresa polaca *3DGence*. Esta impresora ofrece la posibilidad de elegir entre 3 diferentes cabezales de impresión para lograr mejores resultados de impresión según el tipo de material que se desee imprimir.



Figura 7. Impresora 3D: Industry F340.

Otro modelo es la *BigRep Studio* (figura 8) desarrollado por la empresa alemana *BigRep*. Su volumen de impresión de 500mm x 1000 mm x 500mm proporciona espacio para objetos grandes, mientras al mismo tiempo permite una mayor precisión y una impresión más rápida [8].



Figura 8. Impresora 3D: BigRep Studio.

La gran aplicabilidad de los Sistemas Embebidos en cualquier ámbito sectorial, así como el valor añadido que aportan los mismos a los productos que los contienen, hace que el desarrollo de estos sistemas sea un área estratégica preferente para muchas empresas que buscan precisamente este aumento de su competitividad. Así, los Sistemas Embebidos van a jugar un papel vital en nuestra sociedad y se supone revolucionarán los sectores de actividad, como son el sector médico, el de medios de transporte o el de automatización industrial, entre otros.

Un *sistema embebido* es definido como un sistema mínimo digital de propósito específico cuya plataforma electrónica depende de un microprocesador, memorias y periféricos dedicados, tienen tareas específicas y pueden formar parte de sistemas más complejos de hardware y/o software. Son sistemas que están dedicados exclusivamente al procesamiento, control y automatización de procesos físicos en tiempo real. En el diseño de sistemas embebidos hay que considerar diferentes factores los cuales son el bajo costo, reducido número de componentes, bajo consumo de potencia y respuesta en tiempo de real [9].

La característica más importante de los sistemas embebidos es su interacción con el mundo exterior en función del tiempo o en función de la presencia de estímulos. Para garantizar una interacción exitosa con el ambiente, el sistema debe incorporar algunas características, tales como la disponibilidad, fiabilidad y seguridad. Propiedades no funcionales, tales como la tolerancia a fallas, el tamaño, el consumo de potencia, el peso, la disponibilidad, la seguridad y la fiabilidad, deben definirse desde etapas tempranas de la construcción del sistema. Desde la aparición del circuito integrado, existen dos alternativas para realizar un hardware digital: codificación del algoritmo en un microprocesador o mapeo directo del algoritmo en hardware.

Hasta la década de 1970, los diseños de sistemas electrónicos se basaban en componentes analógicos discretos, como transistores, amplificadores operacionales, resistencias, condensadores e inductores. Estos circuitos ofrecían procesamiento simultáneo pero tenían problemas de variación de parámetros con la temperatura y el envejecimiento. La llegada de los componentes basados en TTL sentó las bases del diseño digital. El microprocesador Intel 4004 se convirtió en la primera plataforma digital que se podía configurar con software [10].

Los microprocesadores junto con los microcontroladores y DSPs permiten resolver eficazmente la mayor parte de los problemas electrónicos. En el mundo digital existen infinidad de problemas que se resuelven con un procesador sencillo. En efecto, aún hoy los micros de 8 bits son los más vendidos, seguidos de dispositivos de 4 bits. Sin embargo, existe una serie de problemas donde los micros no son suficientes. Tal es el caso cuando la E/S de datos combina gran cantidad y gran velocidad, o cuando el número de operaciones por muestra es elevado. En la tabla 1 se muestra la evolución de los dispositivos electrónicos. En tal caso, la única opción es la realización de un ASICs (Circuito Integrado de Aplicación Específica), una tecnología que aparece en la década de los 80, como se muestra en la tabla 1 donde se presenta la evolución de los dispositivos electrónicos.

Tabla 1. Evolución de los dispositivos electrónicos.

Componente	Época aproximada	Orden de velocidad
Relés	1930-1940	mseg.
Válvulas	1940-1960	μseg
Transistores	1950-	μseg
Circuito integrado	1960-	μseg-nseg.
Microprocesadores	1971-	μseg-nseg.
ASICs	1980-	nseg-pseg.
Gate Array	1980-	nseg-pseg.
Standard Cells	1890-	nseg-pseg.
Laser PGA	1980-	nseg-pseg.
FPGA	1985	nseg-pseg.

Los dispositivos FPGA se basan en lo que se conoce como arreglos de compuertas, los cuales consisten en la parte de la arquitectura que contienen tres elementos configurables: bloques lógicos configurables (CLB), bloques de entrada y salida (IOB) y canales de comunicación. A diferencia de los CPLD, la densidad de los FPGA se establece en cantidades equivalentes a cierto número de compuertas.

Por adentro, un FPGA está formado por arreglos de bloques lógicos configurables (CLB), que se comunican entre ellos y con las terminales de entrada/salida (E/S) por medio de alambrados llamados canales de comunicación. Cada FPGA contiene una matriz de bloques lógicos idénticos, por lo general de forma cuadrada, conectados por medio de las líneas metálicas que corren vertical y horizontalmente entre cada bloque, como se muestra en la figura 9.

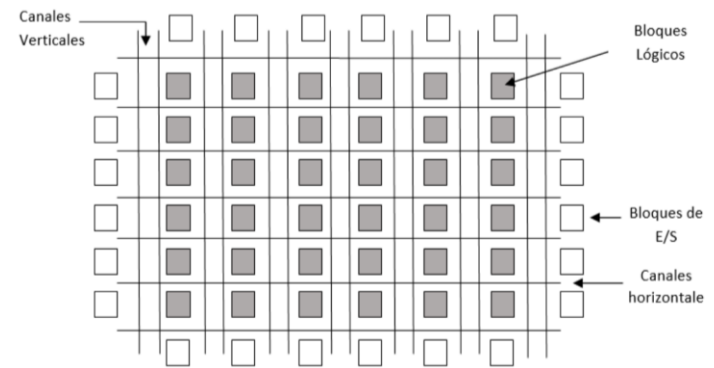


Figura 9. Arquitectura básica de un FPGA.

Actualmente, en la Maestría en Ciencias de la Electrónica opción Automatización de la Benemérita Universidad Autónoma de Puebla, se cuenta con una tarjeta de desarrollo que como parte principal tiene un FPGA de la familia Cyclone III, la cual se muestra en la figura 10. Este fue un diseño de cinco capas y tiene componentes en la parte superior e inferior.

La parte superior, consta de:

1. Un FPGA Cyclone III con matrícula EP3C10F256C6 de Altera.
2. Un puerto de programación compatible con el programador usb-blaster.
3. Un puerto JTAG.

4. Un conector para alimentación externa de 3.3V.
5. Un botón de reinicio.
6. 4 conectores header macho de 50 pines.

La parte inferior, consta de:

1. Memoria EEPROM (EPC2).
2. Convertidor DC-DC de 3.3V a 1.2V,
3. Oscilador de cristal de 100 MHz.
4. Resistencias y capacitores.
5. Circuito de reset.
6. Convertidor DC-DC de 3.3V a 2.5V

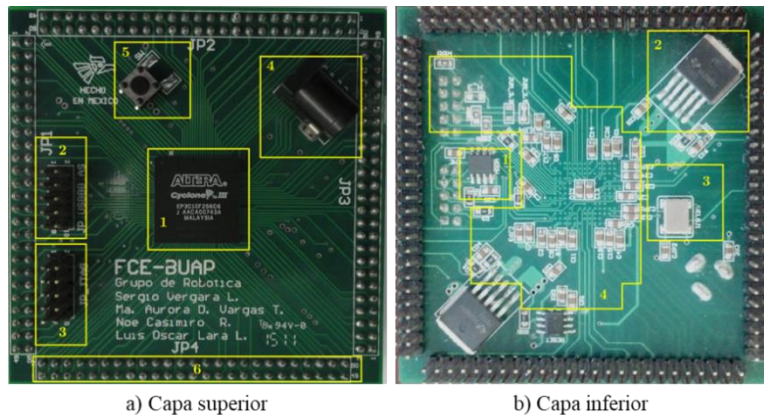


Figura 10. Tarjeta de desarrollo con FPGA Cyclone III.

Otras características importantes de este FPGA son las siguientes:

- 10,320 elementos lógicos
- 256 pines
- Condiciones de temperatura grado comercial: 0°C-80°C
- Grado de velocidad: 6 nano segundos
- Memoria RAM de 423,936 bits
- 23 multiplicadores de 18 x 18
- 2 PLLs
- 182 pines de I/O, (el número de I/O incluye relojes y pines de configuración). Por lo cual, la tarjeta solo ofrece 168 I/O de propósito general.

De igual manera, existen desarrollos realizados por diversas compañías cuyo funcionamiento se basa en un PLD; por ejemplo, la figura 11 ilustra una tarjeta de desarrollo DE0-CV, la cual es una plataforma robusta de diseño de hardware que utiliza el dispositivo FPGA Cyclone V de Intel como control central de sus periféricos, así como el USB Blaster integrado, las capacidades de video y mucho más. Con esta plataforma se obtiene la potencia y el rendimiento necesario para aplicaciones de gran volumen, incluidos puentes de prototipos, unidades de control de motores, tarjetas de captura y dispositivos de mano. Además, permite el procesamiento de datos en paralelo a alta velocidad, lo que reduce los problemas de procesamiento de datos intensivo.

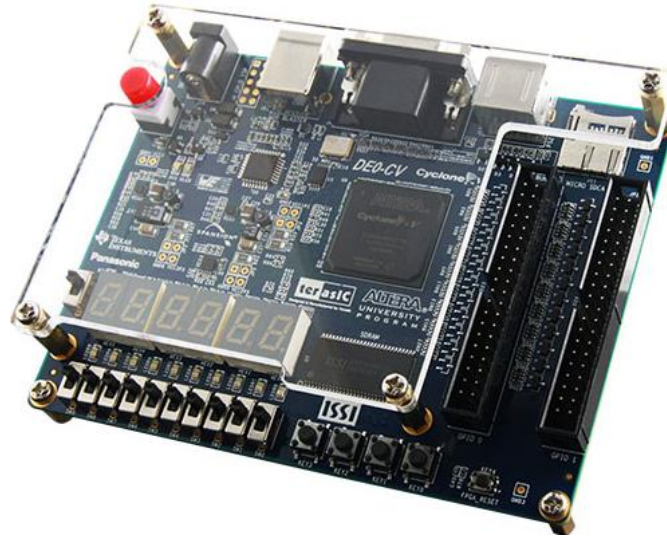


Figura 11. Tarjeta de desarrollo DE0-CV con FPGA Cyclone V de Intel.

Las características de la tarjeta DE0-CV son las siguientes [11]:

- Cyclone V FPGA 5CEBA4F23C7N
- 49,000 elementos lógicos programables
- 3080 Kbits de memoria embebida
- Memoria SDRAM de 64MB x 16 bits de bus de datos
- 4 PLLs fraccionales
- Memoria EPCS64
- 2x20 pines de propósito general (72 pines I/O, 2 pines de alimentación de 3.3VCD y 2 de 5VCD con sus respectivas tierras)
- 6 displays de 7 segmentos
- 10 LEDs
- 10 interruptores
- 4 pulsadores de propósito general
- 1 pulsador para el reset de la CPU

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, esto se debe a que mientras las empresas nacionales tienen que importar tecnología en muchos casos obsoleta, las empresas extranjeras usan tecnología de punta en sus procesos de producción. En la actualidad, existen impresoras 3D muy baratas, desafortunadamente su duración de vida es muy corta, por otra parte, también se pueden encontrar impresoras 3D industriales robustas y más sofisticadas las cuales pueden realizar mayor número de piezas, el inconveniente es que su costo es mucho más elevado. Por lo anterior, es muy importante por la naturaleza de nuestra maestría ganar experiencia en el desarrollo de máquinas que en un futuro puedan usarse en sistemas automatizados de producción.

De igual manera, debido a que el desarrollo de instrumentos controlados por computadora está permitiendo automatizar procesos de caracterización y medición de diferentes sistemas usados en diversas áreas del conocimiento, el principal objetivo para esta

tesis es realizar el control de un robot CNC para que funcione como impresora 3D mediante con el microprocesador diseñado en el nuevo sistema embebido basado en un FPGA Cyclone V modelo 5CEBA4F23C7N, para tener así una plataforma experimental para probar controladores e incrementar la infraestructura de la maestría. Esta tesis elevará la productividad de la maestría ya que se contaría con un prototipo institucional.

Objetivos:

General:

“Desarrollar un sistema de control para un robot cartesiano de 3 gdl por medio de un sistema embebido”

Específicos:

- Revisión bibliográfica de robots y sistemas embebidos.
- Instrumentar el sistema embebido.
- Diseñar e instrumentar el sistema de control para el robot cartesiano.
- Diseñar el hardware necesario para el control del robot.
- Diseñar el software de control.
- Realizar pruebas del sistema diseñado.
- Publicación de resultados.
- Escritura de la tesis.

Organización de la Tesis:

Capítulo 1: Se hace la descripción general del sistema mecatrónico, donde se mencionan las características mecánicas del robot CNC, así como las especificaciones, cambios y mejoras necesarias para poder implementarlo como impresora 3D.

Capítulo 2: Se muestra el desarrollo de la cinemática directa y el modelo dinámico del robot.

Capítulo 3: Se describen los componentes mecánicos que conforman el robot y la electrónica necesaria para complementarlo como impresora 3D. De igual manera, se presenta la caracterización de algunos componentes electrónicos, así como el desarrollo de sus etapas de potencia.

Capítulo 4: Se realiza el firmware y software de control, en el cual se muestran las estrategias y algoritmos de control implementados.

Capítulo 5: Se presentan los resultados experimentales con los algoritmos de control propuestos en los distintos actuadores de forma individual y finalmente, de las impresiones obtenidas de todo el robot funcionando como impresora 3D.

Capítulo 1. Descripción del sistema mecatrónico

Se cuenta con un robot CNC tipo cartesiano de 3 grados de libertad (gdl), como se muestra en la figura 12, cuya estructura mecánica fue desarrollada por [12] en la Maestría en Ciencias de la Electrónica opción en Automatización (MCEA) de la Benemérita Universidad Autónoma de Puebla (BUAP). Dicho robot cuenta con la mecánica de una máquina CNC (Control Numérico por Computadora), lo cual ofrece grandes beneficios para poder hacerle la adaptación necesaria para ser implementado como impresora 3D, o en un futuro como máquina multiherramienta.

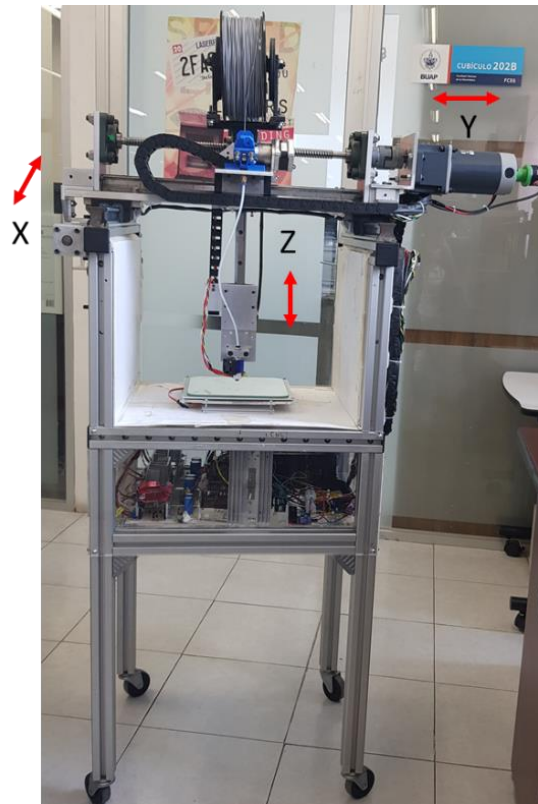


Figura 12. Robot CNC de 3 gdl.

1.1 Características generales

En base a trabajos desarrollados anteriormente, se tiene este robot CNC de configuración tipo cartesiano cuya mecánica se construyó a base de estructuras de acero, en cada gdl se cuenta con rieles de alta precisión apoyados con carros lineales de precisión para obtener un mejor desplazamiento lineal con mínima fricción en el sistema. De igual manera, para transformar el desplazamiento giratorio a lineal se utilizaron tornillos embalados.

En cuanto a los actuadores de los 3 gdl, se tienen motores reductores que se alimentan a 12V, proporcionan 3100 rpm y generan un torque de 3-4 Nm, los cuales para poder ocuparlos como servomotores se les acopló un encoder giratorio de 1000 pulsos/revolución para determinar la posición actual.

En la tabla 2 se muestran las características generales del sistema mecatrónico, es decir, del robot CNC implementado como impresora 3D.

Tabla 2. Características generales del robot.

Descripción	Requerimiento
Número de grados de libertad	3 (X, Y, Z)
Ejes X y Y	<ul style="list-style-type: none"> Desplazamiento: $X=276\text{mm}, Y=245\text{mm}$ Tornillo embalado de 12mm Guía de alta precisión de 32mm Bloque para guía de 32mm
Eje Z	<ul style="list-style-type: none"> Desplazamiento: 145mm Tornillo embalado de 12mm Guía de alta precisión de 15mm Bloque para guía de 15mm
Tamaño (mm)	$950 \times 1470 \times 860$
Volumen de impresión (mm)	$214 \times 214 \times 70$
Alimentación	$120V_{CA}$ y $12V_{CD}$
Cama caliente	MK3
Extrusor	Metal J-Head
Resolución	$\Delta x = 0.286\mu\text{m}$ $\Delta y = 0.287\mu\text{m}$ $\Delta z = 0.171\mu\text{m}$
Velocidad	$v_x = 35 \text{ mm/s}$ $v_y = 25 \text{ mm/s}$ $v_z = 21 \text{ mm/s}$
Material de impresión	PLA
Software	Abierto
Conectividad	Wifi
Sistemas operativos	Abierto

Dicho robot ya contaba con control por posición de los 3 gdl mediante un microprocesador desarrollado en la MCEA de la BUAP [13], el cual estaba embebido en una tarjeta FPGA EP3C10F256C6 de la familia Cyclone III. El microprocesador permite procesar las instrucciones recibidas en lenguaje ensamblador y realizar las correspondientes operaciones aritméticas, con el fin de llevar a cabo el control del robot. Este se diseñó en base a arquitectura MIPS, el cual es un procesador de 32 bits con arquitectura RISC y cuenta con un banco de 32 registros de 32 bits.

En la figura 13 se muestra el diagrama general del funcionamiento de este robot por medio del microprocesador embebido en la FPGA con Cyclone III. El robot puede ser manipulado por el usuario mediante comunicación wifi a través de una interfaz desarrollada en LabView donde se pueden modificar y visualizar los parámetros del sistema, como posiciones deseadas y ganancias de ajuste para el control de cada grado de libertad.

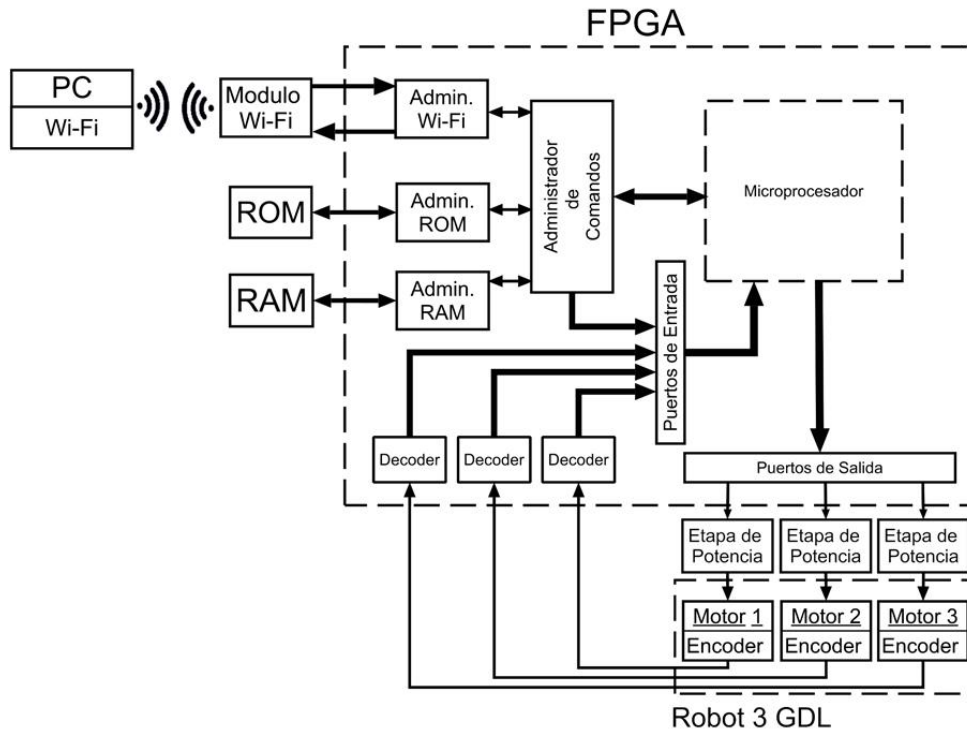


Figura 13. Diagrama general del funcionamiento del sistema inicial.

1.2 Especificaciones

Para utilizar este robot como impresora 3D se necesita mejorar el control de posición en cada grado de libertad para posteriormente poder implementar control de velocidad, es por eso que se cambiarán los encoders de cuadratura por unos nuevos ya que los que traía el robot están dañados.

Este robot cuenta con la herramienta de extrusión que está compuesta por el extrusor de material, la cama caliente y el alimentador de material (motor a pasos bipolar). Cabe mencionar que el extrusor de material se tuvo que reemplazar por uno nuevo ya que, al igual que los encoders, este se encuentra en mal estado. De igual manera, se necesita llevar a cabo el diseño y la construcción del porta carrete, el cual se encargará de sujetar el rollo de filamento del material a extruir.

Para llevar a cabo el proceso de impresión en 3D de manera adecuada mediante la herramienta de extrusión, se necesita desarrollar el control de temperatura, tanto del extrusor de material como de la cama caliente, ya que el extrusor de material tiene que trabajar en un rango de temperatura de 190-210 °C y la cama caliente de 40-60 °C. Es por eso que, para realizar esto también se tiene que implementar la electrónica de potencia adecuada para drenar la suficiente corriente que necesita cada dispositivo para trabajar de manera eficiente.

En cuanto al alimentador de material, se cuenta con un motor a pasos bipolar el cual se encargará de proporcionar material al extrusor, en este caso se utilizará filamento PLA que es uno de los materiales más comunes para impresión 3D, por lo cual es necesario también desarrollar el control de velocidad en lazo abierto para este motor.

Por otra parte, para evitar que el carro de precisión sobrepase los límites del área de trabajo del robot se le adaptarán interruptores de fin de carrera a cada eje para evitar accidentes o daños en el sistema mecánico.

Para realizar el proceso de impresión en 3D se necesita de un software laminador en el cual se carga el archivo con el diseño de la pieza a imprimir (diseño CAD guardado con extensión *.stl*) y se genera el código G de la pieza, posteriormente a través de un software compilador propio se hace la depuración del código G para obtener las coordenadas necesarias para formar la pieza en 3D que pasarán a ser las posiciones deseadas y al ejecutar la secuencia de control estas posiciones se almacenarán en la memoria RAM.

Debido a las mejoras y a la implementación que se le quieren hacer al robot para que funcione como impresora 3D, se necesita expandir el firmware para agregar los bloques que llevarán a cabo los controles de temperatura del extrusor y de la cama caliente, así como también los bloques para el control del motor a pasos que se utilizará como alimentador de material, por lo tanto, la FPGA EP3C10F256C6 de la familia Cyclone III ya no es adecuada para este proyecto ya que no ofrece suficientes elementos lógicos para almacenar el firmware necesario, por lo cual se decidió migrar el microprocesador a una tarjeta más potente como lo es la FPGA 5CEBA4F23C7N de la familia Cyclone V.

1.3 Resolución real del robot CNC

Lo importante, tanto en las impresoras 3D como en las máquinas CNC, es la calidad, la cual se da gracias a la resolución de la máquina. Para conocer la resolución real de nuestro robot se deben considerar ciertas variables del sistema como la longitud que se debe desplazar el carro de precisión en cada eje, el número de vueltas que gira el tornillo embalado para recorrer cierta distancia, la reducción de la caja de engranes y la resolución del encoder.

En este caso, los motores para los ejes X y Y cuentan con reductores de 9:1, es decir, 9 giros del motor por 1 del reductor; pero comprobándolo manualmente, por una vuelta del reductor solo se tienen 8.69 vueltas del motor. Para el motor del eje Z se tiene un reductor de relación 15:1, el cual también se comprobó de forma manual y por una vuelta del reductor se contaron 14.6 vueltas del motor. En cuanto a los tornillos embalados, estos tienen una resolución de una vuelta/cm. Los encoders tienen una resolución de 1000 pulsos/vueltas, al ser encoders de cuadratura se multiplica por 4 que son los posibles cambios, por lo tanto, se tienen 4000 combinaciones, es decir, 4000 conteos/vuelta del motor. A partir de esto, en (1), (2) y (3) se calcula la resolución real de los ejes X, Y y Z, respectivamente.

$$\Delta_x = \frac{L_x}{N * R * C} = \frac{276 \text{ mm}}{27.75 * 8.69 * 4000} = 0.286 \mu\text{m} \quad (1)$$

$$\Delta_y = \frac{L_y}{N * R * C} = \frac{245 \text{ mm}}{24.5 * 8.69 * 4000} = 0.287 \mu\text{m} \quad (2)$$

$$\Delta_z = \frac{L_z}{N * R * C} = \frac{145 \text{ mm}}{14.5 * 14.6 * 4000} = 0.171 \mu\text{m} \quad (3)$$

Donde:

- $\Delta_x, \Delta_y, \Delta_z$ es la resolución de los ejes X, Y y Z .
- L_x, L_y, L_z es la longitud total desplazada en cada eje (mm).
- N es el número de vueltas necesarias para desplazarse en L .
- R es la reducción de la caja de engranes del motor.
- C son los conteos del encoder por vuelta del motor.

Con esto, se puede garantizar tener una máquina de alta precisión que permita realizar impresiones de alta calidad.

1.4 Diagrama general

En la figura 14 se muestra el diagrama a bloques general del sistema mecatrónico que se planea desarrollar para que funcione como impresora 3D. Para esto, el robot puede ser manipulado por el usuario mediante una interfaz desarrollada en LabView donde se pueden modificar y visualizar los parámetros del sistema, como posiciones y temperaturas deseadas y ganancias de ajuste para los controladores, así como también sirve para enviar el software de control hacia el microprocesador a través de protocolo de comunicación Wi-Fi de una computadora como interfaz. El robot se comunicará con el sistema digital mediante la FPGA Cyclone V (con el microprocesador embebido), la cual ejecutará los algoritmos de control para controlar la posición de los 3 gdl del robot CNC (x, y, z), así como también se encargará de mantener la temperatura deseada del extrusor de material y de la cama caliente donde se irá depositando el material extruido y accionará el motor a pasos a determinada velocidad para proporcionar material al extrusor, llevando de esta manera la acción de impresora 3D.

En el caso de los 3 gdl, la FPGA se encargará de leer los encoders de cuadratura de cada servomotor para determinar su posición actual, aplicará la ley de control y mediante el generador de PWM (Pulse Width Modulation) mandará la señal a la etapa de potencia (puente H) de cada motor el cual proporcionará la energía necesaria a cada actuador para que se muevan a las posiciones deseadas.

En cuanto al extrusor de material y la cama caliente, estos traen integrado una resistencia eléctrica para calentarse y un termistor mediante el cual se determina la temperatura actual del dispositivo. Para que la FPGA pueda leer estas señales y procesarlas se debe hacer una conversión de la señal analógica a digital para lo cual se implementarán ADC0804 de 8 bits (uno para cada dispositivo). Después se aplica la ley de control y mediante el generador de PWM se mandará la señal de activación a la etapa de potencia, tanto del extrusor como de la cama caliente, proporcionando la energía necesaria para mantener cada actuador a la temperatura de trabajo que le corresponde.

Finalmente, para el control de velocidad del motor a pasos bipolar, solo se necesita el firmware con la secuencia de pasos adecuada para hacer girar el motor hacia un solo sentido, de modo que, el alimentador de material tire el filamento de manera controlada y evitando saturación en el extrusor de material.

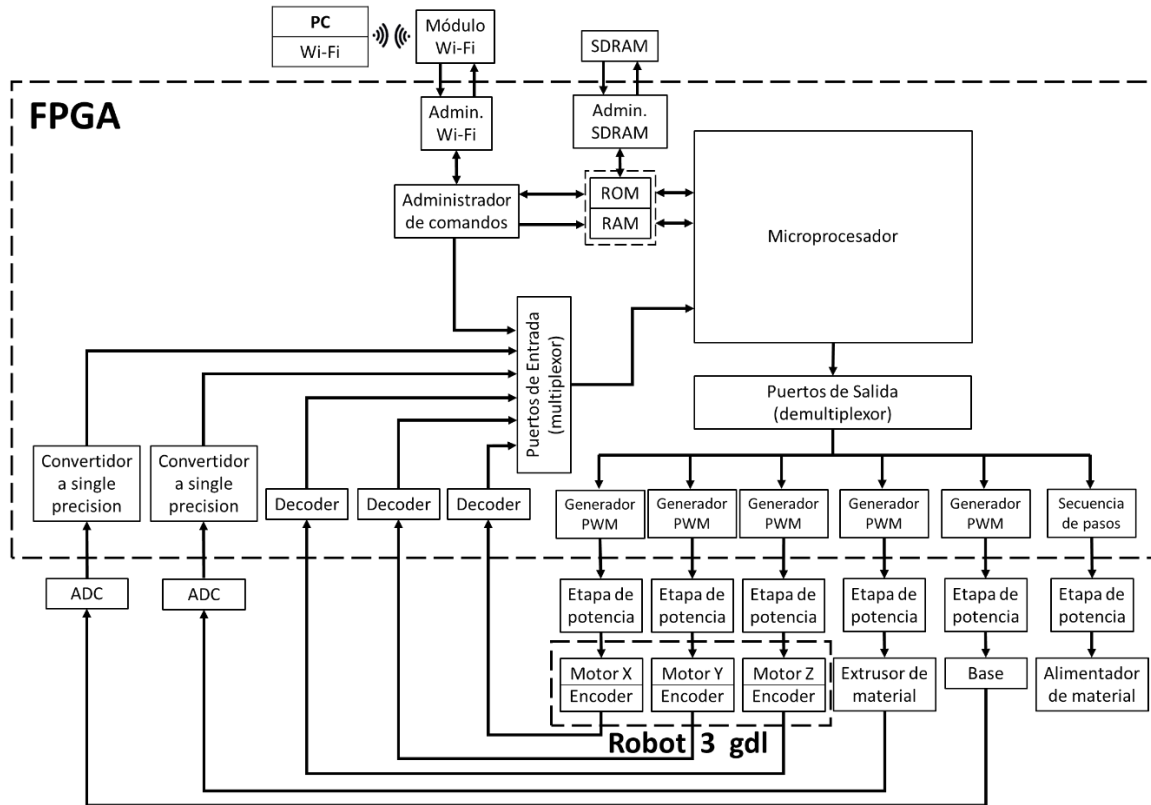


Figura 14. Diagrama general del robot CNC con 3 gdl aplicado a impresión 3D.

1.5 Conclusiones

Este proyecto parte de un robot CNC de 3 gdl controlado por posición. A partir de esto se propone desarrollar la electrónica, el firmware y software de control adecuado para que el robot pueda funcionar como una impresora 3D industrial, ya que trabaja con motores DC y encoders de cuadratura para determinar la posición actual y también cuenta con rieles y carros de alta precisión en cada eje en lugar de bandas dentadas y varillas como las impresoras 3D de escritorio, lo cual garantiza que después de varios usos el sistema no se descalibrará.

Además, gracias a la resolución que presenta esta máquina se lograrán alcanzar posiciones pequeñas de forma precisa, lo cual presenta una ventaja para la implementación como impresora 3D ya que se podrá contar con un robot de alta calidad.

Capítulo 2: Modelo Cinemático y Dinámico

Es importante determinar el modelo dinámico para realizar un control más apropiado del robot y que además presente un mejor desempeño, de igual manera, es muy útil para analizar su comportamiento con diferentes algoritmos de control.

En la figura 15 se muestra la representación geométrica del robot cartesiano de 3 gdl, el cual permitirá visualizar el comportamiento del manipulador. Tanto el modelo de cinemática directa como el modelo dinámico están basados de [12].

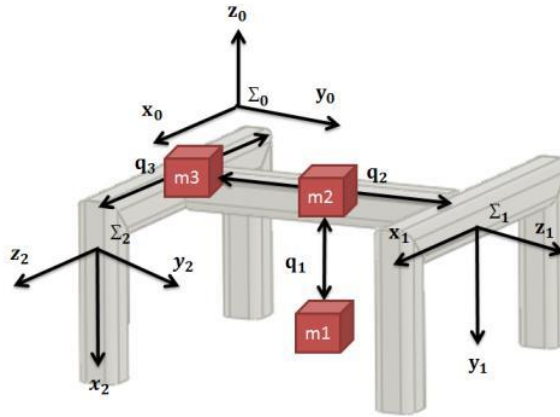


Figura 15. Representación del robot CNC de 3 grados de libertad.

2.1 Cinemática directa

La cinemática es la parte de la física que estudia el movimiento de sistemas mecánicos sin tomar en cuenta las fuerzas que lo producen. Por lo tanto, la cinemática directa de robots manipuladores se encarga del estudio analítico del movimiento del robot con respecto a un sistema de referencia cartesiano fijo $\Sigma(x, y, z)$ relacionando la dependencia que existe entre las coordenadas articulares o generalizadas $\mathbf{q} \in \mathbb{R}^n$, sus parámetros geométricos y las coordenadas cartesianas $[x, y, z]^T \in \mathbb{R}^3$ y de orientación $[\theta, \phi, \psi]^T \in \mathbb{R}^3$ del extremo final del robot [1].

Para determinar el modelo de cinemática directa se utilizará la metodología de Denavit-Hartenberg [14], la cual ofrece un procedimiento sencillo y cuya estructura queda en representación de transformaciones homogéneas. Las variables articulares en esta representación son las siguientes:

- θ_i es el ángulo de articulación.
- d_i/β_i es el desplazamiento de la articulación.
- l_i es la longitud del eslabón.
- α_i es el ángulo de torsión.

La representación de Denavit-Hartenberg se representa por el producto de cuatro transformaciones básicas:

$$H_{i-1}^i = H_{Rz_{i-1}}(\theta_i)H_{Tz_{i-1}}(d_i; \beta_i)H_{Tx_{i-1}}(l_i)H_{Rx_{i-1}}(\alpha_i) \quad (4)$$

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

Tomando la figura 15 como referencia para obtener la tabla de parámetros de Denavit-Hartenberg, se necesita realizar una rotación auxiliar debido a que los ejes del sistema son ortogonales. A continuación, se presenta el proceso realizado para abordar esta problemática.

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

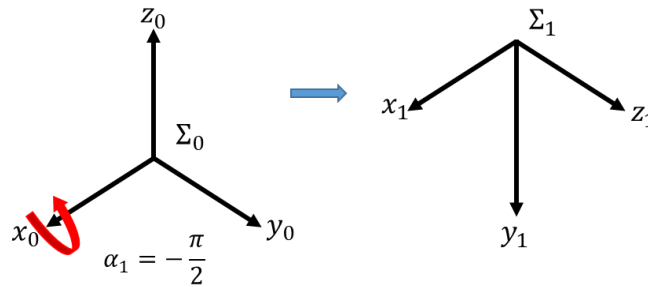


Figura 16. Primera rotación Denavit-Hartenberg para obtener el sistema de referencia Σ_1 .

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 lleva a:

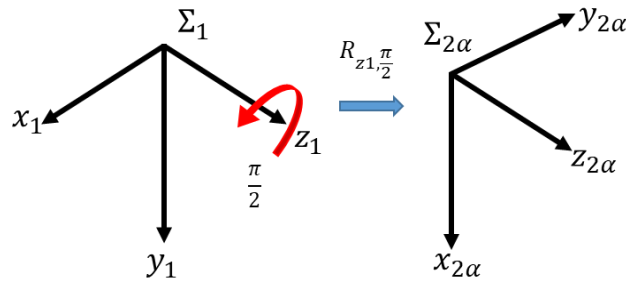


Figura 17. Rotación auxiliar Σ_{2a} para alcanzar la alineación deseada en el sistema Σ_2 .

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 .

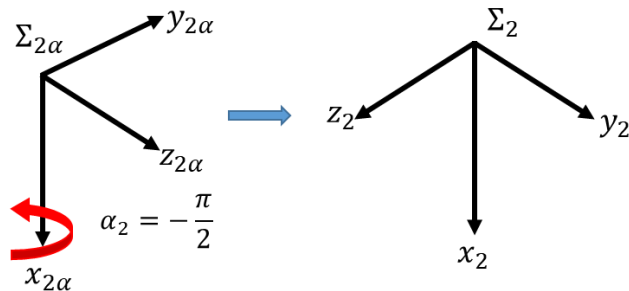


Figura 18. Sistema de referencia Σ_2 , después de la última rotación realizada sobre el eje x_{2a} del sistema Σ_{2a} .

Con la información anterior se pueden adquirir los parámetros necesarios para llenar la tabla de Denavit-Hartenberg, quedando como se muestra en la tabla 3.

Tabla 3. Tabla de parámetros de Denavit-Hartenberg.

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

Sustituyendo en (4) los parámetros obtenidos y después de realizar las operaciones algebraicas necesarias, se tiene para el primer eslabón:

$$H_0^1 = H_{Rz_0}(0)H_{Tz_0}(d_1)H_{Tx_0}(0)H_{Rx_0}\left(-\frac{\pi}{2}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

La cinemática directa será:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix} \quad (6)$$

Para el segundo eslabón, se necesita determinar primero la transformación homogénea del segundo eslabón al primer eslabón, de lo cual se obtiene:

$$H_1^2 = H_{Rz_1}(0)H_{Tz_1}(d_2)H_{Tx_1}(0)H_{Rx_1}\left(-\frac{\pi}{2}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

La transformación homogénea total será:

$$H_0^2 = H_0^1 H_1^2 \quad (8)$$

Es decir,

$$H_0^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

Por lo tanto, la cinemática directa para el segundo eslabón es:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ d_2 \\ d_1 \end{bmatrix} \quad (10)$$

Finalmente, para el tercer eslabón se tiene que utilizar la transformación homogénea auxiliar anteriormente explicada, entonces:

$$H_{2a} = H_{Rz_1}\left(\frac{\pi}{2}\right)^T H_1^2 = H_{Rz_1}\left(\frac{\pi}{2}\right)^T H_{Rz_1}(0)H_{Tz_1}(d_2)H_{Tx_1}(0)H_{Rx_1}\left(-\frac{\pi}{2}\right)$$

$$H_{2a} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

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

$$H_2^3 = H_{Rz_2}(0)H_{Tx_2}(d_3)H_{Tx_2}(0)H_{Rx_2}(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

La transformación homogénea resultante es:

$$H_0^3 = H_0^1 R_{z, \frac{\pi}{2}}^T H_1^2 H_2^3 = \begin{bmatrix} 0 & 0 & 1 & d_3 \\ 0 & -1 & 0 & d_2 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

Entonces, la cinemática directa para el tercer eslabón es:

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} d_3 \\ d_2 \\ d_1 \end{bmatrix} \quad (14)$$

Por lo tanto, la cinemática directa del robot CNC tipo cartesiano es la siguiente:

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} d_3 \\ d_2 \\ d_1 \end{bmatrix} \quad (15)$$

Tomando en cuenta que las coordenadas $[d_3 \ d_2 \ d_1]^T$ son coordenadas cartesianas.

2.2 Dinámica

El modelo dinámico de un robot manipulador permite explicar todos los fenómenos físicos intrínsecos o propios de la naturaleza dinámica que se encuentran en su estructura mecánica, como son los efectos inerciales, fuerzas centrípetas y de Coriolis, par gravitacional y fricción.

A diferencia de la cinemática directa, para determinar el modelo dinámico de un robot manipulador se requiere de ecuaciones diferenciales para la descripción analítica de su sistema mecánico. Por lo tanto, para determinar el modelo dinámico se utilizó la metodología de Euler-Lagrange [15], la cual está ampliamente documentada.

Del modelo de cinemática directa se obtiene la cinemática diferencial, que es la derivada con respecto al tiempo de la cinemática directa, es decir, se calcula a través de la velocidad lineal de cada eslabón, como se muestra en (16):

$$\mathbf{v}_i = \frac{d}{dt} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (16)$$

Para el robot manipulador de 3 gdl, esta ecuación para cada eslabón queda de la siguiente manera:

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

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

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

Posteriormente es necesario determinar la energía cinética, que incluye el movimiento de traslación y rotación, definida como:

$$K(d, \dot{d}) = \sum_1^n \frac{1}{2} [m_i \mathbf{v}_i^T \mathbf{v}_i + \mathbf{I}_i \dot{d}_i^2] \quad (20)$$

donde

- m_i es el vector de las masas del sistema.
- \mathbf{I}_i es el momento de inercia del i -ésimo eslabón.
- $\mathbf{v}_i^T \mathbf{v}_i$ representa la rapidez lineal del centro de masa de cada eslabón, es decir,

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

Por lo tanto, \mathbf{v}^2 para cada eslabón queda representada como:

$$\mathbf{v}_1^T \mathbf{v}_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{d}_1 \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ \dot{d}_1 \end{bmatrix} = \dot{d}_1^2 \quad (22)$$

$$\mathbf{v}_2^T \mathbf{v}_2 = \begin{bmatrix} 0 \\ \dot{d}_2 \\ \dot{d}_1 \end{bmatrix}^T \begin{bmatrix} 0 \\ \dot{d}_2 \\ \dot{d}_1 \end{bmatrix} = \dot{d}_2^2 + \dot{d}_1^2 \quad (23)$$

$$\mathbf{v}_3^T \mathbf{v}_3 = \begin{bmatrix} \dot{d}_3 \\ \dot{d}_2 \\ \dot{d}_1 \end{bmatrix}^T \begin{bmatrix} \dot{d}_3 \\ \dot{d}_2 \\ \dot{d}_1 \end{bmatrix} = \dot{d}_3^2 + \dot{d}_2^2 + \dot{d}_1^2 \quad (24)$$

Por otro lado, en el robot cartesiano solo se cuenta con movimiento de traslación sobre los ejes x, y, z , de esto se puede decir que la energía de rotación para estos ejes es igual a cero, por lo que, al sustituir (22), (23) y (24) en (20) la energía cinética quedará de la siguiente manera:

$$K(d, \dot{d}) = \frac{1}{2} [\dot{d}_1^2 (m_1 + m_2 + m_3) + \dot{d}_2^2 (m_2 + m_3) + \dot{d}_3^2 m_3] \quad (25)$$

Como ya se sabe, la energía potencial $\mathcal{U}(d)$ 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 (d) 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, como se muestra en la ecuación (26), ya que para el segundo y tercer eslabón se mantiene constante.

$$\mathcal{U}(d) = m_1 g d_1 \quad (26)$$

donde, g es la constante gravitacional.

Una vez calculada la energía cinética (25) y la energía potencial (26), se obtiene el lagrangiano, el cual se define como la diferencia entre la energía cinética $K(d, \dot{d})$ y la energía potencial $\mathcal{U}(d)$, es decir, $L(d, \dot{d}) = K(d, \dot{d}) - \mathcal{U}(d)$:

$$L(d, \dot{d}) = \frac{1}{2} \left[\dot{d}_1^2 (m_1 + m_2 + m_3) + \dot{d}_2^2 (m_2 + m_3) + \dot{d}_3^2 m_3 \right] - m_1 g d_1 \quad (27)$$

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

$$\boldsymbol{\tau} = \frac{d}{dt} \left[\frac{\partial L(d, \dot{d})}{\partial \dot{d}} \right] - \frac{\partial L(d, \dot{d})}{\partial d} \quad (28)$$

donde, $\boldsymbol{\tau} \in \mathbb{R}^n$ es el vector de pares aplicados.

A partir de (28) se obtiene el $\boldsymbol{\tau}$ aplicado a cada articulación, por lo tanto, el modelo dinámico del robot CNC tipo cartesiano de 3 gdl es el siguiente:

$$\boldsymbol{\tau}_1 = (m_1 + m_2 + m_3) \ddot{d}_1 - m_1 g \quad (29)$$

$$\boldsymbol{\tau}_2 = (m_2 + m_3) \ddot{d}_2 \quad (30)$$

$$\boldsymbol{\tau}_3 = m_3 \ddot{d}_3 \quad (31)$$

2.3 Conclusiones

Gracias a la cinemática directa se pueden convertir coordenadas articulares a coordenadas cartesianas, lo cual resulta más práctico de interpretar para un programador de robots al programar los movimientos de este. De igual manera, implementar la metodología Denavit-Hartenberg facilita más la obtención del modelo cinemático directo, ya que mediante dicho procedimiento se puede obtener una mínima representación de la orientación y la traslación del robot.

Por otra parte, obtener el modelo dinámico de un robot manipulador ayuda a entender mejor el comportamiento del sistema al aplicar diferentes algoritmos de control y así poder elegir el más adecuado para la aplicación que se requiera; así como también, proporciona una serie de propiedades matemáticas que facilitan el análisis, diseño y desarrollo de estrategias de control. Sin embargo, para finalidades prácticas en diversas aplicaciones resulta fundamental conocer varios aspectos del robot como el valor numérico de cada uno de los parámetros de los eslabones del robot, así como la masa, centros de masa, momentos de inercia y coeficientes de fricción.

Capítulo 3: Hardware

3.1 Componentes para el sistema mecánico

Un sistema mecánico está constituido fundamentalmente por componentes, dispositivos o elementos que tienen como función específica transformar o transmitir el movimiento desde las fuentes que lo generan al transformar distintos tipos de energía.

Los componentes que se utilizaron para el diseño de la estructura mecánica del robot cartesiano de 3 gdl se eligieron de modo tal que nuestra impresora pueda ser comparada con impresoras 3D industriales. Los componentes implementados son los siguientes:

3.1.1 Riel de alta precisión y carro lineal de precisión

Como se observa en la figura 15, el diseño del robot CNC se basa en una configuración cartesiana, en la cual sus 3 gdl son articulaciones prismáticas. Es por eso que para poder realizar los movimientos prismáticos de este sistema con suma precisión se utilizaron rieles de alta precisión apoyados con carros lineales de precisión (figura 19). Esto con el fin de poder obtener un mejor desplazamiento lineal con mínima fricción en el sistema a diferencia de las impresoras 3D de escritorio que para el movimiento de sus ejes utilizan varillas y bandas dentadas, las cuales con el tiempo y el uso van ganando holgura y debido a esto se tienen que cambiar constantemente, lo cual a su vez también provoca que la impresora se descalibre fácilmente.



Figura 19. Riel de alta precisión y carro de precisión.

Características del riel de alta precisión:

- Alta precisión en posicionamiento con alta repetitividad.
- Alta rigidez por su diseño de carga en apoyo sobre 4 puntos de la guía.
- Baja resistencia a la fricción, lo que mantiene su alta precisión por mayor tiempo.
- Ideal para aplicaciones a alta velocidad.

Las guías lineales cuadradas son los componentes más profesionales que se pueden usar para realizar movimiento lineal debido a su diseño estructural capaz de soportar grandes cargas y deslizarlas con la menor fricción y juego posible.

Características del carro de precisión:

- Más bajo nivel de ruido de funcionamiento.
- Retención de grasa óptima dentro del bloque de guía.
- Vida útil más larga.
- Bola espaciador diseñado para durar toda la vida útil del cojinete.
- Control del balón en separadores. Bola de acero de seguimiento dentro de los caminos de recirculación que proporcionan buen funcionamiento y rendimiento mejorando la vida [16].

3.1.2 Tornillo embalado

El robot cuenta con tornillos embalados (figura 20) para transformar el movimiento giratorio del motor a desplazamiento lineal, ya que 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 [17].



Figura 20. Tornillo embalado.

3.1.3 Interruptores de fin de carrera

Ya que el robot no contaba con sensores de fin de carrera se le tuvieron que adaptar uno de estos (figura 21) a cada extremo de cada eje para evitar que el carro de alta precisión sobrepase los límites del área de trabajo y dañe el sistema mecánico.



Figura 21. Interruptor de fin de carrera.

3.1.4 Cadenas porta cables

El robot contaba con cadenas para el acomodo de los cables (figura 22a) pero al ser de dimensiones pequeñas varios cables estaban por fuera y mal acomodados, lo cual puede propiciar a fallas futuras. Para solucionar este problema se tuvieron que cambiar las cadenas por otras más grandes con dimensiones más apropiadas, lo cual llevó a tener que recablear todo el sistema para una mejor estética del robot (figura 22b).

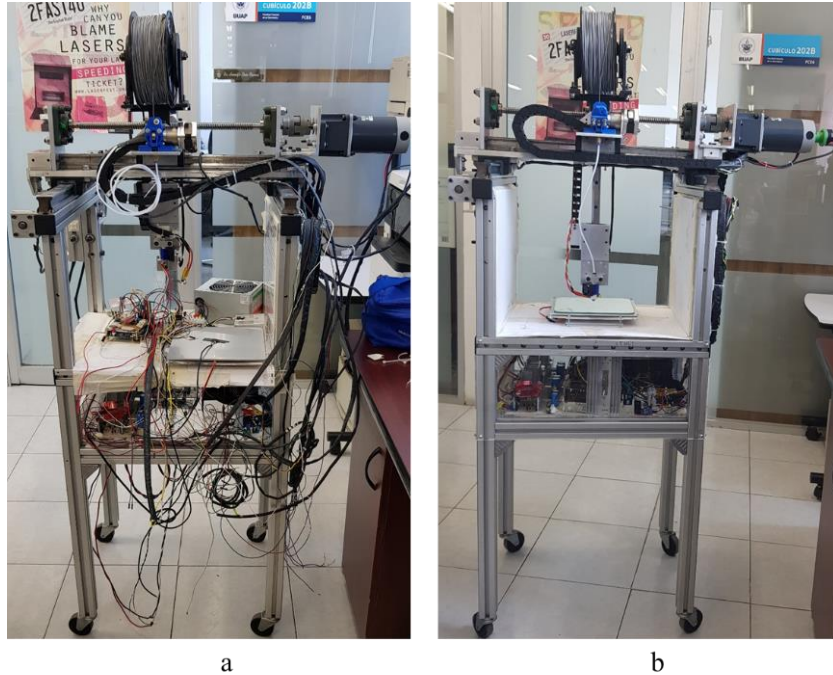


Figura 22. Robot CNC con 3 gdl: a) Robot inicial, b) Robot recableado.

3.1.5 Motores DC y reductores

En cuanto a los actuadores de los 3 gdl (ejes X, Y y Z) se eligieron unos motores de uso robusto ya que a futuro se pretende hacer una modificación al robot para que funcione como multiherramienta, es decir, para que también cuente con la función de devastado, etc. Por lo tanto, se implementaron motores reductores Z5D120-12 (figura 23) para los ejes X y Y, los cuales se alimentan a 12V, cuentan con la característica de proporcionar 3100 rpm (revoluciones por minuto) y generan un torque de 3-4 Nm; en la tabla 4 se muestran las características de este motor. De igual manera, a estos motores se le acoplaron reductores 5GU9K cuyas características se muestran en la tabla 5.



Figura 23. Motor Z5D120-12 con reductor 5GU9K.

Tabla 4. Características del motor Z5D120-12.

Modelo	Voltaje	Potencia	Parámetros sin carga		Parámetros con carga		Peso del motor
			Velocidad	Corriente	Velocidad	Corriente	
Z5D120-12	12 V	120 W	3100 rpm	2 A	2500 rpm	15 A	2.2 kg

Tabla 5. Características del reductor 5GU9K.

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

Acoplando el reductor al motor se tienen las siguientes características:

- Velocidad de salida del motorreductor: 333 rpm.
- Torque: 3.21 Nm [18].

Para el eje Z se acopló otro tipo de motor, el modelo Z4D40-12GN-30S (figura 24), que cuenta con las características mostradas en la tabla 6; a este motor se le adaptó el reductor 4GN15K cuyas características se presentan en la tabla 7.



Figura 24. Motor Z4D40-12GN-30S con reductor 4GN15K.

Tabla 6. Características del motor Z4D40-12GN-30S.

Modelo	Voltaje	Potencia	Parámetros sin carga		Parámetros con carga		Peso del motor
			Velocidad	Corriente	Velocidad	Corriente	
Z4D40-12GN-30S	12 V	40 W	3100 rpm	2 A	2800 rpm	5.2 A	1.8 kg

Tabla 7. Características del reductor 4GN15K.

Tamaño	Tipo de eje	Relación	Diámetro del eje de salida
3.15 pulgadas	Eje paralelo a la cabeza del reductor	15:1	10 mm

Acoplando el reductor al motor se tienen las siguientes características:

- Velocidad de salida: 190 rpm.
- Torque: 1.01 Nm [19].

3.1.6 Encoders

Para poder ocupar los motores DC como servomotores, en un principio se les acopló un encoder giratorio GTS06-VE-RA 1000 de 1000 pulsos/revolución en la parte posterior de cada motor, mediante el cual se puede determinar la posición actual del actuador. Desafortunadamente, debido al mal uso que le dieron a estos encoders (conexión errónea) dejaron de funcionar correctamente, lo cual ocasionaba tener pérdida en las cuentas y así obtener la posición actual del actuador de manera ineficiente.

Por lo tanto, ya que para la implementación del robot cartesiano como impresora 3D se necesita tener buena precisión para una mejor calidad de impresión, se tuvieron que reemplazar los encoders por unos nuevos de modelo E6B2-CWZ6C (figura 25a) de 1000 pulsos/revolución. Las características de este encoder se muestran en la tabla 8.

Tabla 8. Características del encoder E6B2-CWZ6C.

Descripción	Requerimiento
Resolución	1000 pulsos/revolución
Alimentación	5-24 V
Consumo de corriente	80 mA máx.
Fases de salida	A, B y Z
Configuración de salida	NPN colector abierto (figura 25b)

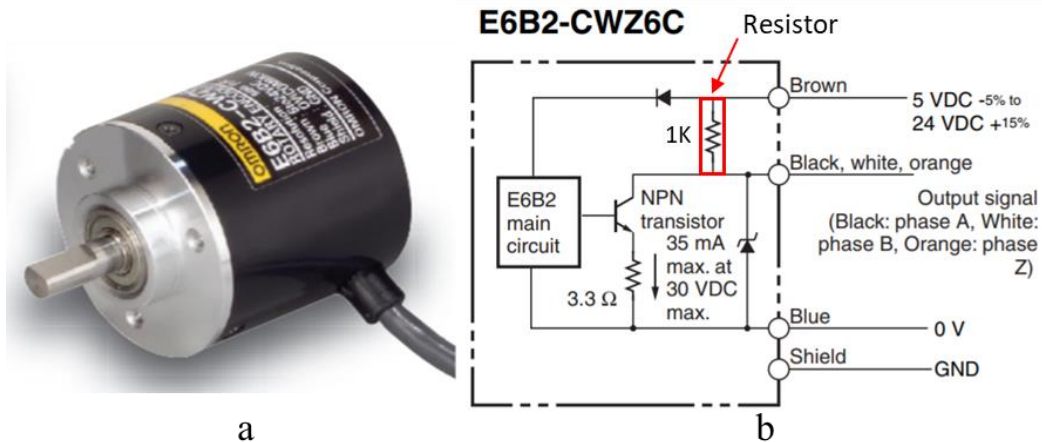


Figura 25. Encoder giratorio: a) E6B2-CWZ6C, b) Conexión para encoder con salida NPN colector abierto.

3.1.7 Herramienta de extrusión

El proceso de extrusión hace referencia a cualquier operación de transformación, esto consiste en hacer pasar de manera forzada un material fundido a través de una boquilla o dado que determinará su forma y espesor, es decir, para producir un artículo de sección transversal constante y de longitud indefinida. Este proceso se lleva a cabo para la fabricación tanto de plásticos como de metales, cerámicas o alimentos.

3.1.7.1 Porta carrete

Ya que este robot no contaba con un soporte donde colocar el rollo de filamento que se empleará para el proceso de extrusión, se tuvo que realizar la construcción del porta carrete (figura 26) para sujetar dicho rollo. Este porta carrete se diseñó de tal manera que soporte 1 Kg (que es lo que normalmente pesan los rollos de filamento) y se colocó sobre el eje Z que es donde se tiene el extrusor de material.

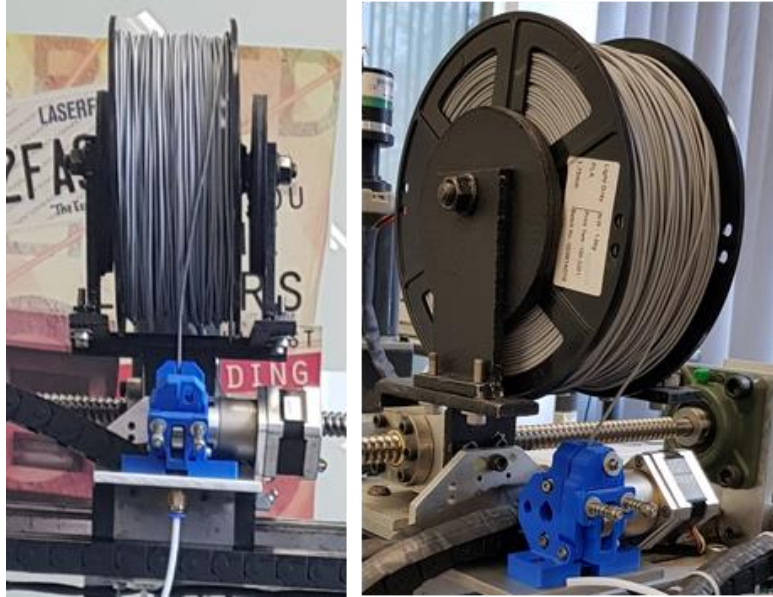


Figura 26. Porta carrete para rollo de filamento.

3.1.7.2 Extrusor de material

Se contaba con un dispositivo denominado Metal J-Head pero debido a las malas condiciones de este se tuvo que cambiar por otro nuevo (figura 27a), el cual será utilizado como extrusor de material que será alimentado con filamento de plástico PLA para llevar a cabo la impresión en 3D. Sus características se presentan en la tabla 9.



Figura 27. Extrusor de material: a) Metal J-Head, b) Partes del extrusor.

Tabla 9. Especificaciones del extrusor de material Metal J-Head.

Descripción	Requerimiento
Material	Metal
Material de impresión	PLA o ABS de 1.75 mm
Resolución de impresión	0.1 mm
Diámetro de la boquilla	0.3 mm
Temperatura de la boquilla	180-240 °C
Voltaje	12 V
Tubo de PTFE	
Material	Teflón
Longitud	1 m
Diámetro exterior	6 mm
Diámetro interior	4 mm
Transductor	
Termistor	100K NTC
Resistencia de cartucho	12 V, 30 W
Longitud del cable	1 m

El cable Bowden (figura 27b) permite reducir la masa en movimiento de la extrusora, por lo cual permite un movimiento más rápido y controlado, menos agitación de la máquina, menos consumo de energía y lo que es más importante: una impresión más rápida. Normalmente, el mecanismo que impulsa el filamento hacia el extremo caliente (donde se derrite) está directamente sobre la extrusora. Esto crea problemas de equilibrio y oscilaciones con un movimiento más rápido que se puede ver en los resultados de impresión.

Si el mecanismo de accionamiento del filamento se coloca en una parte no móvil de la impresora 3D, puede insertarse en un tubo PTFE (teflón), el cual es útil porque es resbaladizo (tiene poca fricción con el plástico). Esto limita el desgaste y la pérdida de energía. El otro extremo del tubo está conectado al extremo caliente del extrusor.

La principal ventaja del Metal J-Head es que por ser de acero inoxidable, todo el metal le permite trabajar con policarbonato hasta 300 °C y con materiales futuros que requieren temperaturas de trabajo más altas que las admitidas por extrusoras de plástico y extremo caliente [20].

Este extrusor trae integrado una resistencia eléctrica para calentarse y un termistor mediante el cual se determina la temperatura actual del extrusor. El termistor es recomendado utilizarlo como sensor de temperatura, ya que, por lo general, presenta una sensibilidad muy alta ante los cambios de temperatura. Este termistor es un semiconductor térmicamente sensible cuya resistencia varía su valor en función de la temperatura a la que se ve sometido. Por ser del tipo NTC (Negative Temperature Coefficient), su coeficiente de temperatura es negativo, es decir, que su resistencia disminuye al aumentar la temperatura.

3.1.7.3 Alimentador de material

Para alimentar al extrusor de material y lograr realizar la impresión 3D, es necesario implementar un alimentador de material que, para este caso, se hará mediante un motor a pasos bipolar engranado con cambio rápido de filamento. El modelo de este motor es 1040220 de la marca Kysan Electronics (figura 28). Sus características se presentan en la tabla 10.

Una de sus principales características 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 de sus características es el cambio rápido de filamento, este le permite cambiar rápidamente el filamento con solo unas pocas vueltas de un solo tornillo de mariposa.

Con sus increíbles engranajes 14:1, la extrusora proporciona un tremendo torque y apriete de filamentos para filamentos de 1.75 mm y 3 mm, como PLA o ABS. La relación de engranaje 14:1 permite un consumo de corriente mucho más bajo desde el controlador paso a paso mientras se extruye y permite un rango más amplio de corriente aceptable desde los potenciómetros del controlador paso a paso. La gran cantidad de torque le permite apretar la rueda loca firmemente contra el filamento, sin tener que preocuparse por el sobrecalentamiento del motor [21].

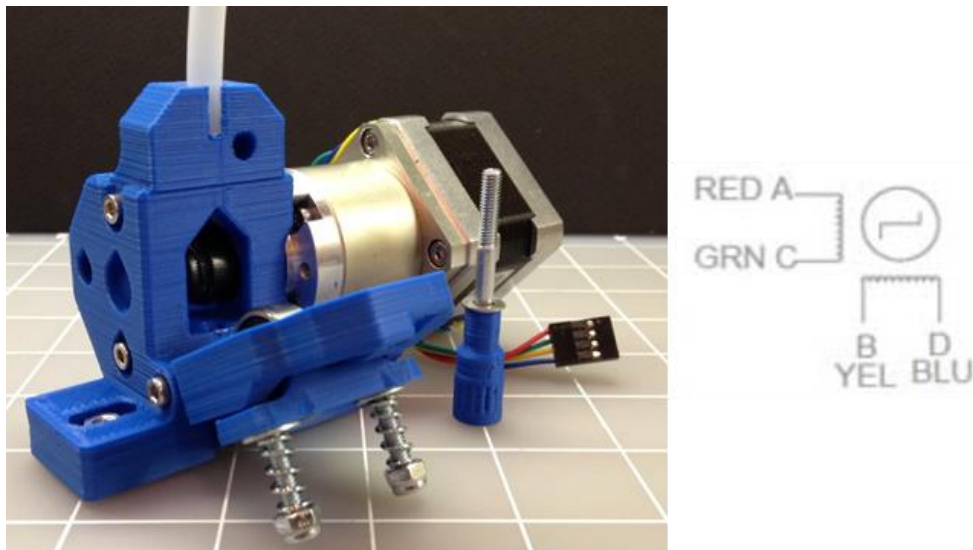


Figura 28. Motor a pasos bipolar con impulsor de filamento y configuración de bobinas.

Tabla 10. Características del alimentador de material.

Descripción	Requerimiento
Ángulo de paso	$1.8^{\circ} \pm 5\%$
Voltaje nominal	2.6 V
Corriente nominal	1 A
Resistencia por fase	$2.6 \Omega \pm 10\%$
Número de fases	2
Torque	24 kg·cm
Relación de transmisión	14:1

3.1.7.4 Cama caliente

Finalmente, se le implementó una cama caliente modelo MK3 ALU-Heatbed Dual Power (figura 29) en la cual se irá depositando el filamento extruido para ir formando la pieza en 3D, esto con el fin de mantener el material caliente durante la impresión y así evitar pandeo en la pieza. Esta cama tiene la característica de contar con fuente de alimentación doble, es decir, que se puede operar con 12 V o 24 V. De igual manera, se pueden usar alambres más finos cuando opera a 24 V y contiene un núcleo de aluminio de 3.2 mm que le permite imprimir directamente sobre la base caliente, sin necesidad de una placa de vidrio [22]. Las características del MK3 se presentan en la tabla 11.

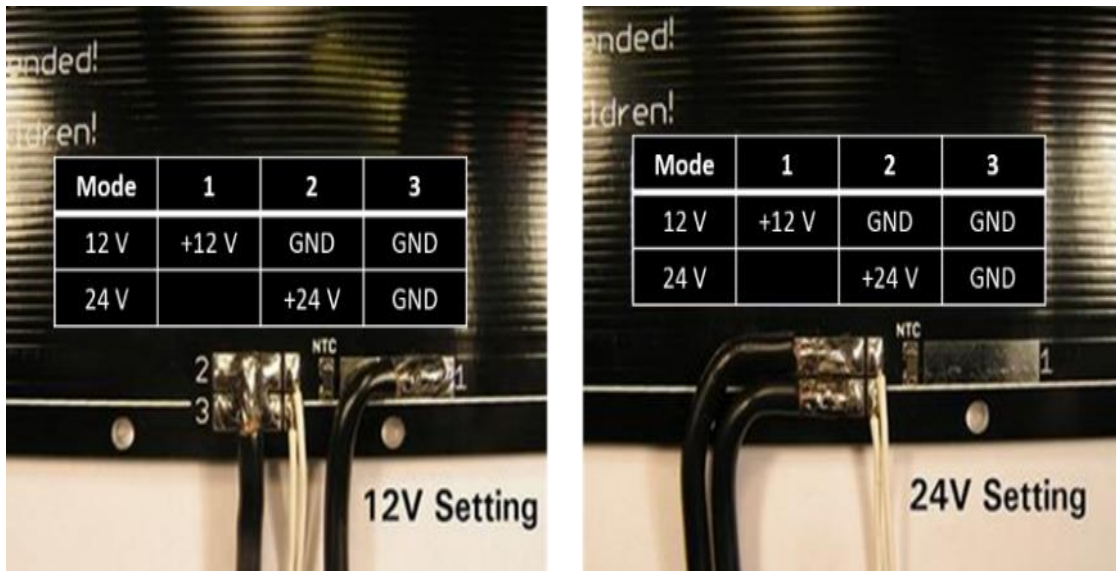


Figura 29. Cama caliente MK3.

Tabla 11. Características del MK3.

Descripción	Requerimiento
Dimensiones	214 mm x 214 mm
1 capa	35 μm (1 oz de base) de cobre
Resistencia para 12 V	entre 1.4 y 1.6 ohmios
Resistencia para 24 V	entre 5.0 y 5.4 ohmios
Temperatura	100 °C posible para 12V y 24V
Temperatura máxima	180 °C
Al ejecutar 24V en los terminales de 12V calentará la base a 100° en solo 2 minutos	

Al igual que el extrusor, esta cama también trae integrada una resistencia eléctrica para calentarse y un termistor del tipo NTC mediante el cual se determina la temperatura actual de la cama.

3.2 Electrónica de potencia

En este apartado se describen los circuitos electrónicos y se explican las etapas de potencia que se desarrollaron para controlar cada planta del sistema mecatrónico.

3.2.1 Etapa de potencia para los motores DC

En cuanto a la electrónica necesaria para activar los motorreductores de cada grado de libertad se implementaron puentes H en configuración Darlington que al final activarán unos transistores Tip35C (NPN) y Tip36C (PNP), los cuales pueden drenar hasta 25A en modo saturación y corte. Para realizar la configuración Darlington se utilizaron transistores convencionales, 3904 (NPN) y 3906 (PNP), sin embargo, para poder saturar los transistores Tip35C y 36C se adaptaron otros transistores Tip35C y 36C en modo amplificación ya que se requiere de 1.5A para saturar a 15A, que es la corriente de consumo de cada motor.

En la figura 30 se muestra el diagrama esquemático del puente H implementado para cada motorreductor. Como se observa en este diagrama, se tiene una configuración resistencia-capacitor (red snubber) conectado al motor, esto con el propósito de reducir los sobretiros a la hora de energizar el motor durante la conmutación de la señal. De igual manera, se maneja una compuerta AND3 la cual recibe la señal del FPGA para después pasarla a la etapa de potencia sirviendo como etapa de acoplamiento de señal.

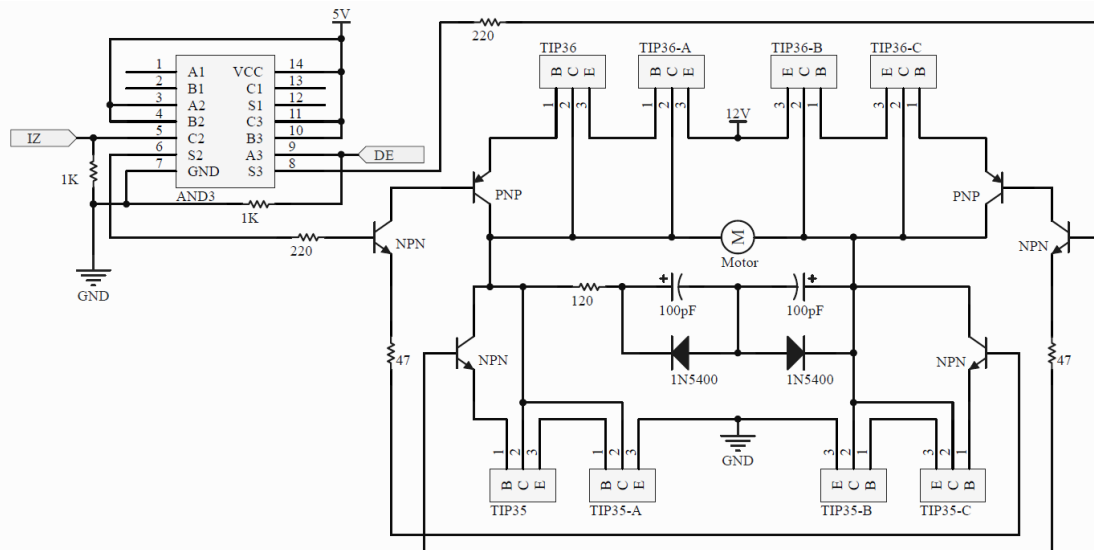


Figura 30. Diagrama esquemático del puente H.

3.2.2 Circuito para el acoplamiento de señales de los encoders

Debido a que el encoder tiene la característica de que su configuración de salida es del tipo NPN colector abierto, para obtener una buena lectura de las señales de cuadratura A y B, se le tuvo que conectar a cada canal una resistencia de 1K a Vcc (5V). Posteriormente, se hizo un divisor de voltaje, ya que los encoders manejan un voltaje de 5V en sus señales de salida y la FPGA solo puede leer señales dentro del rango de 3.2-3.4V, de esta forma no se

daña la FPGA y se puede hacer la lectura de las señales satisfactoriamente. En la figura 31 se muestra el circuito para el acoplamiento de las señales de cada encoder.

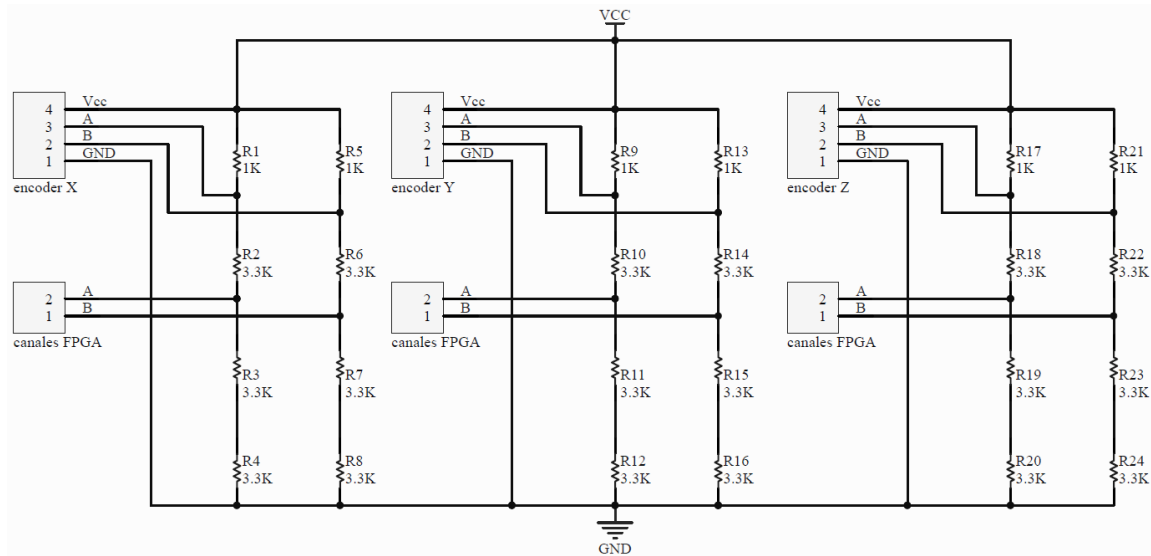


Figura 31. Diagrama esquemático para las señales de los encoders.

En la figura 32 se presentan las señales de cuadratura de los canales A y B, en las cuales se observa que los canales están desfasados 90° y trabajan dentro del rango de voltaje requerido.

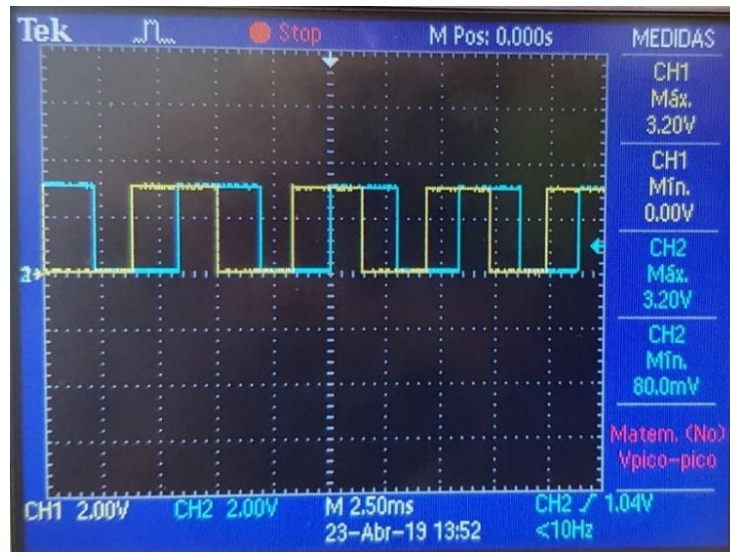


Figura 32. Señales de cuadratura de los canales A y B.

3.2.3 Etapa de potencia para el extrusor de material

Para activar el extrusor mediante una señal PWM que manda la FPGA de 3.3V a una frecuencia de 100Hz, se diseñó un circuito de potencia con transistores NPN en configuración Darlington para amplificar la corriente de 1mA que manda la señal de la FPGA y poder drenar 2.5A para controlar la resistencia del extrusor, como se muestra en la figura 33.

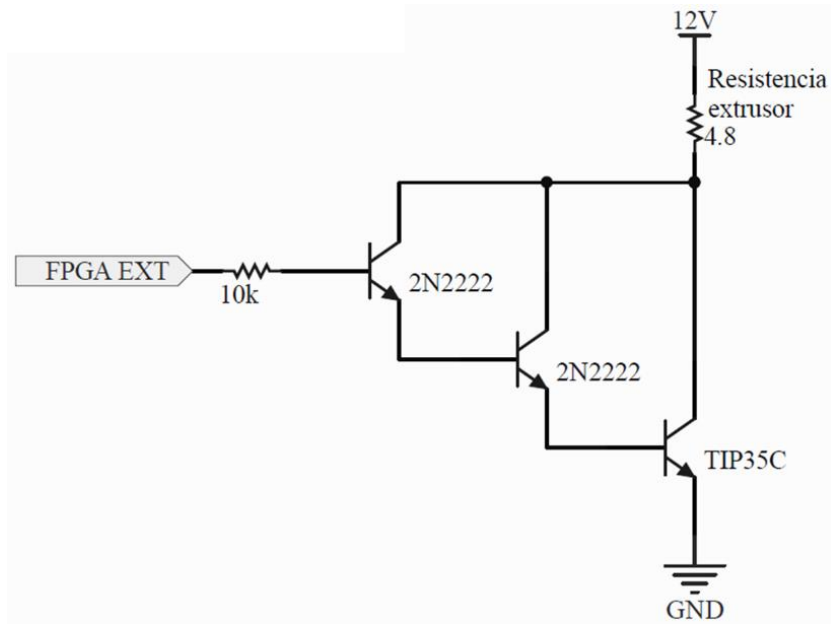


Figura 33. Circuito de potencia para el extrusor de material.

Para saber si la configuración Darlington es la ideal para drenar los 2.5A que se necesitan para activar el extrusor, se calculó la corriente que pasa por cada transistor (utilizando la Ley de Ohm y la Ley de las corrientes de Kirchoff en [23]) hasta llegar a la señal que es activada por la salida de la FPGA.

Primero, empezando con el TIP35C, se determina la corriente de base en modo saturación. Para lograr la saturación se necesita forzar la beta (ganancia), es decir, utilizar una beta mucho menor a la real; ya que se trata de un transistor de alta potencia su beta va de 15 a 75, para este caso se utilizará una beta forzada de 10.

$$I_B = \frac{I_C}{\beta_F} = \frac{2.5A}{10} = 0.25 A \quad (32)$$

Continuando con el transistor 2N2222 (de en medio), se toma la corriente de base (32) como la corriente de colector para este transistor ya que es la que se necesita drenar para que el TIP35C se sature. Para este transistor su hoja de datos presenta una beta de 100 a 300, por lo tanto, se tomará una beta forzada de 50.

$$I_B = \frac{I_C}{\beta_F} = \frac{0.25A}{50} = 0.005 A \quad (33)$$

Esta corriente activará el transistor 2N2222 (de en medio), por lo tanto, pasa a ser la corriente de colector del siguiente transistor 2N2222 (izquierda) que será activado por la FPGA, para el cual se calcula su corriente de base en modo saturación:

$$I_B = \frac{I_C}{\beta_F} = \frac{0.005A}{50} = 0.1 mA \quad (34)$$

Sabiendo que la salida de la FPGA solo manda señales de 1mA, este circuito en configuración Darlington es suficiente para poder drenar los 2.5A que se necesitan para activar el extrusor de material con una corriente de activación de 0.1mA.

Finalmente, se determina la resistencia de base que es la encargada de activar el circuito. Para esto, se tiene que el voltaje de la FPGA es de 3.3V y se tiene una caída de tensión de 0.7V en cada transistor, por lo tanto, el valor de la resistencia de base es la siguiente:

$$R_B = \frac{V_{BB} - V_{BE_T}}{I_B} = \frac{3.3V - (3 * 0.7V)}{0.1mA} = 12k\Omega \approx \mathbf{10k\Omega} \quad (35)$$

3.2.3.1 Caracterización del extrusor

Ya que el extrusor cuenta con un termistor tipo NTC de 100k, mediante este sensor se podrá determinar la temperatura actual a la que se encuentra el extrusor. Para esto se utilizó un circuito potenciométrico formado por dicho termistor y una resistencia de 10k, colocados en serie a modo de divisor de voltaje, como se muestra en la figura 34.

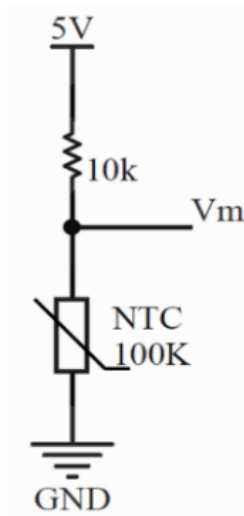


Figura 34. Circuito divisor de voltaje para el extrusor.

Este circuito es alimentado por un voltaje, el cual provoca una tensión de salida en el punto medio del divisor de voltaje (Vm), mediante el cual, al graficarlo con respecto a la temperatura se podrá obtener la ecuación característica a la que trabaja el termistor y así poder determinar la temperatura a la que se encuentra el extrusor.

Para caracterizar el extrusor, la etapa de potencia se activó mediante un generador de funciones a una frecuencia de 100Hz y 3.3V pico pico, que representaría la señal de la FPGA, con esto se fue variando de 10 en 10 el ciclo de trabajo del PWM para activar su etapa de potencia (figura 33), empezando con el 10% hasta el 100% y después de 5 minutos de funcionamiento se obtenía la temperatura actual del extrusor. Para cada intervalo se empezaba con una temperatura de 25 °C.

En la tabla 12 se muestran los resultados obtenidos al realizar la caracterización del extrusor mediante la variación de su ancho de pulso (PWM).

Tabla 12. Datos obtenidos al variar el ciclo de trabajo del extrusor.

Ciclo de trabajo (%)	Temperatura (°C)	Voltaje medido (V)
10	42.2	3.583
20	69.9	1.946
30	83.7	0.927
40	107.7	0.449
50	139.6	0.233
60	165.4	0.129
70	190	0.080
80	211	0.050
90	234.4	0.034
100	225.8	0.026

En la figura 35 se muestra la gráfica de los datos obtenidos de temperatura con respecto al ciclo de trabajo, en esta se puede observar que el funcionamiento del extrusor al variar su ciclo de trabajo es de forma lineal o casi lineal, lo cual permitirá cerrar el lazo de control de manera eficiente.

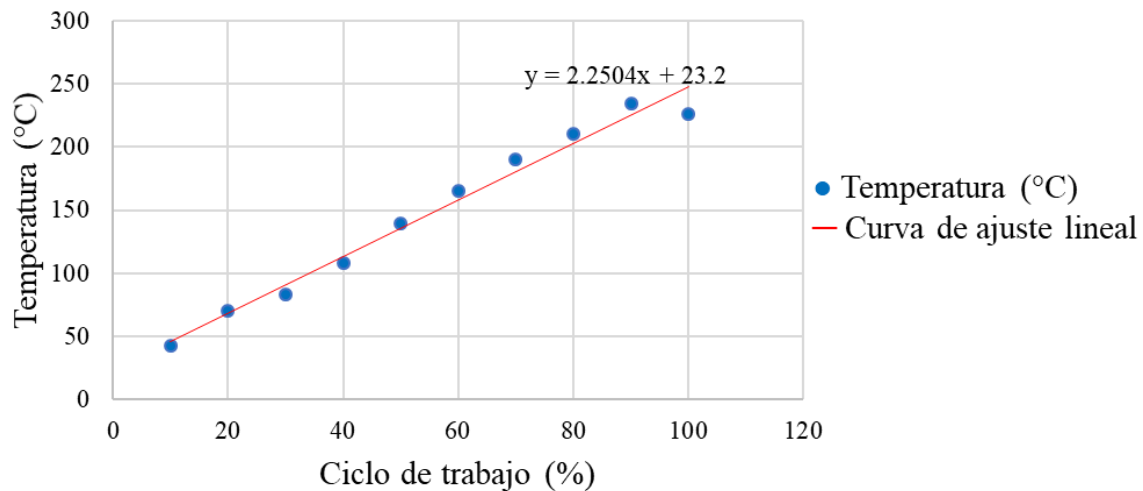


Figura 35. Gráfica de linealidad del extrusor.

Por otra parte, como se puede observar en la tabla 12, los valores obtenidos del voltaje medido disminuyen demasiado conforme la temperatura va aumentando y ya que el extrusor se tiene que controlar a una temperatura entre 190-210 °C se le tuvo que agregar un buffer para amplificar el voltaje de salida y así el ADC lo pueda interpretar de manera más eficiente, como se muestra en la figura 36.

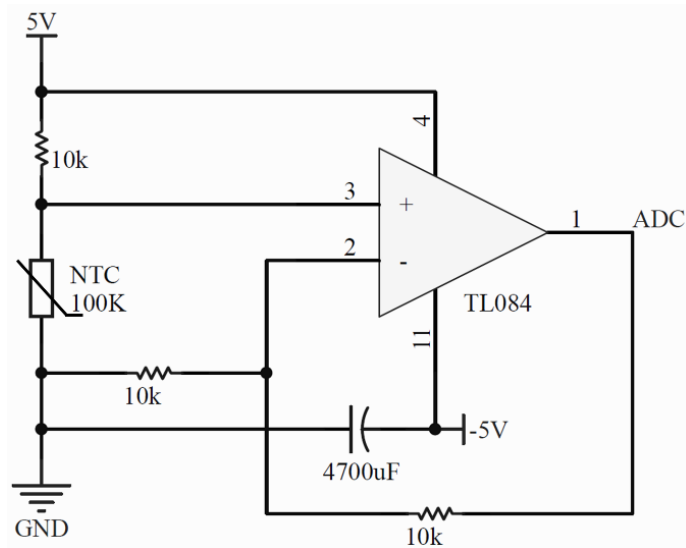


Figura 36. Circuito amplificador para el termistor del extrusor de material.

Para esto, se implementó un amplificador no inversor [24] con ganancia de 2, la cual se calculó mediante la ecuación (36):

$$A = \frac{V_o}{V_i} = \left(1 + \frac{R_f}{R_i}\right) = \left(1 + \frac{10k}{10k}\right) = 2 \quad (36)$$

Por lo tanto, al multiplicar los voltajes medidos de la tabla 12 por la ganancia ($A=2$), se obtienen los valores mostrados en la tabla 13:

Tabla 13. Datos del voltaje amplificado.

Ciclo de trabajo (%)	Temperatura (°C)	Voltaje medido (V)	Voltaje amplificado (V)
10	42.2	3.583	7.166
20	69.9	1.946	3.892
30	83.7	0.927	1.854
40	107.7	0.449	0.898
50	139.6	0.233	0.466
60	165.4	0.129	0.258
70	190	0.08	0.16
80	211	0.05	0.1
90	234.4	0.034	0.068
100	225.8	0.026	0.052

En las figuras 37 y 38 se puede observar que el comportamiento del termistor del extrusor no es lineal, por lo tanto, para determinar su ecuación de ajuste para poder calcular su temperatura actual se proponen 2 curvas de ajuste: la logarítmica (figura 37) y la potencial (figura 38), ambas gráficas presentan su ecuación característica.

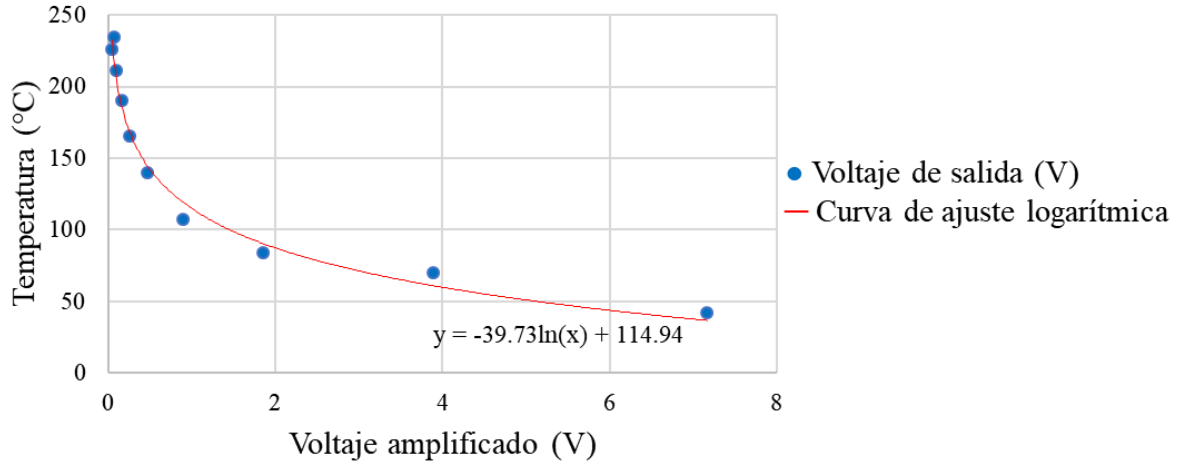


Figura 37. Gráfica de voltaje medido contra temperatura con curva de ajuste logarítmica.

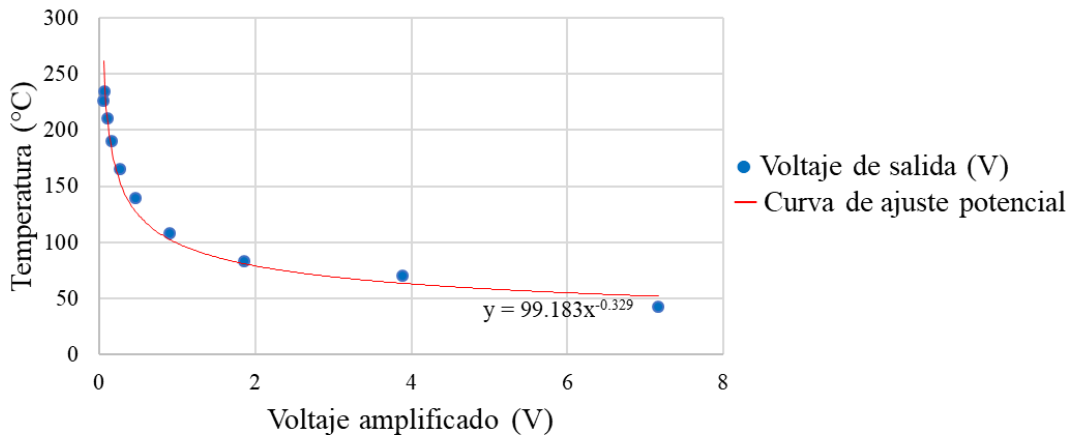


Figura 38. Gráfica de voltaje medido contra temperatura con curva de ajuste potencial.

Ya que a simple vista no se puede determinar que curva es la más adecuada, mediante la ecuación de mínimos cuadrados (37) se calculó el error de cada ecuación para así poder elegir que curva de ajuste es la mejor al presentar el menor error.

$$\overline{\Delta y} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y - y_i)^2} \quad (37)$$

Donde,

- n es el número de mediciones que se hicieron, en este caso fueron 10.
- y es la temperatura medida (experimental).
- y_i es la temperatura que se calculó con su respectiva ecuación de ajuste.

Como se puede observar en la tabla 14, la ecuación de ajuste que presenta menor error al ser evaluada mediante la ecuación de mínimos cuadrados es la ecuación logarítmica, por lo tanto, esa será la ecuación de ajuste que se implementará para cerrar el lazo de control.

Tabla 14. Cálculo del error mediante mínimos cuadrados.

Experimental		Ecuación de ajuste		Mínimos cuadrados	
Vm (V)	T (°C)	$y = -39.73\ln(x) + 114.94$	$y = 99.183x^{-0.329}$	Logarítmico	Potencial
3.583	42.2	9.1588191	41.31564758	1091.719635	0.782079204
1.946	69.9	33.4109827	50.50492972	1331.448385	376.1687511
0.927	83.7	62.8738646	64.46033105	433.7279161	370.1648612
0.449	107.7	91.6753604	81.82249495	256.7890737	669.6452678
0.233	139.6	117.737622	101.5317019	477.9635722	1449.195318
0.129	165.4	141.227033	123.3325609	584.3323372	1769.66943
0.08	190	160.209462	144.3261938	887.4761809	2086.096574
0.05	211	178.882706	168.4614881	1031.52059	1809.524994
0.034	234.4	194.205076	191.2516543	1615.631907	1861.779736
0.026	225.8	204.863204	208.8986016	438.3494144	285.6572694
Sumatoria / 10				814.8959012	1067.868428
Raíz				28.54638158	32.6782562

3.2.3.2 Regulador de voltaje

Debido a que el buffer que se utiliza para amplificar la señal del termistor del extrusor (figura 36) necesita ser alimentado con $\pm 5V$, se implementó un módulo LM2596 (figura 39). Este módulo es un convertidor de voltaje DC-DC Step-Down, es decir, es un regulador de tipo conmutado reductor, cuyo voltaje deseado se selecciona mediante un potenciómetro multivuelta. Tiene como función entregar un voltaje de salida constante inferior al voltaje de entrada frente a variaciones del voltaje de entrada o de carga. Soporta corrientes de salida de hasta 3A, voltaje de entrada entre 4.5V a 40V y voltaje de salida entre 1.23V a 37V [25].

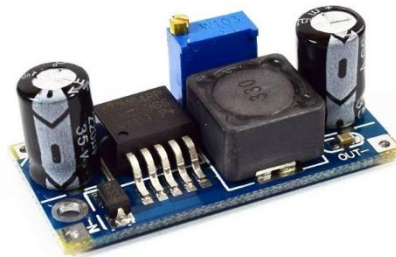


Figura 39. Módulo regulador de voltaje LM2596.

Para este caso, se implementaron 2 módulos de este tipo. El diagrama de conexión electrónica se muestra en la figura 40, donde, la figura 40a es para el voltaje de salida de 5V y la figura 40b para el voltaje de salida de -5V.

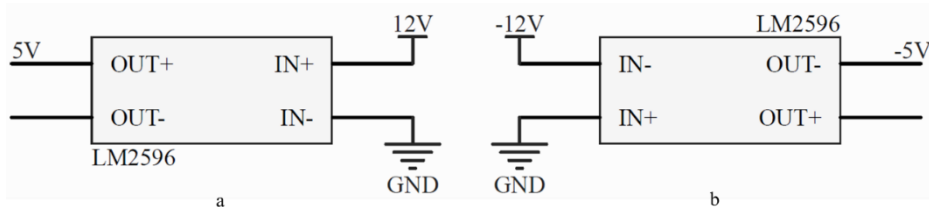


Figura 40. Conexión para los LM2596: a) Voltaje de salida 5V, b) Voltaje de salida -5V.

3.2.3.3 ADC0804 para el extrusor de material

Para que el voltaje medido pueda ser leído por la tarjeta FPGA, de modo que le permita procesar la señal dentro del firmware y mediante el control de temperatura mandar una señal de control a la etapa de potencia por PWM, se requiere de un ADC, en este caso se implementó un ADC0804 de 8 bits de resolución (38), el cual se caracterizó para determinar si su respuesta es lineal. La descripción de sus pines se presenta en la tabla 15.

$$\text{Resolución} = \frac{V_{\text{entrada}}}{2^n - 1} = \frac{5V}{2^8 - 1} = \mathbf{0.0196} \quad (38)$$

$$V_{\text{salida}} = \text{cuentas} * \text{resolución} \quad (39)$$

Tabla 15. Función de los pines del ADC0804.

Pin		I/O	Descripción
No.	Nombre		
1	CS	I	Selección de chip
2	RD	I	Lectura
3	WR	I	Escritura
4	CLK IN	I	Entrada externa de reloj o uso de reloj interno con elementos RC externos
5	INTR	O	Solicitud de interrupción
6	V _{IN(+)}	I	Entrada diferencial analógica +
7	V _{IN(-)}	I	Entrada diferencial analógica -
8	A GND	I	Pin de tierra analógica
9	V _{REF/2}	I	Referencia de voltaje de entrada para ajustar correctamente la escala de lectura
10	D GND	I	Pin de tierra digital
11	DB7	O	BIT 7
12	DB6	O	BIT 6
13	DB5	O	BIT 5
14	DB4	O	BIT 4
15	DB3	O	BIT 3
16	DB2	O	BIT 2
17	DB1	O	BIT 1
18	DB0	O	BIT 0
19	CLK R	I	Pin de entrada de resistencia RC de temporización para generador de reloj interno
20	V _{CC} (o V _{REF})	I	Voltaje de alimentación +5V

Para que el ADC0804 se mantenga leyendo constantemente, es decir, sin tener que esperar la señal de activación para convertir el dato de entrada, se implementó el diseño del circuito con conexión en modo continuo que presenta su hoja técnica, como se muestra en la

figura 41. Cabe mencionar que a este circuito se le agregaron algunos filtros para eliminar un poco de ruido y tener una señal más limpia.

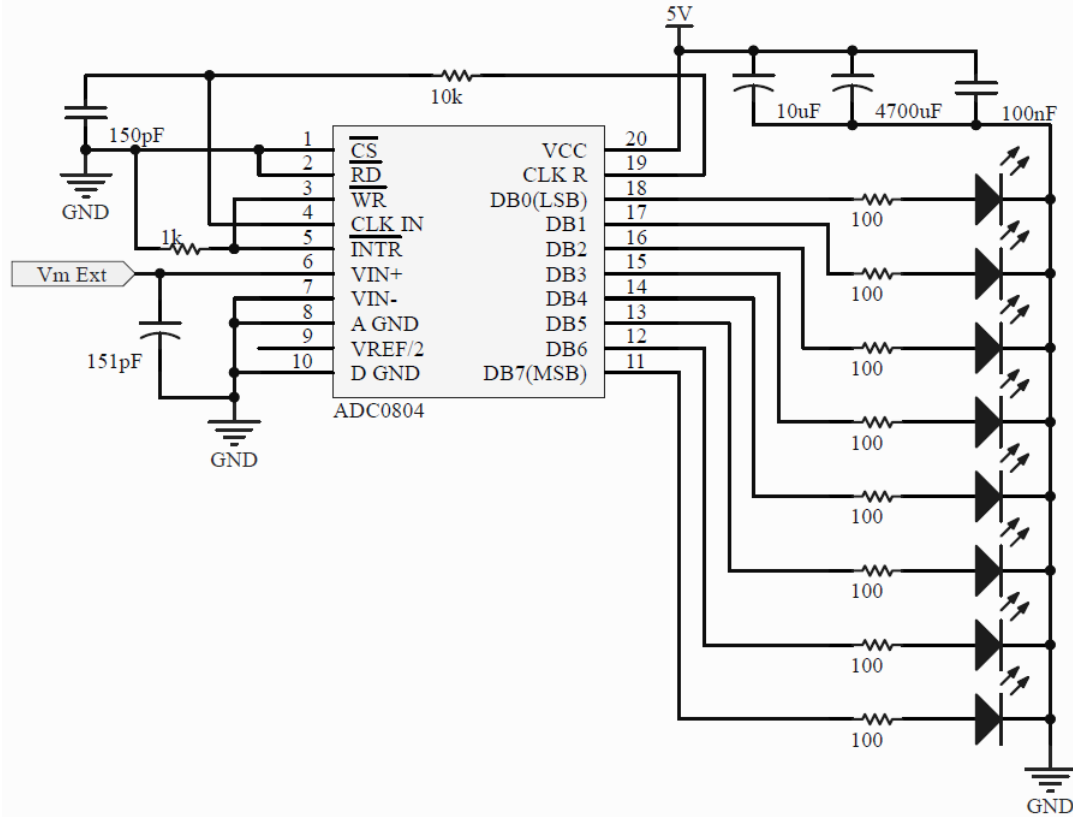


Figura 41. Conexión en modo continuo para el ADC0804 del extrusor.

Para decodificar la señal de salida que se visualiza mediante los 8 leds que representan los 8 bits del ADC, se dividen los 8 bits en dos caracteres hexadecimales, uno con los 4 bits más significativos (MS) y el otro con los 4 bits menos significativos (LS). Después, el voltaje de salida se interpreta como una suma de fracciones multiplicado por el voltaje de alimentación, como se muestra en la ecuación (40) [26].

$$V_{salida} = \left(\frac{MS}{16} + \frac{LS}{256} \right) V \quad (40)$$

Por ejemplo, si a la salida los leds representan un 1100 0110, los bits más significativos (1100) representan la C en hexadecimal (12 en decimal) y los bits menos significativos (0110) representan el 6 en hexadecimal (6 en decimal), entonces, al aplicar la ecuación (40) se tiene:

$$V_{salida} = \left(\frac{12}{16} + \frac{6}{256} \right) * 5V = 3.86 V \quad (41)$$

Finalmente, para poner a prueba el circuito y caracterizar el ADC0804 para determinar su ecuación, a la entrada de este (pin 6) se conectó una fuente variable y se fue regulando el voltaje en incrementos de 0.1V empezando en 0.1V hasta 5V. Con esto se obtuvo la gráfica mostrada en la figura 42, en la cual se puede observar que la respuesta del ADC es lineal, ya que el voltaje de salida es igual o muy cercano al voltaje de entrada.

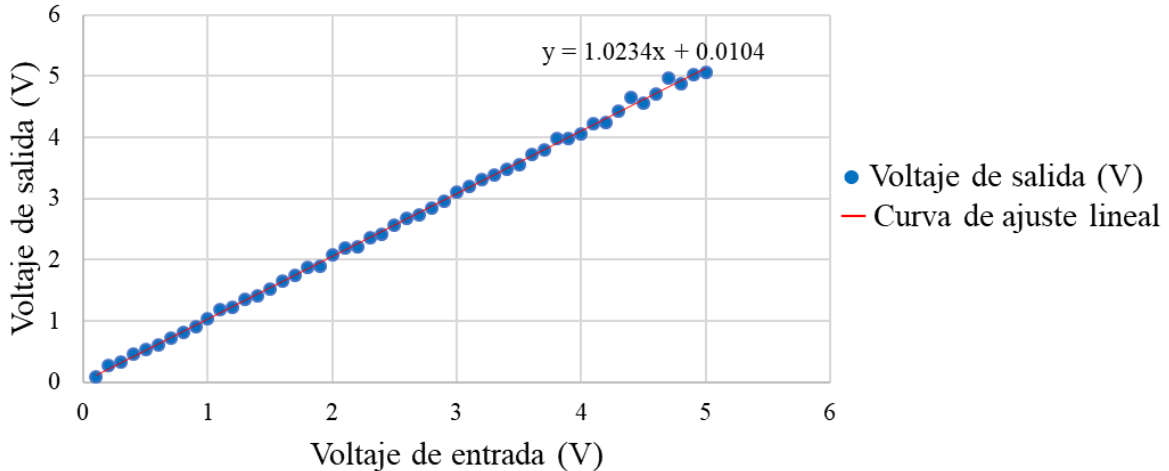


Figura 42. Gráfica de linealidad del ADC0804 del extrusor.

3.2.4 Etapa de potencia para la cama caliente

Al igual que con el extrusor, para activar la cama mediante una señal PWM que manda la FPGA de 3.3V a una frecuencia de 100Hz, se diseñó un circuito de potencia con transistores NPN en configuración Darlington para amplificar la corriente de 1mA que manda la señal de la FPGA y poder drenar 11.66A para controlar la resistencia de la cama caliente o base, como se muestra en la figura 43.

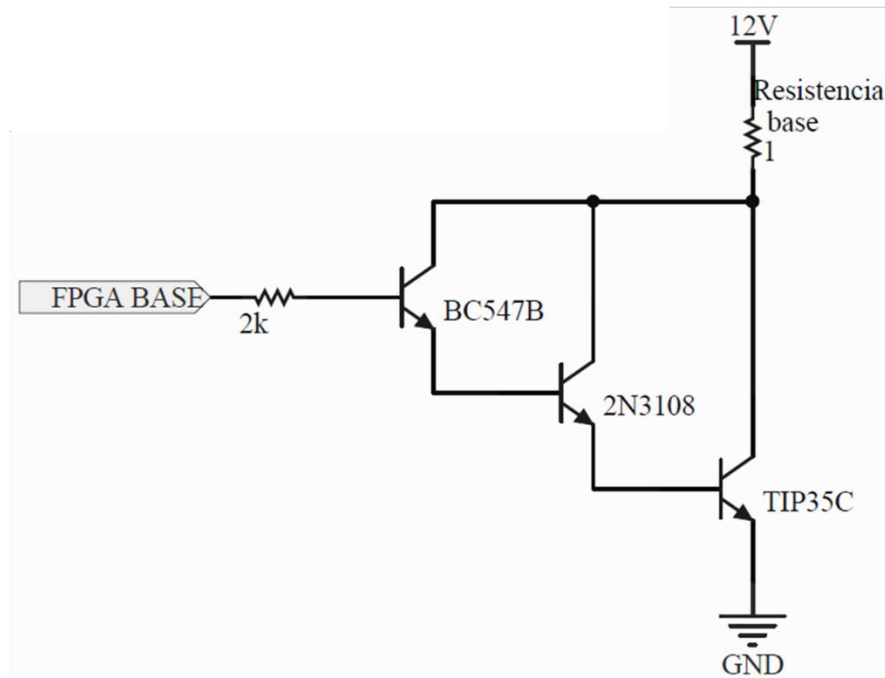


Figura 43. Circuito de potencia para la cama caliente.

Para saber si la configuración Darlington es la ideal para drenar los 11.66A que se necesitan para activar la base, se calculó la corriente que pasa por cada transistor hasta llegar a la señal que es activada por la salida de la FPGA.

Primero, empezando con el TIP35C, se determina la corriente de base en modo saturación. Para lograr la saturación se necesita forzar la beta (ganancia), es decir, utilizar una beta mucho menor a la real; ya que se trata de un transistor de alta potencia su beta va de 15 a 75, para este caso se utilizará una beta forzada de 10.

$$I_B = \frac{I_C}{\beta_F} = \frac{11.66A}{10} = 1.166 A \quad (42)$$

Continuando con el transistor 2N3108, se toma la corriente de base (42) como la corriente de colector para este transistor ya que es la que se necesita drenar para que el TIP35C se sature. Para este transistor su hoja de datos presenta una beta de 20 a 120, por lo tanto, se tomará una beta forzada de 10.

$$I_B = \frac{I_C}{\beta_F} = \frac{1.166A}{10} = 0.1166 A \quad (43)$$

Esta corriente activará el transistor 2N3108, por lo tanto, pasa a ser la corriente de colector del siguiente transistor BC547B que será activado por la FPGA, el cual presenta una ganancia de 200 y al calcular su corriente de base en modo saturación queda como se muestra en la ecuación (44):

$$I_B = \frac{I_C}{\beta_F} = \frac{0.1166A}{200} = \mathbf{0.583 mA} \quad (44)$$

Sabiendo que la salida de la FPGA solo manda señales de 1mA, este circuito en configuración Darlington es suficiente para poder drenar los 11.66A que se necesitan para activar la base con una corriente de activación de 0.583mA.

Finalmente, se determina la resistencia de base que es la encargada de activar el circuito. Para esto, se tiene que el voltaje de la FPGA es de 3.3V y se tiene una caída de tensión de 0.7V en cada transistor, por lo tanto, el valor de la resistencia de base es la siguiente:

$$R_B = \frac{V_{BB} - V_{BE_T}}{I_B} = \frac{3.3V - (3 * 0.7V)}{0.583mA} = 2,058\Omega \approx \mathbf{2k\Omega} \quad (45)$$

3.2.4.1 Caracterización de la cama caliente

Al igual que el extrusor, la cama también cuenta con un termistor tipo NTC de 100k, con el cual se podrá determinar la temperatura actual a la que se encuentra la cama. Mediante el mismo procedimiento que se siguió al caracterizar el termistor del extrusor, para este termistor también se utilizó un circuito potenciométrico formado por dicho termistor y una resistencia de 100k, colocados en serie a modo de divisor de voltaje, como se muestra en la figura 44.

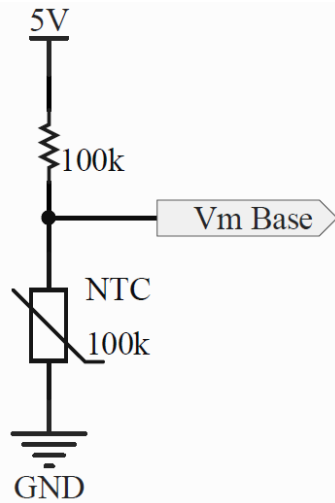


Figura 44. Circuito divisor de voltaje para la cama caliente.

Este circuito es alimentado por un voltaje, el cual provoca una tensión de salida en el punto medio del divisor de voltaje (V_m), mediante el cual, al graficarlo con respecto a la temperatura se podrá obtener la ecuación característica a la que trabaja el termistor y así poder determinar la temperatura a la que se encuentra la cama.

Para proceder con la caracterización del termistor, se adaptó un sensor de temperatura LM35 en el centro de la base mediante el cual se verificaba la temperatura actual a la que este se encontraba. Después, mediante un generador de funciones, se fue variando de 10 en 10 el ciclo de trabajo del PWM para activar su etapa de potencia, empezando con el 10% hasta el 100% y después de 15 minutos de funcionamiento se obtenía la temperatura actual de la base. Para cada intervalo se empezaba con una temperatura de 25 °C.

En la tabla 16 se muestran los resultados obtenidos al realizar la caracterización de la cama mediante la variación de su ancho de pulso (PWM).

Tabla 16. Datos obtenidos al variar el ciclo de trabajo de la cama caliente.

Ciclo de trabajo (%)	Temperatura (°C)	Voltaje medido (V)
10	32.9	2.172
20	40.3	1.745
30	47.7	1.386
40	55.1	1.103
50	60.9	0.909
60	67.4	0.724
70	73.2	0.600
80	78.6	0.496
90	84.1	0.413
100	87.6	0.366

En la figura 45 se muestra la gráfica de los datos obtenidos de temperatura con respecto al ciclo de trabajo, en esta se puede observar que el funcionamiento de la base al variar su ciclo de trabajo es de forma lineal o casi lineal, lo cual es favorable para poder cerrar el lazo de control.

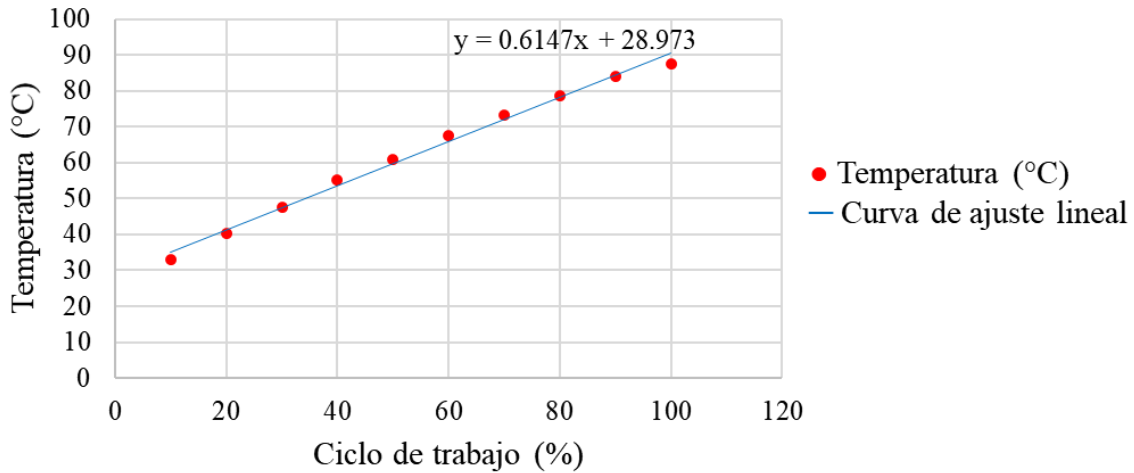


Figura 45. Gráfica de linealidad de la cama caliente.

En la figura 46 se presenta la gráfica de voltaje medido con respecto a la temperatura actual de la base y su ecuación de ajuste. En esta gráfica se puede observar que el comportamiento del termistor de la cama no es lineal, por lo tanto, al determinar su ecuación de ajuste para poder calcular su temperatura actual la que mejor se adaptó fue la curva de ajuste logarítmica.

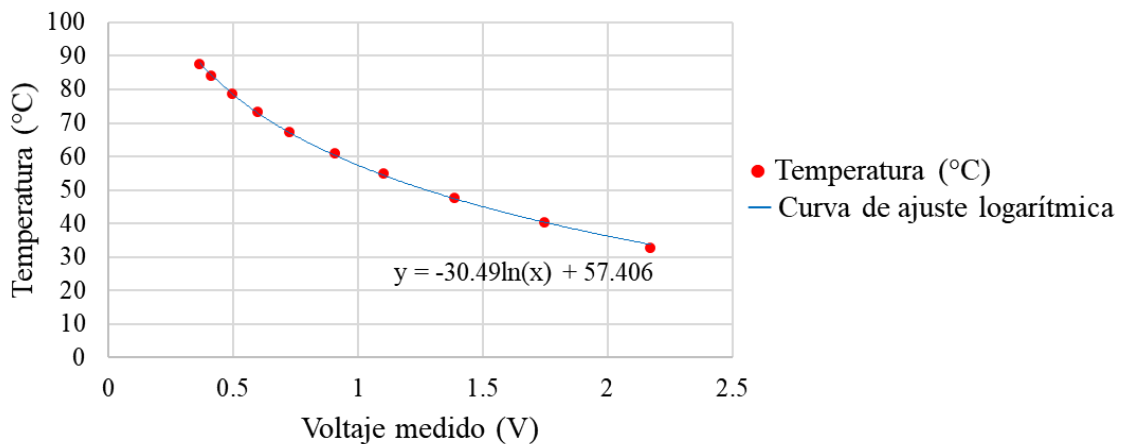


Figura 46. Gráfica de voltaje medido contra temperatura de la cama caliente.

3.2.4.2 ADC0804 para la cama caliente

Para este caso, también se implementó un ADC0804 para que la FPGA pueda interpretar la lectura del termistor y llevar a cabo el lazo de control. Por lo tanto, dicho ADC también se tuvo que caracterizar y al igual que con el ADC del extrusor, para que este se mantenga leyendo constantemente, se implementó el diseño del circuito con conexión en modo continuo que se muestra en la figura 47.

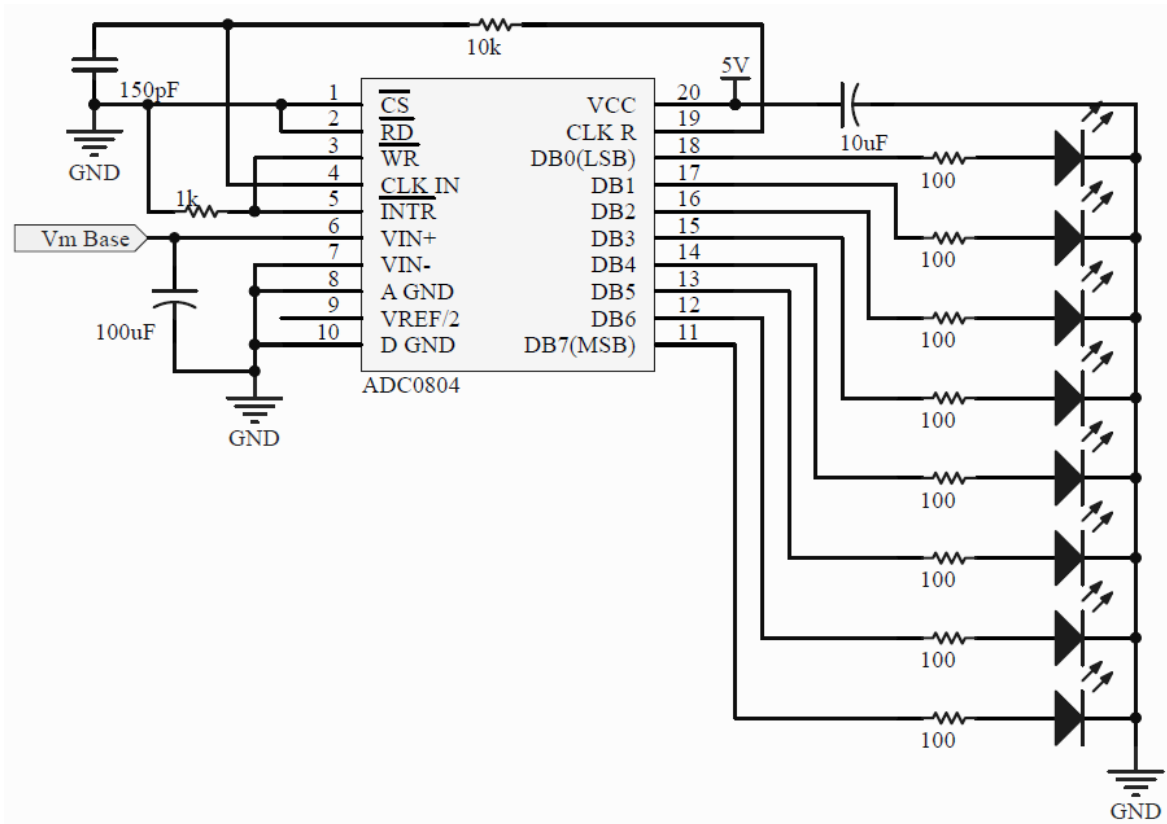


Figura 47. Conexión en modo continuo para el ADC0804 de la cama caliente.

Finalmente, para poner a prueba el circuito y caracterizar el ADC0804 para determinar su ecuación, a la entrada de este (pin 6) se conectó una fuente variable y al igual que con el otro ADC, se regulaba el voltaje en incrementos de 0.1V empezando en 0.1V hasta 5V. Con esto se obtuvo la gráfica mostrada en la figura 48, en la cual se puede observar que la respuesta del ADC es lineal, ya que el voltaje de salida es igual o muy cercano al voltaje de entrada, por lo tanto, este ADC si es apropiado para implementarlo en el sistema.

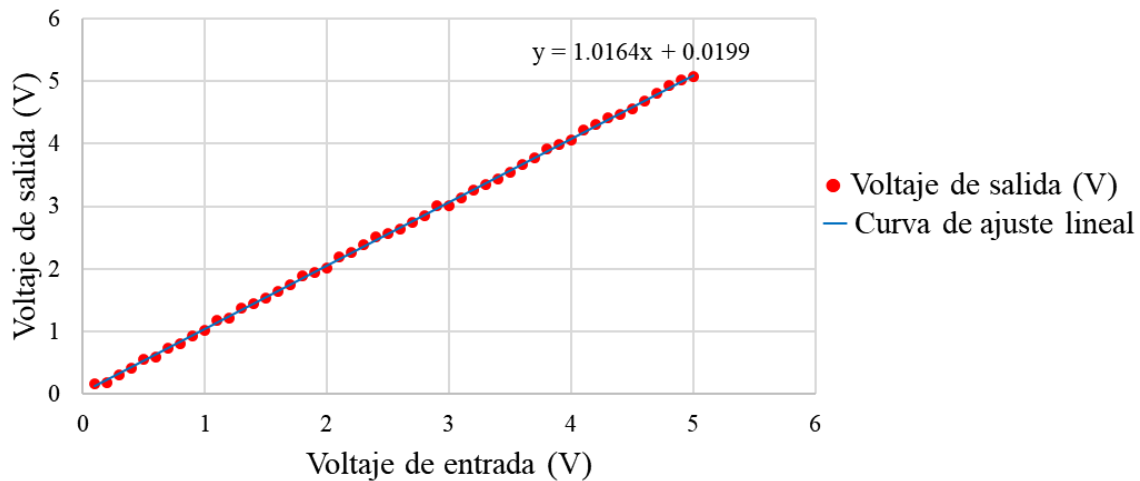


Figura 48. Gráfica de linealidad del ADC0804 de la cama caliente.

3.2.5 Alimentador de material

Ya que se trata de un motor a pasos bipolar, este tipo de motores requieren de un circuito de control externo para su funcionamiento (driver o etapa de potencia), el cual permitirá suministrar energía suficiente al motor para que se mueva. Para esto, se requiere un puente H por cada bobina del motor, por lo tanto, al ser un motor de 2 bobinas es necesario realizar la etapa de potencia con dos puentes H.

Es por eso que, para llevar a cabo el control del giro del alimentador se implementó un circuito integrado L298, ya que en su interior está constituido por dos puentes H completos de alto voltaje y alta corriente y proporciona dos entradas de habilitación para habilitar o deshabilitar el dispositivo independientemente de las señales de entrada.

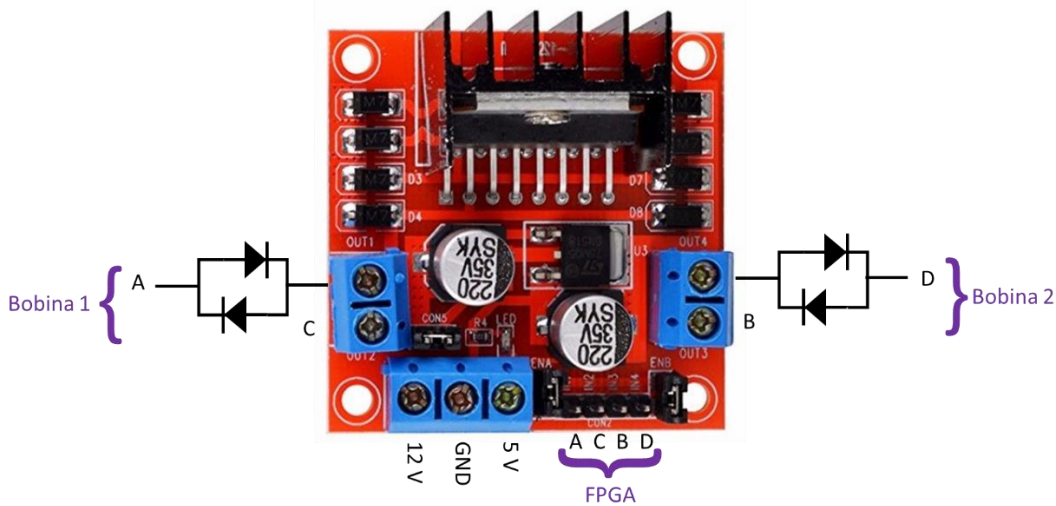


Figura 49. Circuito integrado L298.

Como se muestra en la figura 49, a uno de los canales de cada bobina se le conectó un arreglo de diodos para limpiar la señal que va del puente dichas las bobinas, esto con el fin de poder regular la frecuencia a la que se moverá el motor, ya que en un principio la frecuencia de salida no correspondía a la que se le asignaba en la entrada, ver figura 50.



Figura 50. Arreglo de señal para motor a pasos: a) Señal sin diodos, b) Señal con diodos.

3.2.6 Interruptores de fin de carrera

Estos interruptores se adaptaron uno en cada extremo de cada eje para detectar cuando el carro de precisión llega al final del eje y así evitar que ocasione daños en la mecánica, por lo tanto, al tener 3 ejes, en total en el sistema se implementaron 6 de estos (2 por cada eje). La configuración de estos interruptores es normalmente abierta, es decir, mientras el interruptor esté abierto el motor seguirá girando, de lo contrario, los motores se detendrán cuando el carro de precisión haya llegado al final del eje y presione el interruptor cerrando el circuito, en este caso se le conectó un led como indicador de que el carro llegó al extremo final. En la figura 51 se muestra el diagrama electrónico del circuito diseñado para cada uno de estos interruptores.

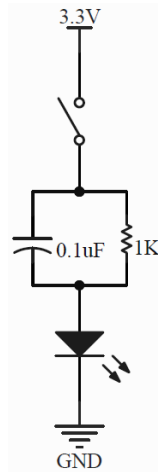


Figura 51. Diagrama esquemático del circuito para los interruptores de fin de carrera.

3.2.7 Módulo Wi-Fi

Se implementa un módulo Wi-Fi RN-XV (figura 52) para hacer la comunicación entre el sistema mecatrónico y la computadora, es decir, mediante este módulo se activará el microprocesador y se le mandarán las instrucciones para llevar a cabo el control del robot.

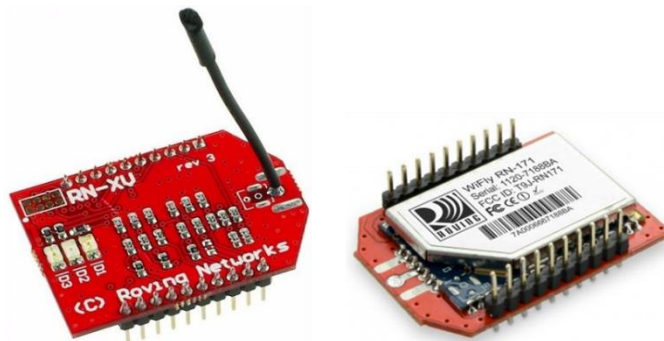


Figura 52. Módulo Wi-Fi.

El módulo "WiFly" es un dispositivo de acceso LAN inalámbrico integrado completo e independiente. Este dispositivo cuenta con protocolo de comunicación TCP/IP y solo requiere de 4 pines (POWER, TX, RX, GND) para su aplicación. Una vez que se aplica la

configuración inicial, el módulo puede acceder automáticamente a la red Wi-Fi y enviar o recibir datos en serie sobre UART a una velocidad de 460,800 bits por segundo [27]. En la tabla 17 se muestran las características de este módulo.

Tabla 17. Características del módulo Wi-Fi.

Parámetro	Especificaciones
Frecuencia	2402 ~ 2480MHz
Modulación	Compatibilidad 802.11b: DSSS 802.11g: OFDM
Intervalos de canal	5MHz
Canales	1 – 14
Rango de transmisión	1 – 11 Mbps para 802.11b 6 – 54Mbps para 802.11g
Sensibilidad de recepción	-83dBm
Nivel de salida (Clase 1)	0dBm a +12dBm (configurable por software)

3.2.8 Tarjeta de implementación

Debido a la ampliación que se le hizo al sistema mecatrónico, también se tuvo que ampliar el firmware necesario para llevar a cabo el control de todo el robot CNC, es decir, el firmware para controlar los motores de los ejes X, Y y Z, para el control de temperatura de la base y del extrusor y para el control de velocidad del alimentador de material.

Al necesitar todo esto, también se requiere de un mayor número de elementos lógicos programables en la FPGA que se utilice, por lo tanto, un firmware con esta magnitud ya no puede ser empleado en la tarjeta que se utilizaba anteriormente (DE0-nano), es por eso que se decidió migrar el sistema embebido a una tarjeta DE0-CV, ver figura 53, ya que cuenta con un mayor número de elementos lógicos.

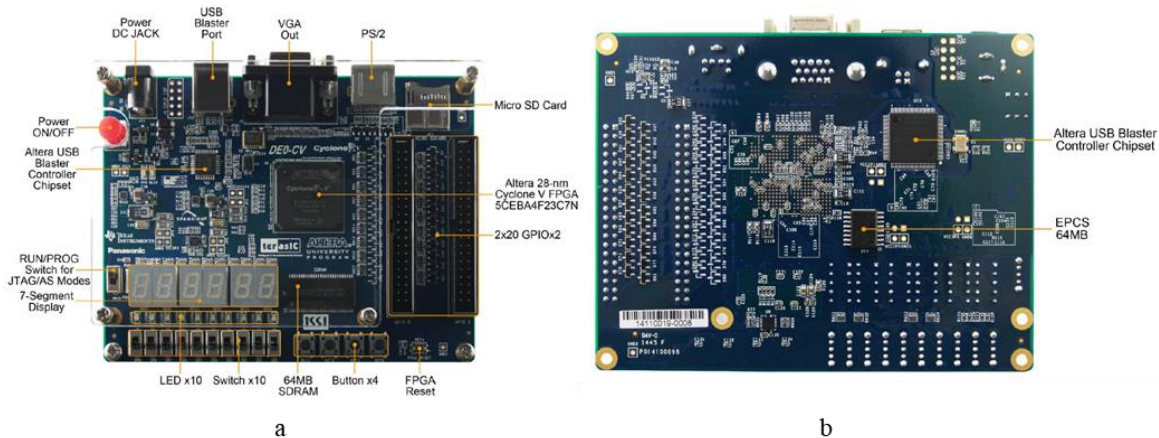


Figura 53. Tarjeta de desarrollo DE0-CV: a) Vista superior, b) Vista inferior.

Las características de la tarjeta DE0-CV son las siguientes [11]:

- Cyclone V FPGA 5CEBA4F23C7N
- 49,000 elementos lógicos programables
- 3080 Kbits de memoria embebida

- Memoria SDRAM de 64MB x 16 bits de bus de datos
- 4 PLLs fraccionales
- Memoria EPCS64
- 2x20 pines de propósito general (72 pines I/O, 2 pines de alimentación de 3.3VCD y 2 de 5VCD con sus respectivas tierras)
- 6 displays de 7 segmentos
- 10 LEDs
- 10 interruptores
- 4 pulsadores de propósito general
- 1 pulsador para el reset de la CPU

3.3 Conclusiones

Describir los componentes mecánicos y electrónicos, así como las etapas de potencia y los circuitos que se desarrollaron e implementaron para el correcto funcionamiento del robot CNC aplicado a impresión 3D, facilitará el poder comprender de manera más clara el firmware y el software que se desarrollaron para lograr controlar de manera eficiente el sistema mecatrónico, los cuales se explicarán en el siguiente capítulo.

Por otra parte, se hace mención del cambio de tarjeta FPGA para implementar el sistema embebido, ya que además de ofrecer un mayor número de elementos lógicos también ofrece la ventaja de contar con una memoria SDRAM la cual se podrá sustituir por las memorias RAM y ROM externas con las que contaba anteriormente para el almacenamiento de datos, ya que la velocidad de acceso a la memoria SDRAM es mucho más rápida, lo cual mejorará la velocidad del microprocesador.

Capítulo 4: Sistema de control

4.1 Firmware

Para llevar a cabo el control del robot se utilizó una tarjeta de desarrollo DE0-CV con FPGA de la familia Cyclone V, cuyo firmware embebido se desarrolló en el software Quartus II mediante lenguaje AHDL. En la figura 54 se muestra el diagrama a bloques general del firmware diseñado para que el robot CNC funcione como impresora 3D.

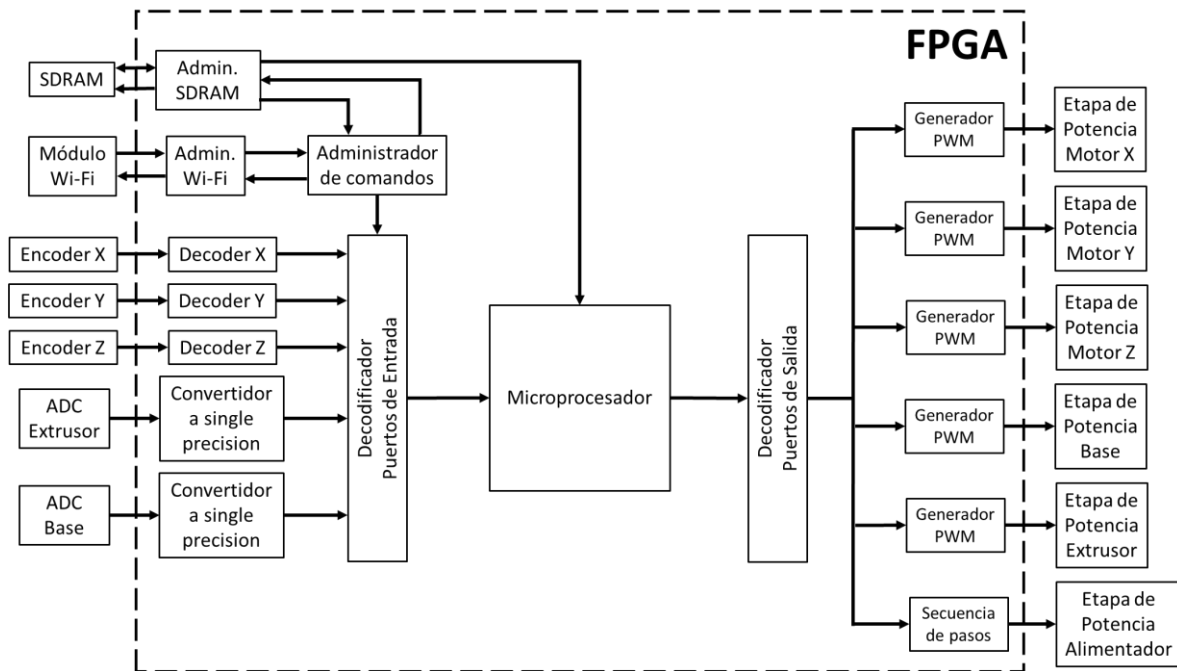


Figura 54. Diagrama general del firmware.

4.1.1 Microprocesador

Para realizar el control del robot se utilizó un microprocesador embebido en la FPGA desarrollado en la MCEA de la BUAP [13]. Este microprocesador se diseñó en base a arquitectura MIPS (Microprocessor without Interlocked Pipeline Stages), el cual es un procesador de 32 bits con arquitectura RISC (Reduced Instruction Set Computer) y cuenta con un banco de 32 registros de 32 bits. El firmware cuenta con protocolo de comunicación Wi-Fi para poder tener comunicación entre una computadora y la tarjeta de adquisición de datos. De igual manera, permite procesar las instrucciones recibidas en lenguaje ensamblador y realizar las correspondientes operaciones aritméticas con el fin de llevar a cabo el control del robot. En la tabla 18 se muestran las instrucciones del microprocesador en lenguaje ensamblador y en la tabla 19 se muestran las variables que se utilizan para estas instrucciones.

Tabla 18. Instrucciones del microprocesador en ensamblador.

Instrucción	Registros, constantes, direcciones y puertos			Descripción
Instrucciones de transferencia				
MOV.S	X	Z		$X \rightarrow Z$
MOVC.S	X			$F,A \rightarrow Y$
RIO.S	P	Z		$P \rightarrow Z$
WIO.S	X	B		$X \rightarrow B$
MOVRX	D	Z		$V(D) \rightarrow Z$
MOVWX	X	D		$X \rightarrow V(D)$
Instrucciones condicionales				
C.EQ.S	X	Y	L	Saltar L líneas si $(X==Y)$
C.LE.S	X	Y	L	Saltar L líneas si $(X<=Y)$
C.LT.S	X	Y	L	Saltar L líneas si $(X>Y)$
Instrucciones aritméticas				
ADD.I	X	Y	Z	$X+Y \rightarrow Z$
SUB.I	X	Y	Z	$X-Y \rightarrow Z$
ADD.S	X	Y	Z	$X+Y \rightarrow Z$
SUB.S	X	Y	Z	$X-Y \rightarrow Z$
MUL.S	X	Y	Z	$X*Y \rightarrow Z$
DIV.S	X	Z		$1/X \rightarrow Z$
EXP.S	X	Z		$EXP(X) \rightarrow Z$
SIN.S	X	Z		$SIN(X) \rightarrow$
CTE.I	A			$A \rightarrow I(n)$
CTE.S	F			$F \rightarrow I(N)$
Instrucciones de salto				
JUMP	E			$CP = E$
Instrucciones de retardo				
DELAY				Retardo de 2.5 ms

Tabla 19. Variables de las instrucciones MIPS.

Campo	Descripción
X	Registro fuente 1
Y	Registro fuente 2
Z	Registro destino
P	Puerto de entrada
B	Puerto de salida
D	Dirección de memoria RAM
L	Número de líneas a saltar
A	Número entero
F	Número flotante
E	Dirección de memoria ROM
CP	Contador de programa
I(n)	Dato en la memoria ROM
V(D)	Dato en la dirección D de la memoria RAM

Por lo tanto, para adaptar el microprocesador a las necesidades del robot CNC aplicado a impresión 3D, al firmware se le tuvieron que agregar los bloques necesarios para procesar las señales de los interruptores, de los encoders y de los ADC, así como también se diseñaron los bloques para generar las señales PWM para activar la etapa de potencia de cada actuador y se le agregó un bloque para la adquisición de datos mediante bluetooth. En el apéndice C se muestra el firmware completo embebido en la tarjeta de desarrollo DE0-CV con FPGA de la familia Cyclone V.

4.1.1.1 Administrador Wi-Fi

El bloque administrador de Wi-Fi permite que la FPGA se comunique con protocolo serial al módulo WiFly, el cual se conecta de forma inalámbrica a la computadora, de esta manera, se logra tener comunicación entre la computadora a través del software de control y el robot CNC para impresión 3D.

De igual manera, el administrador de Wi-Fi tiene comunicación directa con el administrador de comandos y se encarga de activar el microprocesador para dar inicio a la ejecución de las instrucciones en ensamblador y llevar a cabo el control del sistema.

4.1.2 Administrador SDRAM

El microprocesador se comunica directamente con la memoria RAM para leer y escribir datos, por lo tanto, el bloque administrador de memoria SDRAM es un bloque de control de memoria que permite enviar comandos para escribir y leer datos de 32 bits en forma paralela y se encarga de hacer comunicación con los datos que manda el administrador de Wi-Fi, es decir, los datos que se envían mediante la computadora.

4.1.3 Decodificador de señales

Ya que a los servomotores que se utilizan para los 3 gdl se les acoplaron encoders incrementales de cuadratura para poder determinar la posición actual en cada eje, para interpretar las señales A y B de los encoders, cuyas señales están desfasadas 90°, al firmware se le agregaron bloques para decodificar dichas señales, donde se lleva a cabo el conteo de los pulsos de los encoders para poder determinar la posición actual de cada eje y mediante dichos conteos se puede determinar el sentido de giro de cada motor.

En las señales A y B se encuentra codificada la información correspondiente al avance y su dirección, a partir de ese valor cualquier cambio que haga el encoder genera un cambio en las combinaciones lógicas de la señal y según el cambio que se genere se puede saber si el giro del motor es en sentido horario o en sentido antihorario, como se muestra en la figura 55.

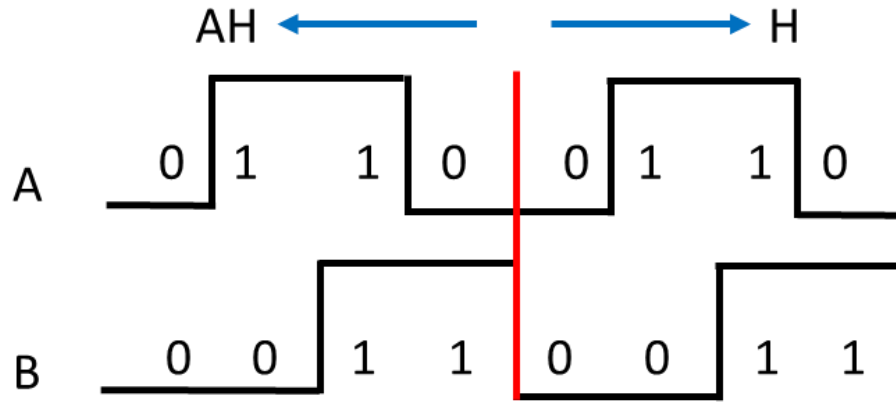


Figura 55. Combinaciones de las señales de cuadratura.

La información correspondiente al desplazamiento se obtiene directamente de A o B, un ciclo de la señal corresponde al mínimo avance, se puede usar como referencia el flanco de subida o de bajada [28]. Para los encoders que se utilizarán en este robot, que son de 1000 pulsos por revolución, el mínimo avance se obtiene mediante la siguiente ecuación:

$$\frac{360^\circ}{1000} = 0.36^\circ \quad (46)$$

En la figura 56 se muestra el diagrama de la máquina de estados para llevar a cabo el conteo de los cambios que generan las señales A y B del encoder para determinar el sentido de giro del motor.

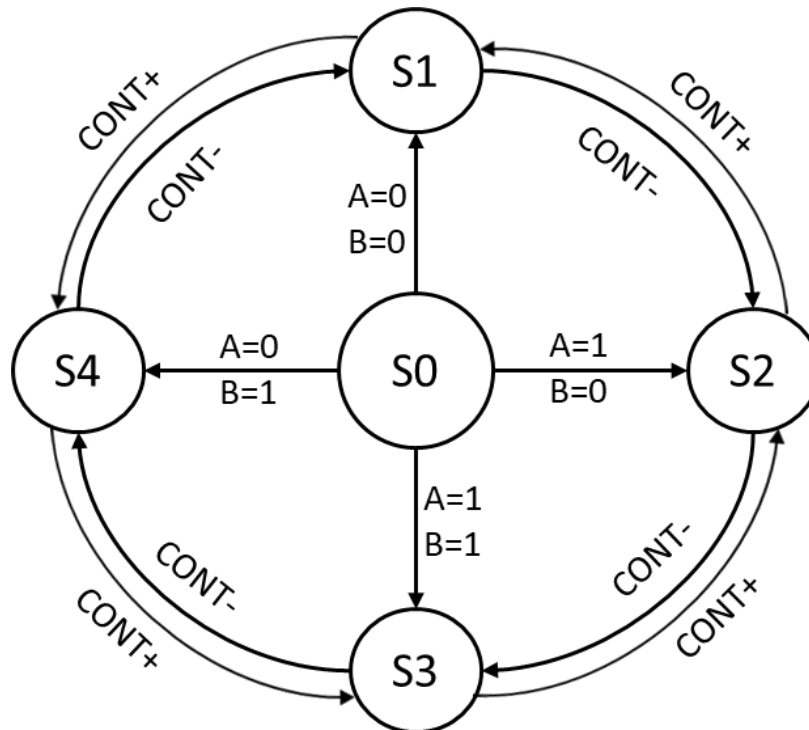


Figura 56. Diagrama de estados para la lectura de las señales A y B del encoder.

En la figura 57 se muestra el bloque donde se encuentra incluido el firmware para llevar a cabo la decodificación de las señales de los encoders para cada grado de libertad.

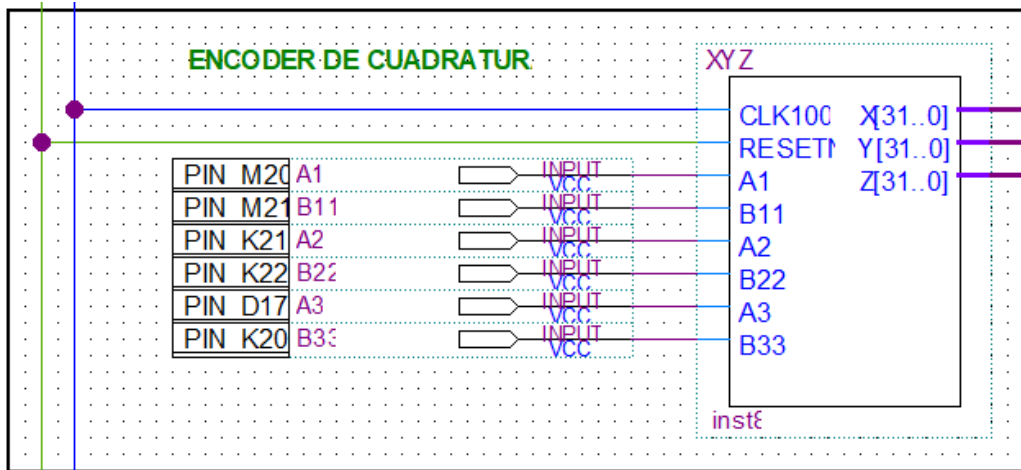


Figura 57. Bloque general para la decodificación de los encoders.

Dentro de este bloque, se genera una señal de reloj para la máquina de estados que lleva el conteo de los pulsos para determinar la posición y el sentido de giro (figura 58), esto con el fin de muestrear de manera eficiente las señales del encoder y no tener pérdida de muestras. Para lograr esto, se hace un arreglo con flip-flops tipo D (cuadro rojo) para generar un atraso en la señal original, ya sea de la señal A o de la señal B, después estas señales se ingresan a la máquina de estados (bloque azul) donde se determina la posición actual y finalmente dicho valor se convierte de entero a punto flotante (bloque verde), ya que el microprocesador solo trabaja con datos en precisión simple.

Por otra parte, en el cuadro morado se tiene una compuerta *and* que servirá para resetear el contador del encoder cuando el interruptor esté accionado.

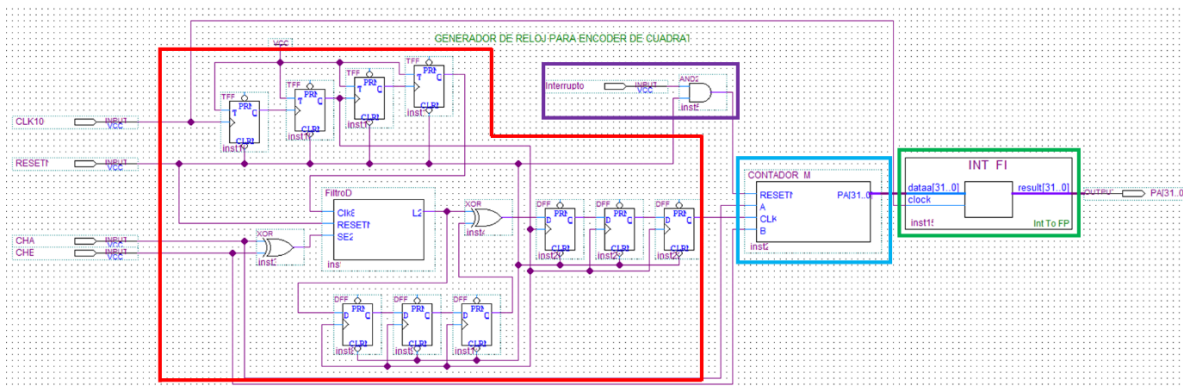


Figura 58. Firmware para generar la señal de reloj y el conteo para la máquina de estados.

4.1.4 Convertidor para ADC

En la figura 59 se muestra el bloque que contiene el firmware para interpretar y procesar las señales que se obtienen de los ADC y así determinar la temperatura actual de cada dispositivo (extrusor o base).

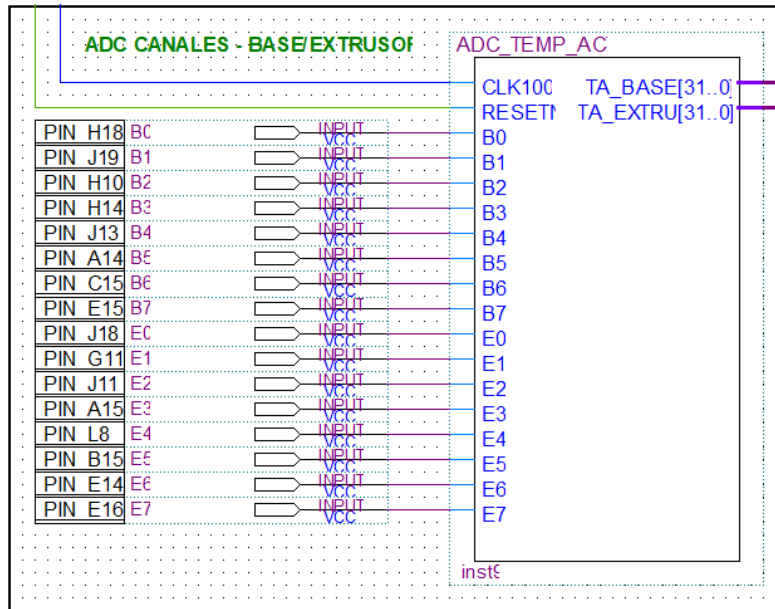


Figura 59. Bloque general para el acondicionamiento de las señales de los ADC0804.

Dentro del bloque mostrado en la figura 59 se encuentran los bloques mostrados en la figura 60. En este diagrama se encuentra el bloque llamado *ADC_TA*, en el cual se encuentra un código en AHDL que detalla el procedimiento para realizar el procesamiento de las señales que se obtienen de los ADC0804.

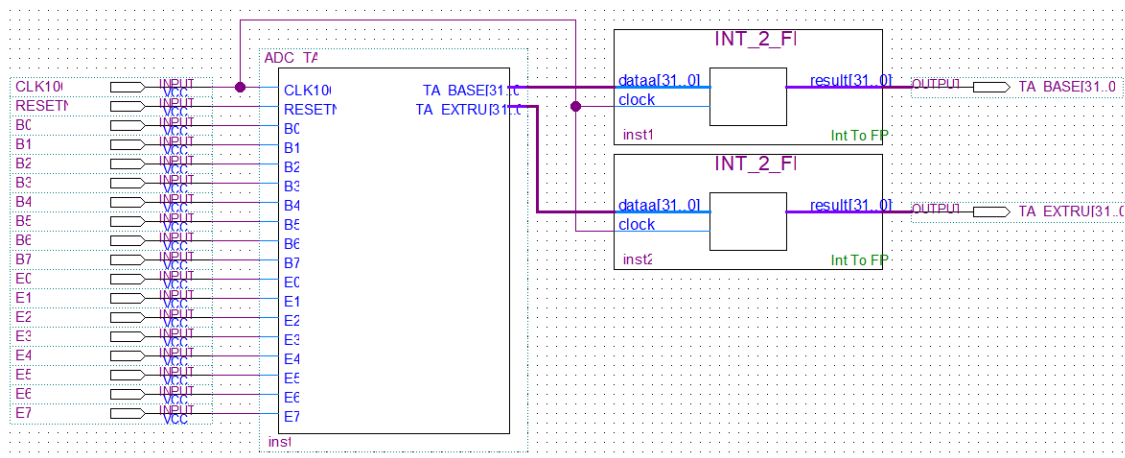


Figura 60. Conversión de las señales de entero a punto flotante.

El diagrama de estados presentado en la figura 61 detalla el procedimiento que se sigue en el código creado en AHDL, en el cual se declaran las variables de entrada que son los 8 bits de cada ADC (las B representan los 8 bits para la base y las E son para los 8 bits del extrusor) y las variables de salida, que para este caso se almacenarán los 8 bits en un arreglo de 32 bits que a su salida serán convertidos de entero a punto flotante, como se observa en la figura 60, ya que el microprocesador solo puede leer datos en formato de precisión simple, es decir, de 32 bits en punto flotante.

Después, se tiene una máquina de dos estados (S0 y S1), en el estado S0 se almacenan bit a bit los 8 bits de entrada en los 8 bits menos significativos de la variable de salida, del

bit 8 al 31 se ponen en bajo, después se habilita todo el bus, es decir, los 32 bits, y una vez teniendo todo esto se salta al estado S1.

En el estado S1 solo se espera un ciclo de reloj y se regresa al estado S0, para así asegurar que el programa almacena todos los bits y de esta manera no perder datos.

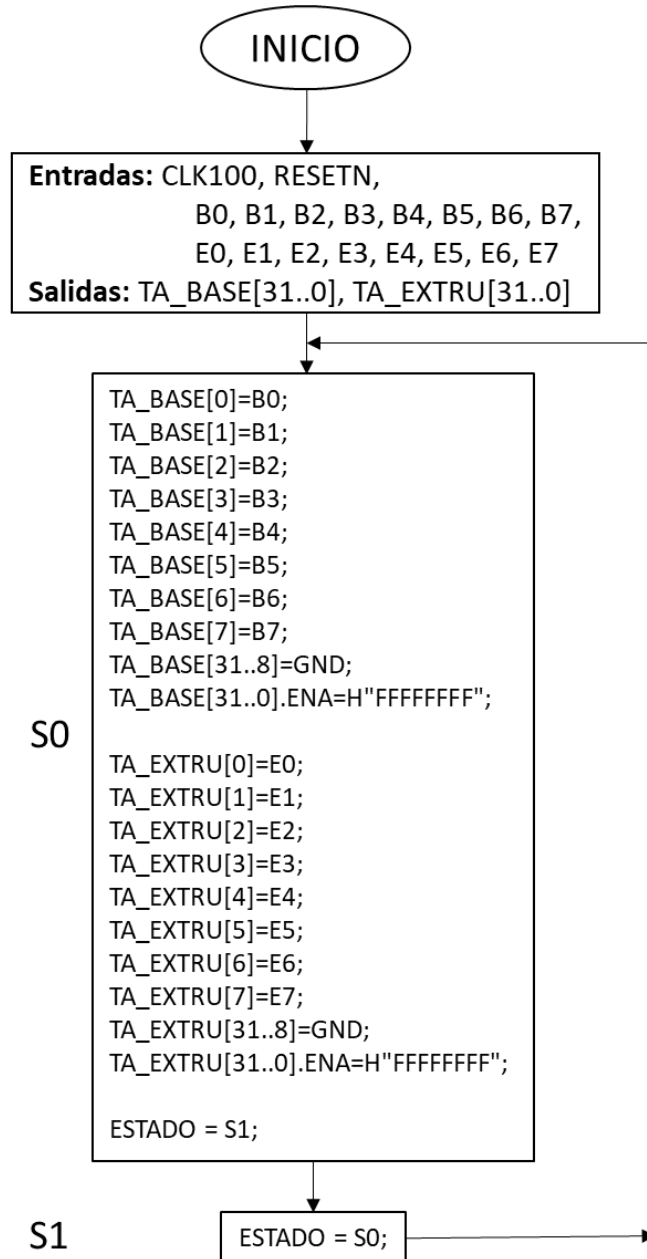


Figura 61. Diagrama de flujo para almacenar las señales de los ADC0804.

4.1.5 Interruptores de proximidad

Para que el microprocesador pueda leer los sensores de proximidad que se instalaron en cada eje, al firmware se le agregó el bloque mostrado en la figura 62.

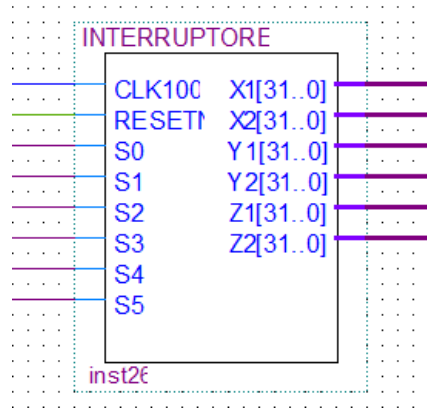


Figura 62. Bloque general para el acondicionamiento de los interruptores.

Dentro de este bloque se encuentra el firmware presentado en la figura 63, donde el bloque llamado *FIN_CARRERA* contiene un código en AHDL para realizar el procesamiento de las señales que se obtienen de dichos interruptores (ver diagrama de estados en la figura 64) y posteriormente convertirlas a punto flotante.

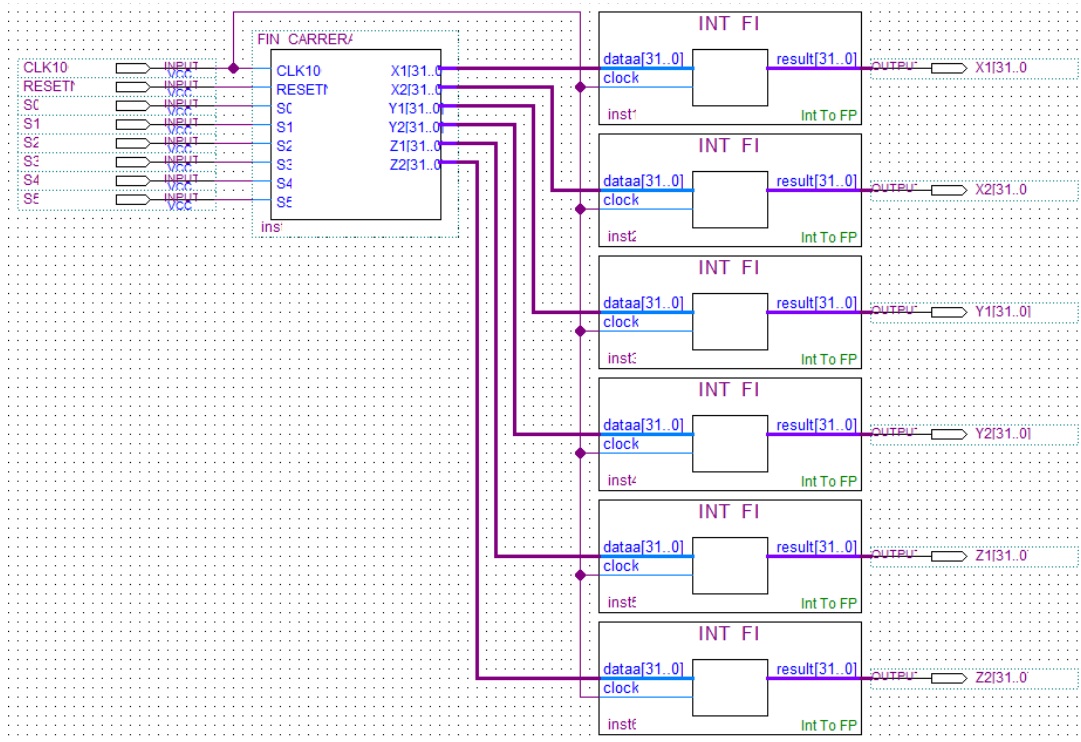


Figura 63. Firmware para la lectura y conversión de las señales de los interruptores.

En el diagrama de la figura 64 se tiene una máquina de 2 estados, S0 y S1. Para saber si un interruptor está encendido o apagado solo se necesita una señal de 1 bit, por lo tanto, en el estado S0 se almacena dicho bit en el bit menos significativo de un bus de 32 bits, después los otros 31 bits se mandan a bajo y se habilita todo el arreglo para poder pasar al estado S1, donde solo se mantiene un ciclo de reloj y regresa al estado S0, esto con el fin de almacenar bien los datos y no perder información. Finalmente, las señales de salida de 32 bits se convierten de entero a punto flotante como se muestra en la figura 63.

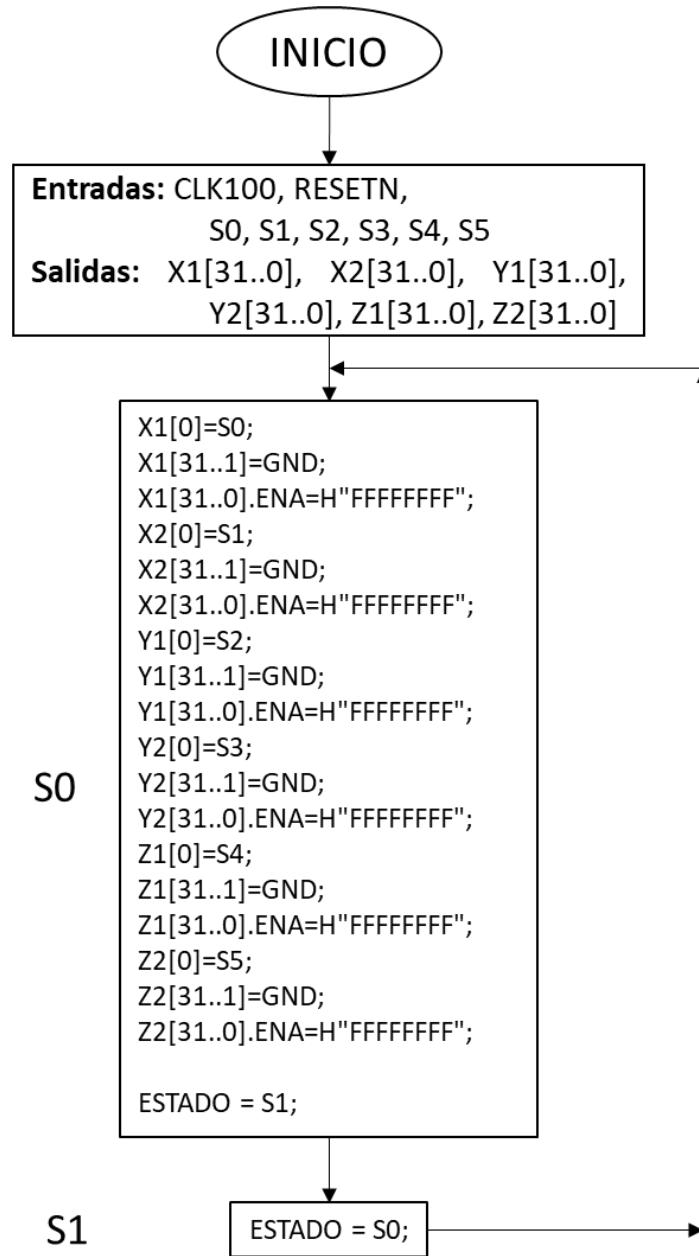


Figura 64. Diagrama de estados para la lectura de los interruptores.

Por otra parte, también se le agregó un circuito de protección mediante un arreglo con compuertas *and* y *not* a la salida de los bloques generadores de PWM de los motores, como se ve en la figura 65, para que al poner en funcionamiento los motores y estos lleguen a alguno de los extremos accionando algún interruptor se corte la energía de dicho motor y no siga avanzando con el fin de evitar algún daño en la mecánica del robot.

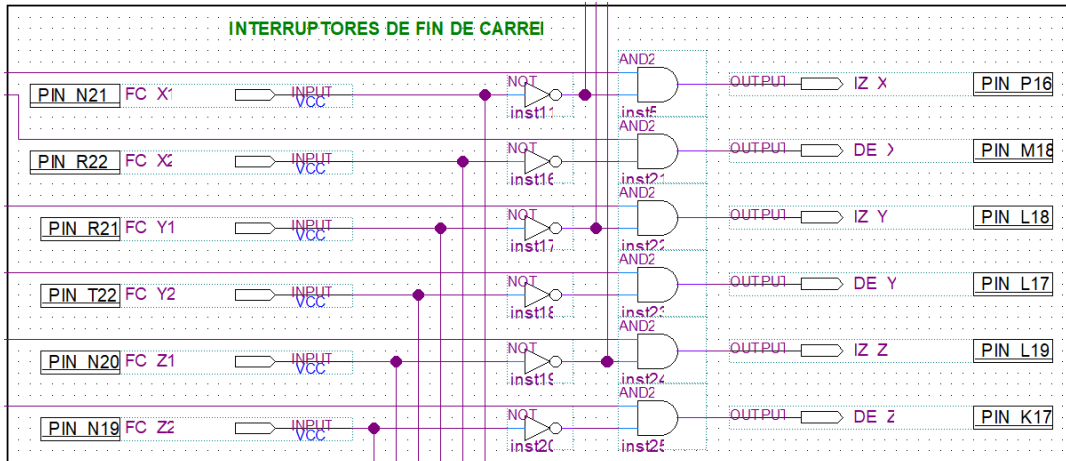


Figura 65. Circuito de protección para interruptores de fin de carrera.

4.1.6 Generador PWM

Para mandar la señal de control mediante PWM a las etapas de potencia se crearon los bloques mostrados en la figura 66. Básicamente el firmware para generar la señal de PWM y controlar la velocidad de los motores y la temperatura del extrusor de material y la cama caliente es el mismo. En el caso de los 3 gdl, se manda la señal de PWM hacia los puentes H para activar los motores y según el conteo que se esté haciendo será el sentido de giro en cada motor. Para el extrusor de material y la cama caliente, la señal de PWM se manda hacia sus respectivas etapas de potencia y de esta manera, permitir que cada dispositivo empiece a calentarse.

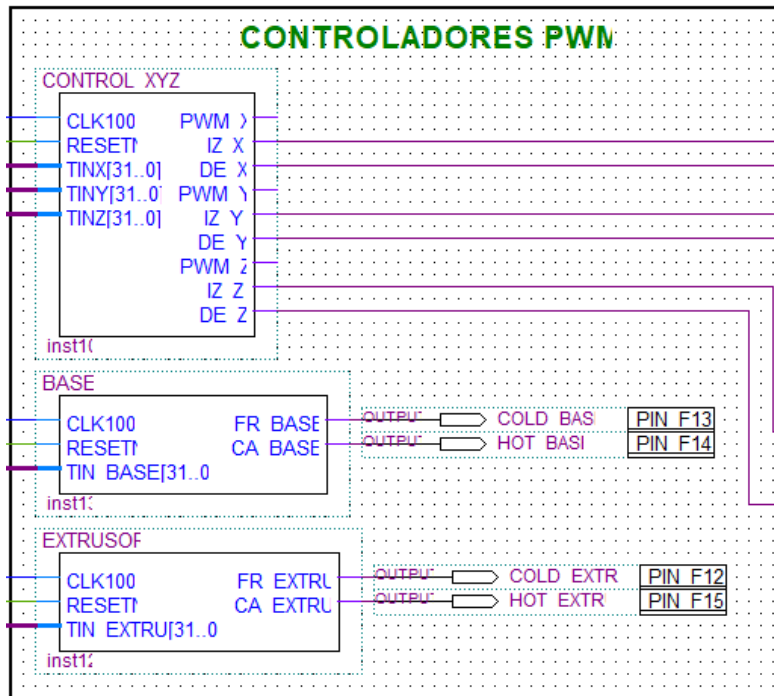


Figura 66. Bloques de control mediante PWM.

Dentro de cada bloque se encuentra el firmware necesario para mandar la señal de activación a su respectiva etapa de potencia mediante PWM, como se muestra en la figura 67.

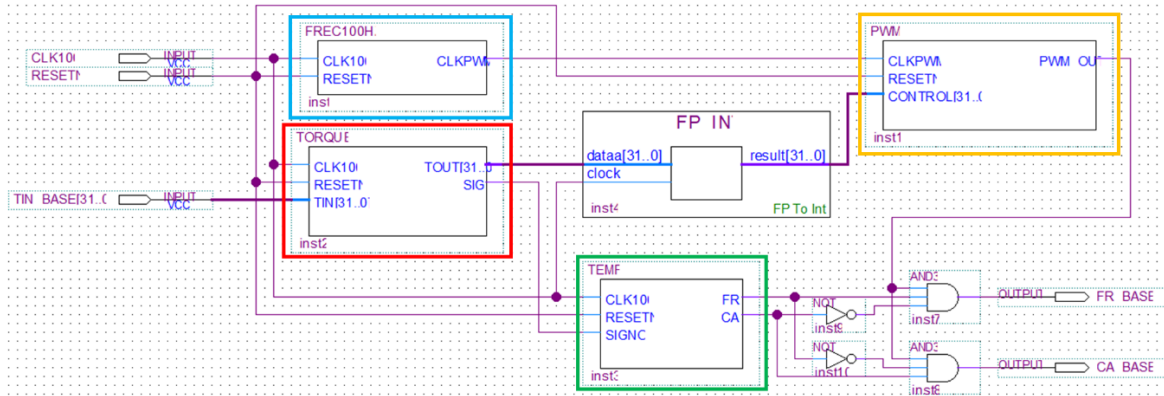


Figura 67. Firmware para el control de la etapa de potencia mediante PWM.

En el bloque encerrado en azul se crea un divisor de frecuencia, ya que la frecuencia a la que trabajará, tanto la base como el extrusor y los motores, es mucho menor a la frecuencia que trabaja la FPGA. Por lo tanto, para determinar el número de pulsos necesarios para un ciclo de reloj de la FPGA se implementa la ecuación (47).

$$\text{Núm. de pulsos} = \left(\frac{1}{f_d * f_{FPGA} * vm_{8\text{ bits}}} \right) \quad (47)$$

donde,

- f_d es la frecuencia deseada del actuador.
- f_{FPGA} es la frecuencia de la FPGA, es decir, $100\text{MHz} = 10\text{ns}$.
- $vm_{8\text{ bits}}$ es el valor máximo decimal de 8 bits, es decir, 255.

En la tabla 20 se presentan las frecuencias a las que deben trabajar los actuadores.

Tabla 20. Frecuencia de trabajo de los actuadores.

Actuador	Frecuencia
Motor X	2.5 kHz
Motor Y	2.5 kHz
Motor Z	75 Hz
Extrusor	100 Hz
Base	100 Hz

Entonces, al sustituir valores en la ecuación (47), para los motores X y Y se tiene:

$$\text{Núm. de pulsos} = \left(\frac{1}{2.5\text{ kHz} * 10\text{ ns} * 255} \right) \approx \mathbf{157\text{ pulsos de reloj}} \quad (48)$$

Para el motor Z se tiene:

$$\text{Núm. de pulsos} = \left(\frac{1}{75\text{ Hz} * 10\text{ ns} * 255} \right) \approx \mathbf{5,229\text{ pulsos de reloj}} \quad (49)$$

Y para el extrusor y la base:

$$\text{Núm. de pulsos} = \left(\frac{1}{100 \text{ Hz} * 10 \text{ ns} * 255} \right) \approx \mathbf{3922 \text{ pulsos de reloj}} \quad (50)$$

Teniendo el número de pulsos necesarios, mediante un contador se mantiene $3922/2=1961$ pulsos en alto y después los 3922 pulsos en bajo (para el caso del extrusor y la base), de esta forma se genera el reloj que va a comandar el bloque generador de PWM (bloque amarillo). En este bloque se asigna un valor de 0 a 255 el cual vendría siendo el ciclo de trabajo, para el caso de los motores X y Y será del 60%, para la base el 50% y para el extrusor el 80%.

El bloque rojo recibe el dato que manda el micro, el cual viene en formato de single precision, de este dato se almacena su bit de signo a una variable de salida (SIG) que servirá para activar el bloque encerrado en verde. Después, a ese bit de signo se le asigna un valor en bajo, el resto de los bits se mantienen igual y se habilitan los 32 bits, de esta manera, el número que representan los 32 bits será siempre positivo. A su salida se mete a un convertidor de punto flotante a entero para que el bloque generador de PWM lo pueda interpretar.

Finalmente, el bloque en verde definirá a que sentido irá la señal de control, es decir, si la temperatura a controlar es muy alta (caliente) se activa una salida con la señal de control mediante PWM para la etapa de potencia que irá en sentido positivo, de lo contrario, la señal de control será negativa, es decir, que la temperatura a controlar es baja o fría.

4.1.7 Secuencia de pasos para el control de velocidad del motor a pasos

Para que el extrusor de material tire el filamento de manera controlada y evitando saturación a la hora de realizar la impresión 3D, es necesario controlar la velocidad de funcionamiento del alimentador de material, es decir, del motor a pasos, de modo que mediante la frecuencia de la secuencia de pulsos que se manda a la etapa de potencia del alimentador se llevará a cabo el control de su velocidad. Este tipo de control será de lazo abierto ya que solo se necesita asignar una velocidad determinada para que el alimentador se mantenga en funcionamiento constante.

Por ser un motor a pasos bipolar, este cuenta con dos bobinas, por lo que se tienen cuatro cables y cada par corresponde a las terminales de una bobina. De acuerdo a la configuración de la bobina, la corriente puede fluir en dos direcciones, necesitando un control bidireccional o bipolar. En general, con respecto al sentido de giro de los motores a pasos bipolares, el sentido de giro depende de la dirección del flujo de la corriente por las bobinas ya que esta induce en el embobinado un campo magnético que genera un polo magnético norte y sur, de ahí que el rotor se mueva para que uno de los polos del rotor sea opuesto al de la bobina (localizado en el estator), como se muestra en la figura 68.

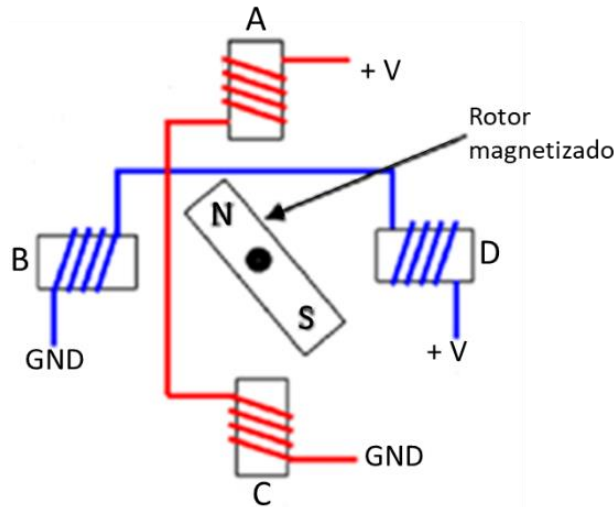


Figura 68. Modelo conceptual del motor a pasos bipolar.

En base a la configuración del motor a pasos que se implementará, se tiene que A, B, C y D son cada uno de los cuatro cables que conforman las 2 bobinas, donde una bobina está conformada por las terminales AC y la otra bobina por BD.

Para hacer que el motor a pasos gire, se necesita una secuencia de pasos para la activación de sus bobinas internas y hacer que su rotor gire hacia un sentido u otro, con este tipo de motor siempre están las dos bobinas activas y lo que se hace es ir invirtiendo el voltaje de alimentación de las bobinas mediante una secuencia. Para este caso, no es necesario que el motor gire hacia ambos lados, basta con que se mantenga hacia un solo sentido de giro, en este caso será en sentido horario, es por eso que para controlar su giro en este sentido se asignó la secuencia mostrada en la tabla 21.

Tabla 21. Secuencia para controlar el giro del motor a pasos bipolar.

Terminal	Secuencia			
	1	2	3	4
A (rojo)	VCC	GND	GND	VCC
B (amarillo)	VCC	VCC	GND	GND
C (verde)	GND	VCC	VCC	GND
D (azul)	GND	GND	VCC	VCC

Mediante el diagrama de flujo mostrado en la figura 69 se explica el proceso realizado para llevar a cabo el control del giro del motor mediante la secuencia de pasos propuesta. Como se puede observar, se utilizó un contador para llevar un orden en la secuencia.

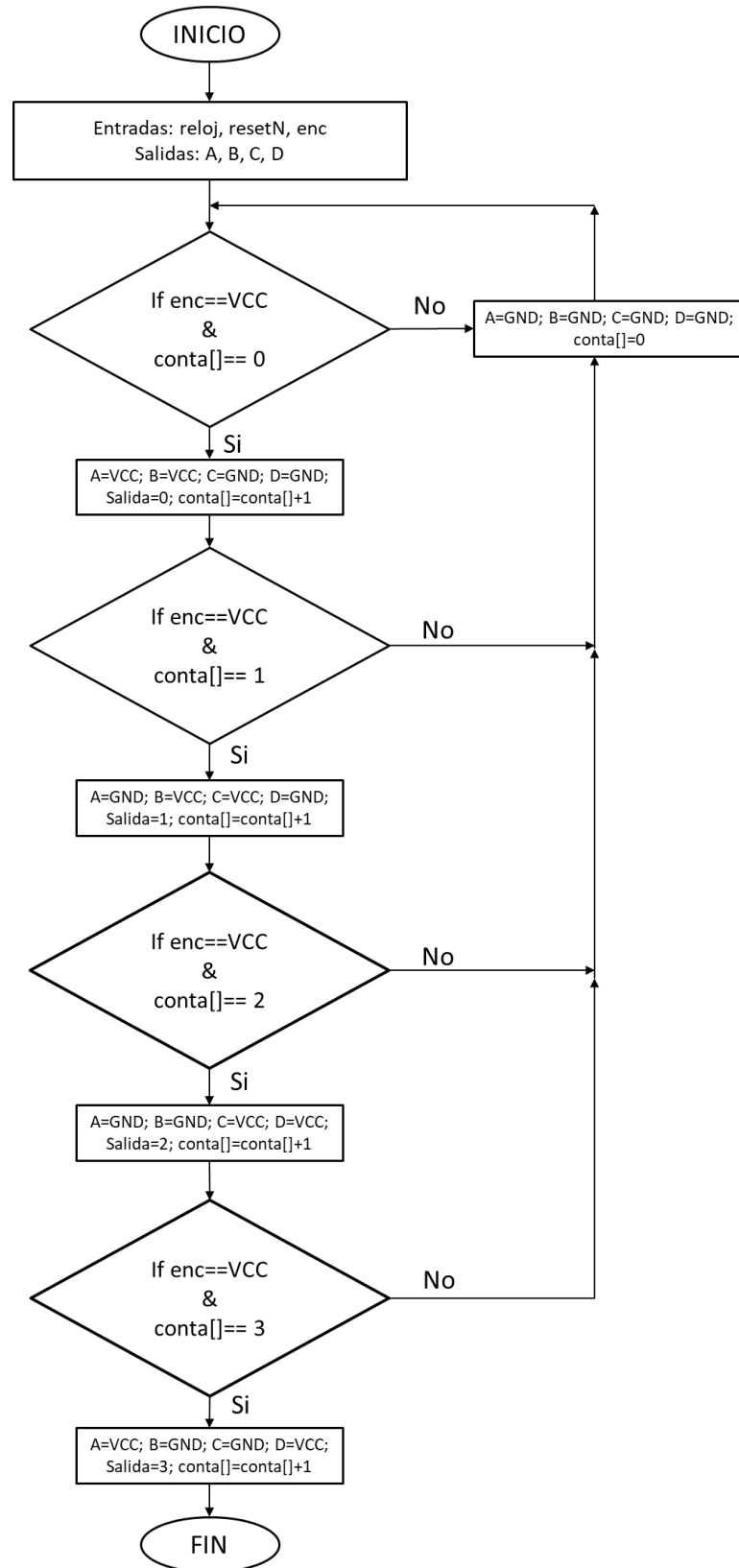


Figura 69. Diagrama de flujo de la secuencia para controlar el giro del alimentador de material.

La figura 70 muestra el bloque que contiene el firmware para llevar a cabo tanto el encendido como el control de velocidad del alimentador.

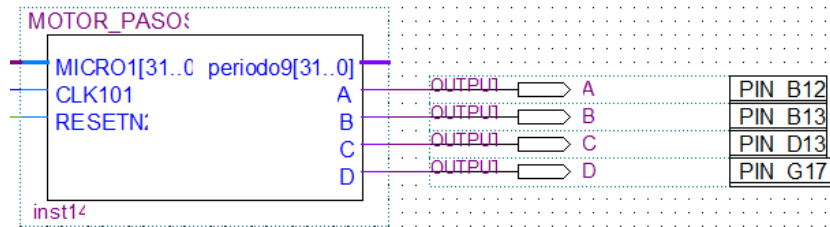


Figura 70. Bloque para el control del motor a pasos.

Dentro de este bloque, se encuentran otros bloques con instrucciones determinadas que de acuerdo a su conexionado permitirán llevar a cabo el proceso deseado, ver figura 71.

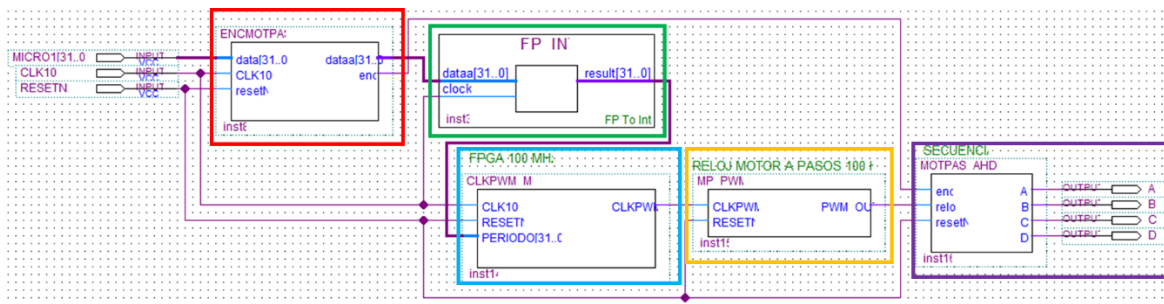


Figura 71. Firmware para el control del motor a pasos.

El bloque encerrado en rojo hace referencia al dato que llega del micro para realizar el control del alimentador. En la interfaz de usuario se asigna una constante que representa la frecuencia a la que se desea que gire el motor. Por tratarse de un motor a pasos, su velocidad se define por el cambio de las bobinas, es decir, la velocidad a la que cambien estas bobinas será la velocidad a la que gire el motor, por lo tanto, se necesita una frecuencia que indique a que razón (por minuto o por segundo) va a cambiar, es por eso que se asigna un bloque para almacenar esta constante.

Ya que la frecuencia deseada es mucho menor a la que maneja la FPGA, se implementa un divisor de frecuencia. Por lo tanto, para determinar el número de pulsos necesarios para un ciclo de reloj de la FPGA, se utiliza la ecuación (47). Como el motor a pasos debe moverse a una velocidad muy baja, entonces se le asigna una frecuencia de 20 Hz, por lo tanto, sustituyendo valores en dicha ecuación, se obtiene:

$$\text{Núm. de pulsos} = \left(\frac{1}{20\text{Hz} * 10\text{ns} * 255} \right) \approx 19,608 \text{ pulsos de reloj} \quad (51)$$

Entonces, lo que hace el bloque encerrado en rojo es determinar el bit de signo (bit 31) de la constante que llega a través del micro. Con este bit de signo se hace el encendido para indicarle al bloque que controla directamente al motor (bloque encerrado en morado) que va a estar encendido cuando se mande un número positivo. Por lo tanto, como se tiene el dato en formato single precision, se le extrae el bit de signo y se compara su valor, si es positivo (0) se habilita el *enable* en la salida *enc* y posteriormente el mismo dato que entró se pasa a otro bloque para convertirlo de formato single a formato entero (bloque encerrado en verde),

ya que Quartus no realiza operaciones en formato single precision, entonces una vez que se tiene el dato en formato entero se manda al bloque que crea el contador (bloque encerrado en azul).

Cuando llega a este bloque el dato que contiene la mitad del número de pulsos calculado previamente, empieza a contar y se mantiene en alto 9,804 pulsos, después los pulsos se multiplican por dos para mantener el valor en bajo, de esta forma se genera el reloj que va a comandar el siguiente bloque que es el generador de PWM (bloque encerrado en amarillo). A este bloque se le asigna un valor de 0 a 255, llamado ciclo de trabajo (duty cycle), en este caso se tiene un ciclo de trabajo del 50%.

Finalmente, lo que sale del bloque generador de PWM es la señal de reloj que va a estar a la frecuencia que se le asigna al principio, entonces esta frecuencia entra al bloque encerrado en morado, que es donde se asigna la secuencia de pasos para controlar el giro del motor, de modo que el cambio de las bobinas se hará a razón de esta frecuencia. Las salidas (A, B, C, D) de este bloque se conectan a la etapa de potencia del motor a pasos (figura 49).

En la figura 72 se puede observar el desfase que hay entre los canales y como siguen la secuencia de pasos que fue asignada en la tabla 21 para que el motor a pasos se pueda mover en sentido horario.

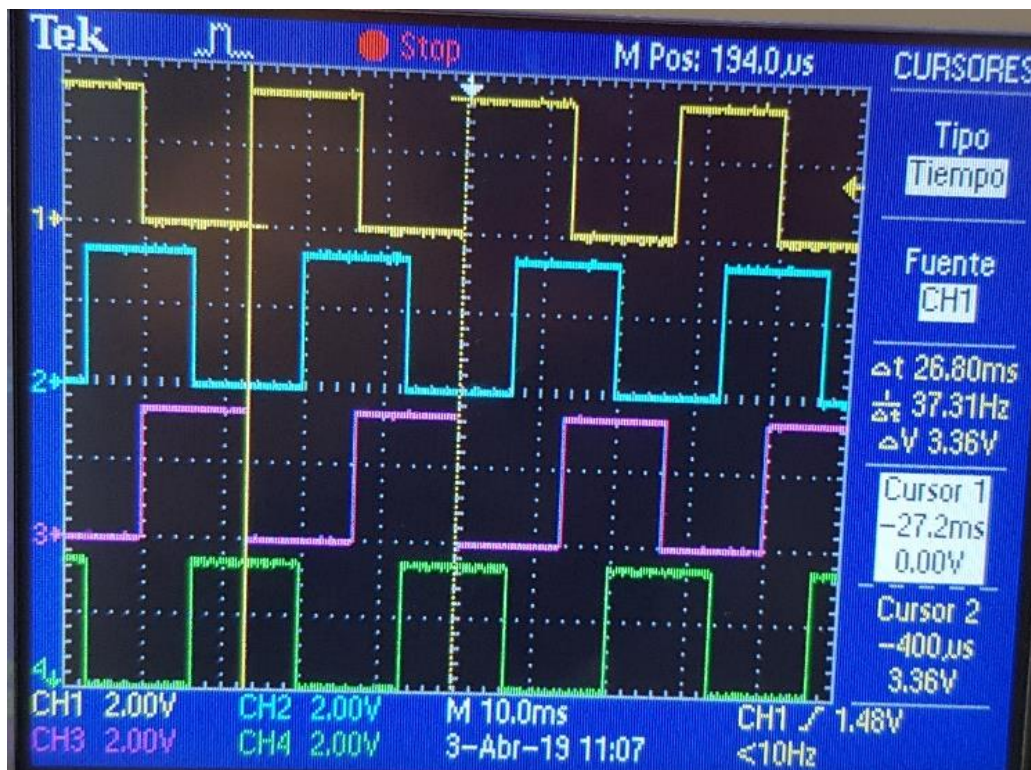


Figura 72. Señales de la secuencia de pasos del alimentador de material.

4.2 Software

4.2.1 Lazo de control

4.2.1.1 Control de posición

Para realizar el control de posición de los 3 ejes del robot, se implementó un control con función tangente hiperbólica para mantener la acción de control acotada, ya que además de tener sustento científico, una de las cualidades de este esquema de control es que permite operar al actuador (base o extrusor) en la parte lineal lejos de los límites de saturación.

En la figura 73a se muestra el diagrama a bloques del control de posición implementado para el eje Z y en la figura 73b para los ejes X y Y.

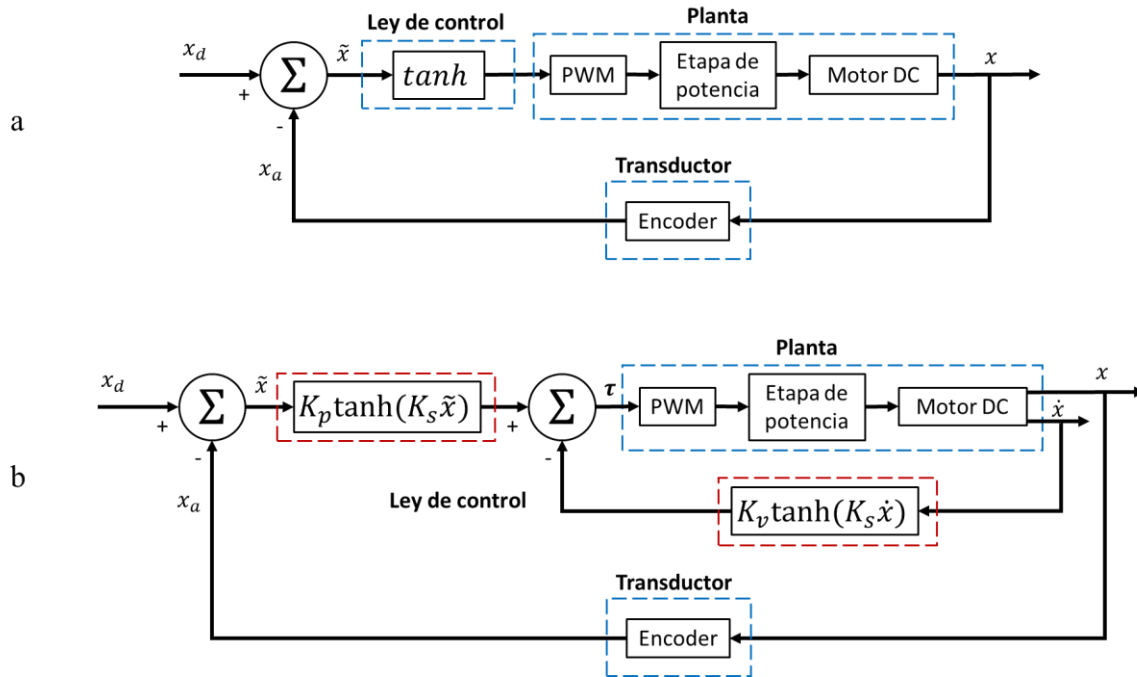


Figura 73. Lazo cerrado para el control de posición: a) Eje Z, b) Ejes X y Y.

El esquema de control para los ejes X y Y está dado por [1] y [29] mediante la siguiente ecuación:

$$\tau = K_p \tanh(K_s \tilde{x}) - K_v \tanh(K_s \dot{x}) \quad (52)$$

donde,

- K_p es la ganancia proporcional y es definida positiva.
- K_v es la ganancia derivativa y es definida positiva.
- K_s es la pendiente del control tangente hiperbólica $\tanh(K_s \tilde{x})$ y de la acción derivativa $\tanh(K_s \dot{x})$ y es definida positiva.
- \tilde{x} es el error de posición, definido como la diferencia entre la posición deseada y la posición actual, es decir, $\tilde{x} = x_d - x_a$.
- \dot{x} es la velocidad de movimiento que se emplea para inyección de amortiguamiento.

- $K_p \tanh(K_s \tilde{x}) = 0 \Leftrightarrow \tilde{x} = 0$.
- $K_v \tanh(K_s \dot{x}) = 0 \Leftrightarrow \dot{x} = 0$.

La velocidad se calcula mediante la siguiente ecuación:

$$\dot{x}(k) = \frac{x(k) - x(k-1)}{h} \quad (53)$$

donde,

- $x(k)$ es la posición actual.
- $x(k-1)$ es la posición anterior.
- h es la frecuencia de muestreo, para este caso es igual a $2.5ms$.

Para el caso del eje Z el esquema de control queda de la siguiente manera:

$$\tau = K_p \tanh(K_s \tilde{x}) \quad (54)$$

En los siguientes diagramas de flujo se presentan las estrategias de control que se implementaron de forma más detallada para cerrar el lazo de control en los 3 ejes con control de posición, en los cuales se utilizan los algoritmos de control (52) para los ejes X y Y y (54) para el eje Z. De igual manera se toman en cuenta las condiciones para determinar que tanta energía (PWM) se le proporcionará a cada motor.

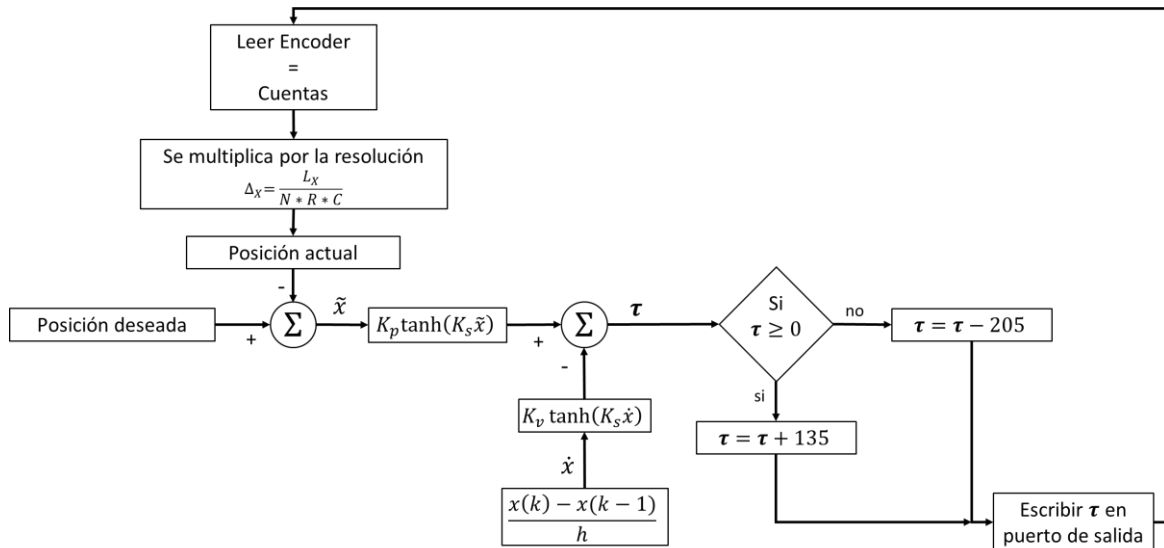


Figura 74. Diagrama de flujo para el control de posición del eje X.

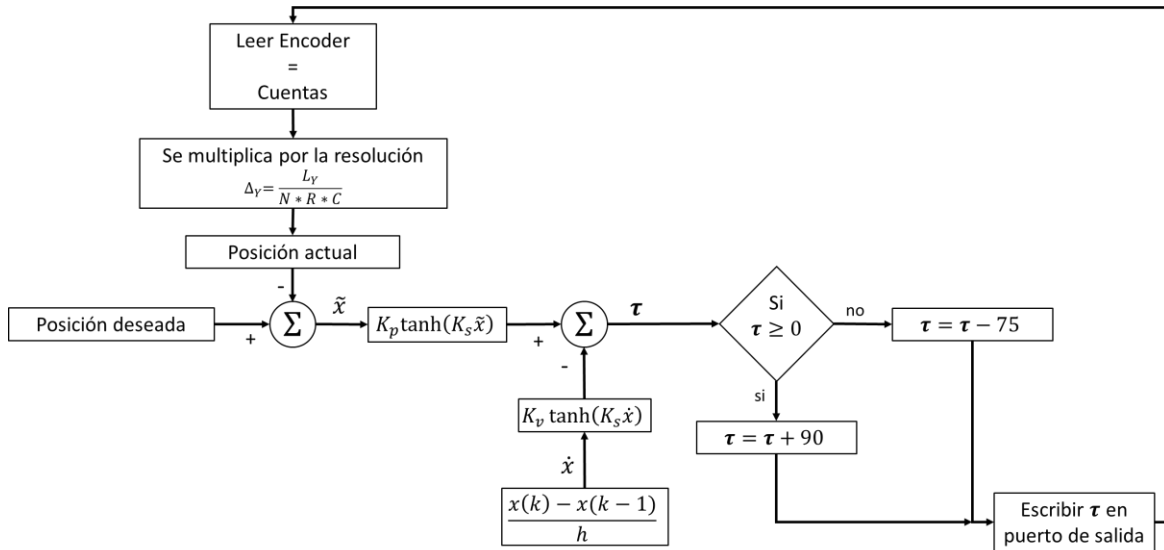


Figura 75. Diagrama de flujo para el control de posición del eje Y.

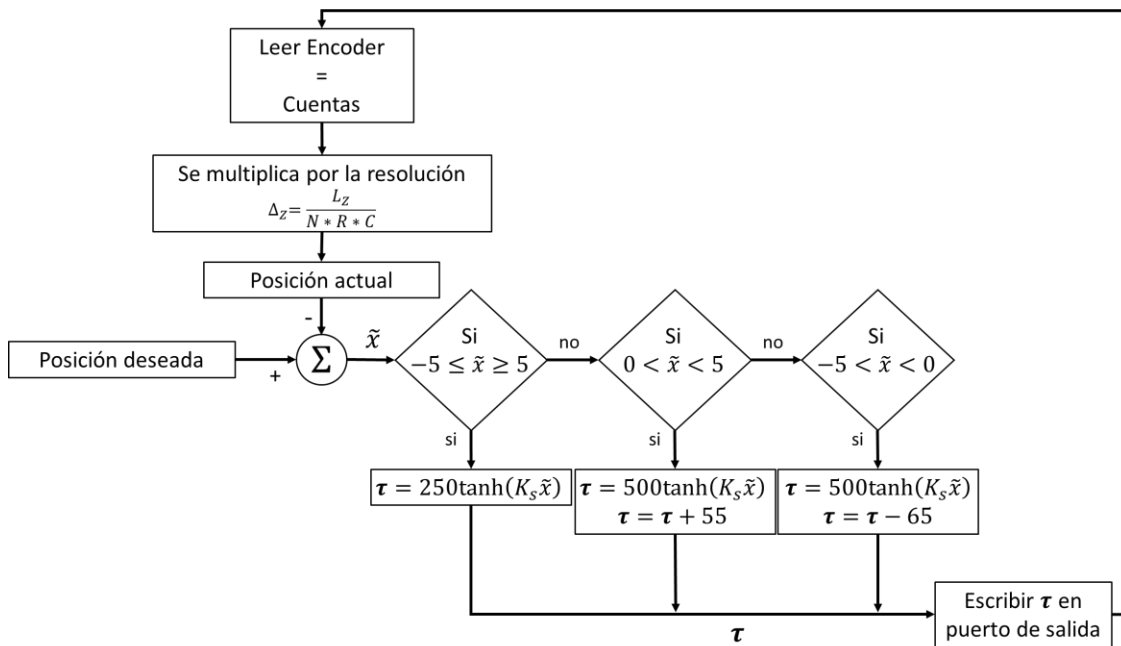


Figura 76. Diagrama de flujo para el control de posición del eje Z.

4.2.1.2 Control de velocidad

Para controlar el movimiento de los ejes X y Y, es decir, para que logren realizar una impresión más rápida o lenta, se implementó un control de velocidad autosintonizable, cuyo algoritmo de control está dado de la siguiente manera:

$$\tau_i = \tau_{i-1} + K_p \tanh(K_s \dot{x}) \quad (55)$$

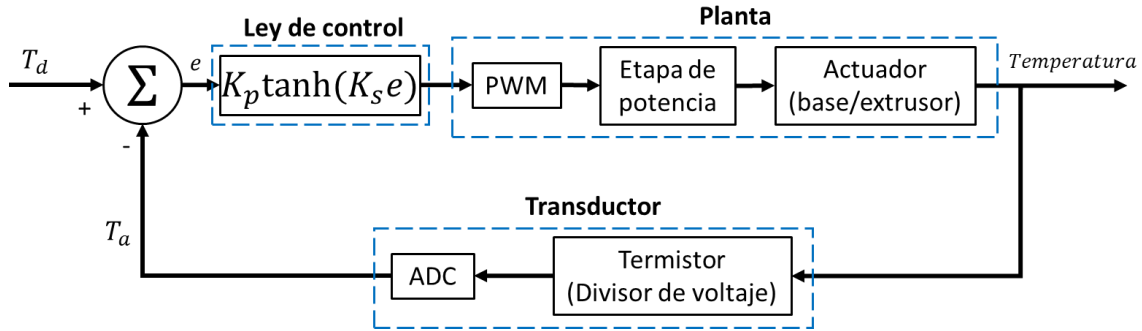


Figura 78. Lazo cerrado para el control de temperatura.

Donde,

- T_d es la temperatura deseada ($^{\circ}\text{C}$).
- T_a es la temperatura actual ($^{\circ}\text{C}$).
- e es el error de temperatura, es decir, $e = T_d - T_a$.

El algoritmo de control con función tangente hiperbólica que se propone para el control de temperatura es el siguiente:

$$\tau = K_p \tanh(K_s e) \quad (56)$$

donde,

- K_p es la ganancia proporcional y es definida positiva.
- K_s es la matriz diagonal definida positiva que representa la pendiente de la función $\tanh(K_s e)$.
- $\tanh(e) = 0$ cuando $e = 0$.

En la figura 79 se presenta el diagrama de flujo sobre el procedimiento que se sigue de forma detallada para cerrar el lazo de control de temperatura, en el cual se implementa el algoritmo de control (56), así como las condiciones que se toman en cuenta para determinar la energía (PWM) que se le proporcionará al extrusor de material o a la base.

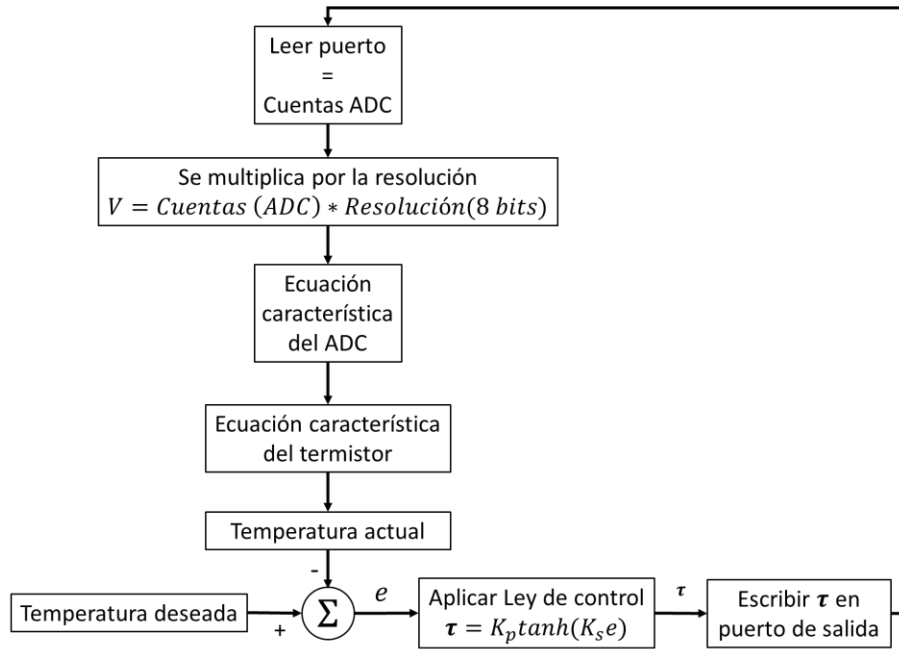


Figura 79. Lazo cerrado para el control de temperatura del extrusor y la base.

4.2.2 Software de usuario

Se cuenta con dos softwares de usuario:

1. *Compilador de posiciones deseadas*. Necesario para formar la figura en 3D, dichas posiciones se obtienen a partir de una pieza diseñada en CAD (figura 80).

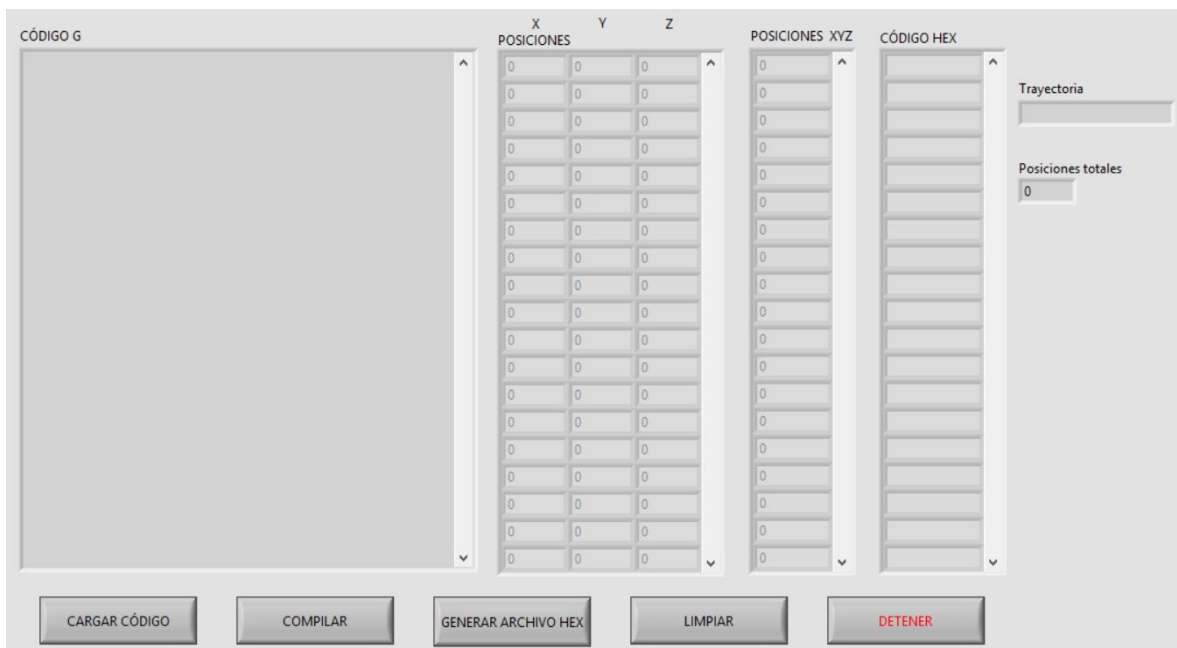


Figura 80. Interfaz gráfica del compilador de posiciones deseadas.

2. *Interfaz de usuario.* El cual permite tener comunicación entre el robot y la computadora mediante vía Wi-Fi (figura 81).

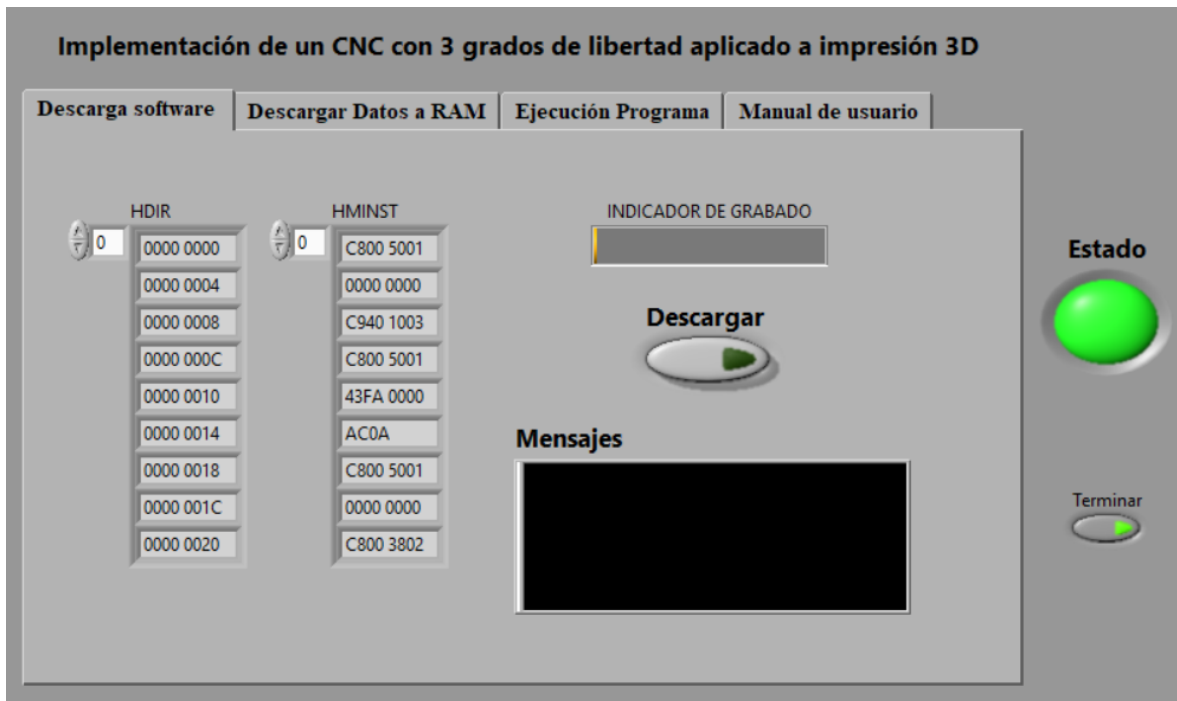


Figura 81. Interfaz de usuario para controlar el sistema mecatrónico.

4.2.2.1 Compilador de posiciones deseadas

Mediante la interfaz gráfica de LabView se diseñó el compilador mostrado en la figura 80, el cual permite obtener las posiciones deseadas que se necesitan para imprimir una pieza en 3D. El proceso necesario para que el robot cartesiano pueda leer las posiciones y realizar la impresión en 3D se describe en el diagrama de flujo de la figura 82.

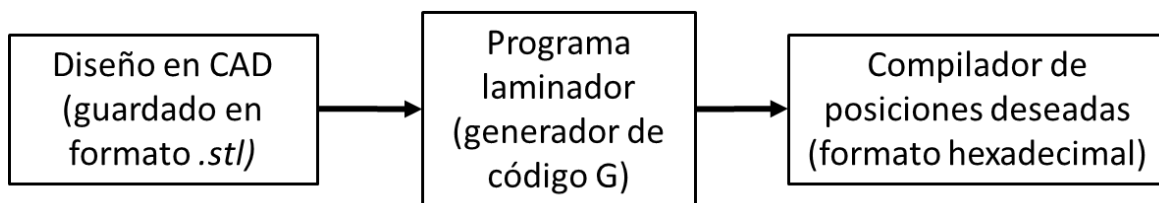


Figura 82. Proceso previo a la impresión en 3D.

El procedimiento consta de los siguientes pasos:

- 1) Se hace el diseño en un programa CAD de la figura que se desee imprimir en 3D.

El diseño asistido por computadora, mejor conocido por sus siglas en inglés CAD, es todo sistema informático que proporciona distintas herramientas de diseño para crear objetos, ya sean reales o inventados por uno mismo, en representación 2D y 3D. Este tipo de plataforma es mayormente utilizado por ingenieros, arquitectos y diseñadores.

Entre los programas más populares de CAD se encuentran:

- SolidWorks
- Auto CAD
- Autodesk Inventor
- CoCreate Solid Designer
- 3D Max

A partir de alguno de estos programas se diseña el modelo 3D de lo que se quiera imprimir en 3D, un ejemplo puede ser como la que se muestra en la figura 83.

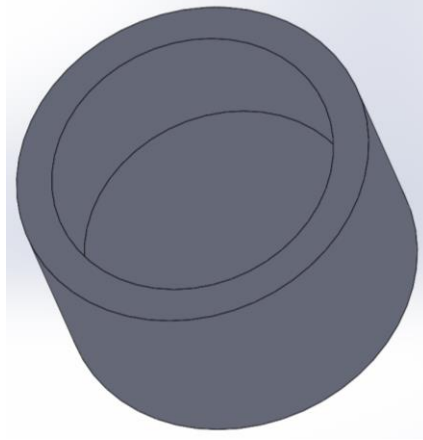


Figura 83. Ejemplo de modelo en 3D diseñado en SolidWorks.

2) Se guarda el archivo con extensión *.stl*.

Un archivo STL (Standard Triangle Language) es un formato de archivo informático creado desde aplicaciones CAD que define la geometría de objetos 3D en forma de malla, descartando información como el color, texturas o propiedades físicas que sí incluyen otros formatos CAD.

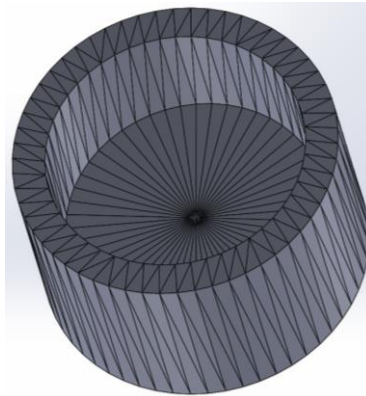


Figura 84. Modelo 3D en formato *.stl*.

3) Mediante un programa laminador se abre el archivo *stl* para generar su código G.

Algunos programas de laminado son:

- Slic3r

- Repetier-Host
- Cura
- 3DPrinterOS

En un programa laminador (figura 85), a partir del archivo *stl* abierto se genera el código G que a su vez se guarda con la extensión *.gcode*, como se muestra en la figura 86. Para este caso se utiliza el programa Slic3r, el cual cuenta con pestañas para ingresar algunas características de la impresora 3D, características del filamento, la velocidad de impresión, la densidad de relleno de la pieza, entre otras cosas.

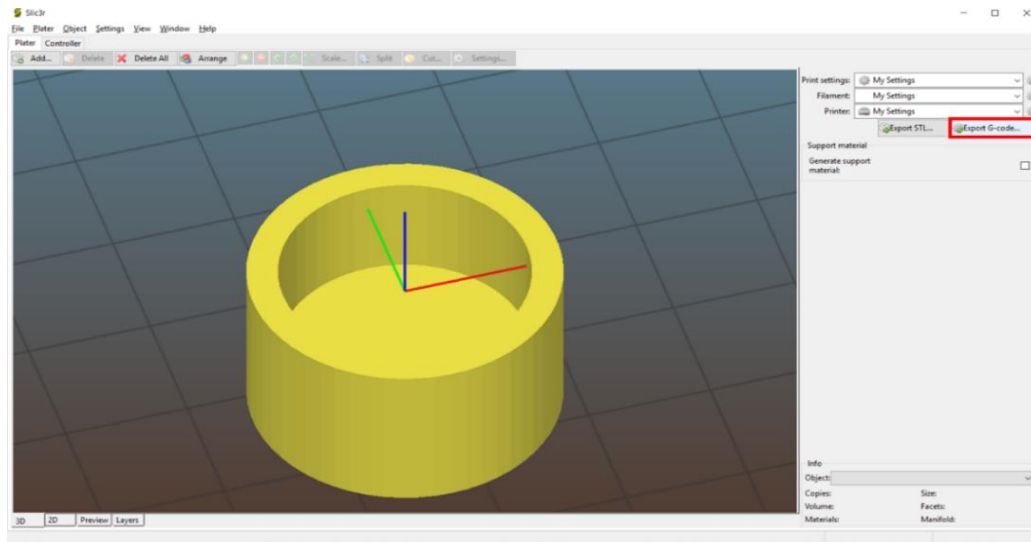


Figura 85. Modelo 3D laminado.

```
; generated by Slic3r 1.3.0-dev on 2018-03-08 at 11:27:39

; external perimeters extrusion width = 0.40mm (3.02mm^3/s)
; perimeters extrusion width = 0.48mm (7.54mm^3/s)
; infill extrusion width = 0.48mm (10.05mm^3/s)
; solid infill extrusion width = 0.48mm (2.51mm^3/s)
; top infill extrusion width = 0.48mm (1.88mm^3/s)

M107
M190 S120 ; set bed temperature and wait for it to be reached
M104 S225 ; set temperature
G28 ; home all axes
G1 Z5 F5000 ; lift nozzle

; Filament gcode

M109 S225 ; set temperature and wait for it to be reached
G21 ; set units to millimeters
G90 ; use absolute coordinates
M82 ; use absolute distances for extrusion
G92 E0
G1 Z0.400 F7800.000
G1 E-2.00000 F2400.00000
G92 E0
G1 X93.529 Y93.308 F7800.000
G1 E2.00000 F2400.00000
G1 F1800
G1 X94.057 Y92.808 E2.01058
G1 X95.089 Y91.931 E2.03027
G1 X95.667 Y91.492 E2.04084
G1 X96.788 Y90.732 E2.06054
G1 X97.411 Y90.357 E2.07111
```

Figura 86. Código G del modelo 3D laminado.

- 4) Desde el compilador de posiciones deseadas (figura 87) se carga el código G y se obtienen las coordenadas para cada eje.

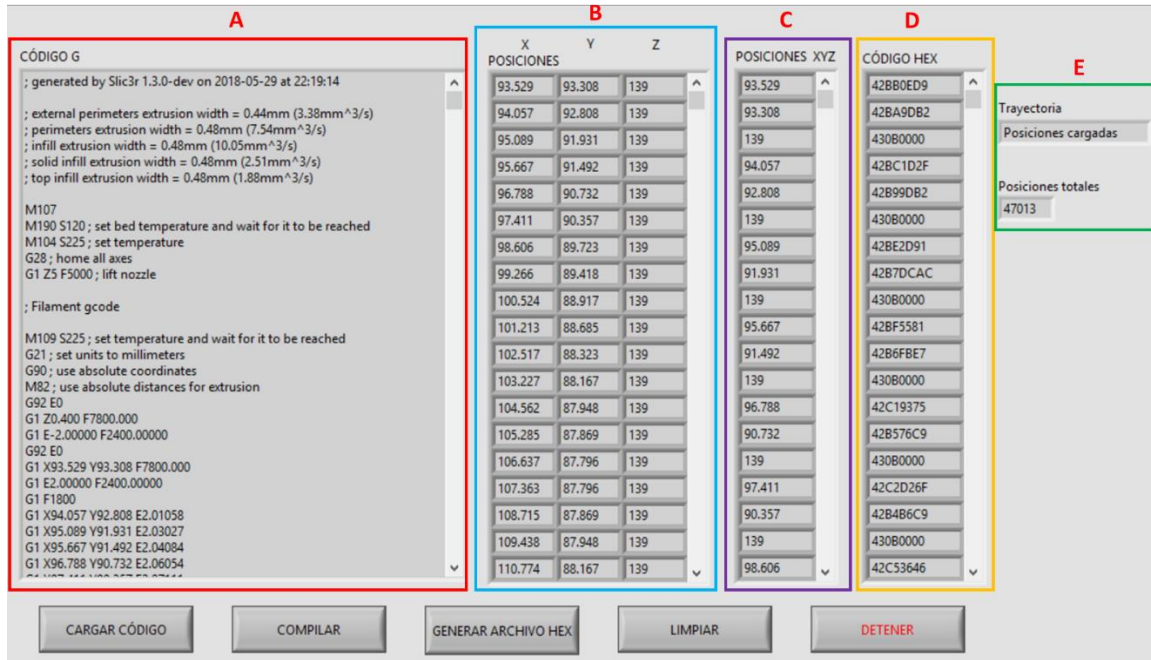


Figura 87. Compilador para obtener posiciones deseadas.

Al ejecutar el compilador, se oprime el botón *CARGAR CÓDIGO*, el cual abrirá una ventana para seleccionar el archivo a compilar (código G), ver figura 88; una vez cargado el archivo en la parte encerrada en A aparecerá su contenido.

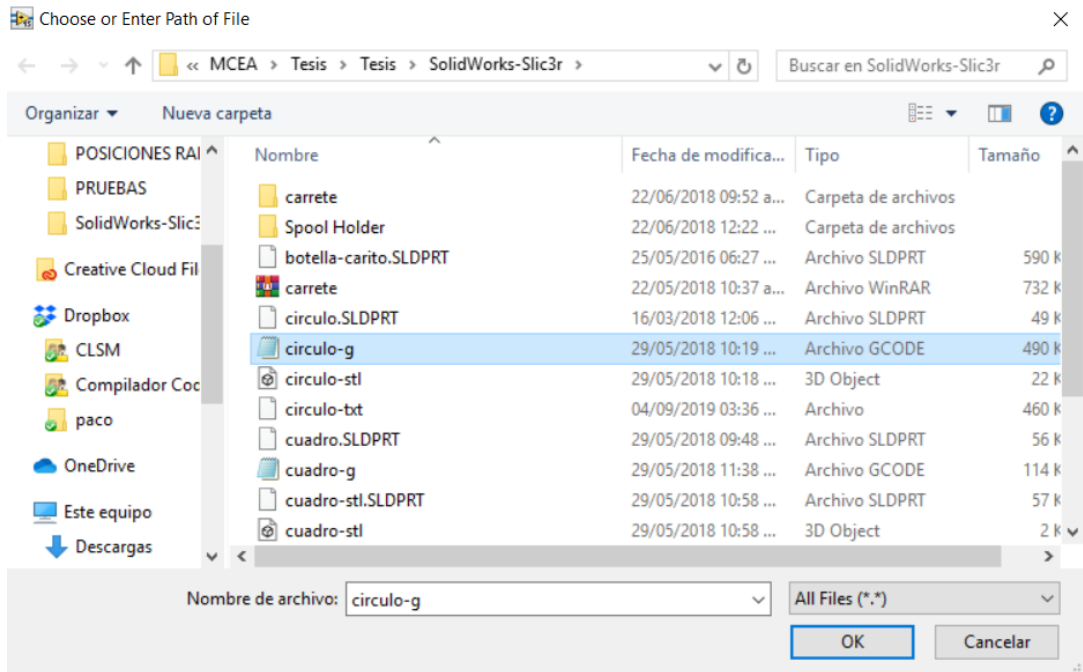


Figura 88. Ventana para seleccionar el archivo a compilar.

Después se oprime el botón COMPILAR para comenzar con la depuración del código G y obtener las coordenadas de cada eje las cuales irá almacenando en un arreglo de $m \times 3$, como se muestra en “B”. En “C” se acomodan dichas posiciones en un solo vector de manera intercalada para después ser convertidas a hexadecimal en “D” para que puedan ser interpretadas por el microprocesador. En “E” se tiene un indicador que muestra el estado del proceso y otro que da el número de posiciones totales que se obtienen del vector “D”.

Una vez que finalizó la compilación se oprime el botón *GENERAR ARCHIVO HEX* para guardar las posiciones en formato hexadecimal en un archivo con extensión *.txt* (figura 89). El botón *LIMPIAR* se ocupa cuando se quiere volver a hacer una compilación, esto con el fin de limpiar todos los bloques y obtener otra compilación exitosa sin tener que detener el programa.

```
42BB0ED9
42BA9DB2
430B8000
42BC1D2F
42B99DB2
430B8000
42BE2D91
42B7DCAC
430B8000
42BF5581
42B6FBE7
430B8000
42C19375
42B576C9
430B8000
42C2D26F
42B4B6C9
```

Figura 89. Posiciones deseadas en formato hexadecimal.

4.2.2.2 Interfaz de usuario

Para llevar a cabo el control de todo el robot, es decir, el control de posición y velocidad de los 3 gdl, los controles de temperatura (extrusor y cama caliente) y el control de velocidad para el motor a pasos, se utiliza una interfaz gráfica (figura 81) desarrollada en LabView. La cual, a través de protocolo de comunicación Wi-Fi, permite la comunicación entre la computadora y el microprocesador embebido en la tarjeta de adquisición (FPGA), además trabaja a una velocidad de muestreo de $2.5ms$.

Como se muestra en la figura 90, la interfaz de usuario cuenta con 3 pestañas para llevar a cabo distintas funciones. En la pestaña “1” se tiene la pantalla principal de la interfaz, de la cual en “A” se descarga a la memoria ROM un archivo que contiene el código con las instrucciones necesarias mediante el cual se realizará el control de todo el sistema, es decir, donde se cierran todos los lazos de control. Este código está basado en instrucciones en ensamblador para que el microprocesador las pueda interpretar y realizar las correspondientes operaciones aritméticas.

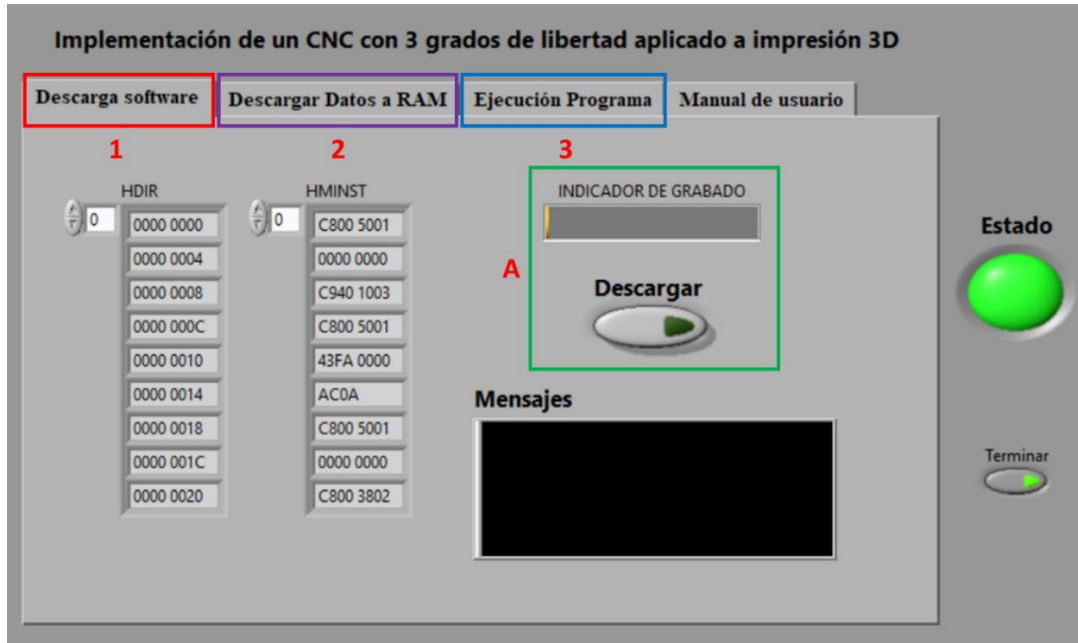


Figura 90. Ventana para descargar el software de control.

Posteriormente, en la figura 91 se muestra la pestaña 2 de la figura 90. En esta ventana se descarga a la memoria RAM el archivo que contiene las posiciones deseadas en formato hexadecimal que se obtuvieron mediante el compilador de posiciones deseadas. Antes de descargar el archivo que contiene dichas posiciones en "B", se le puede asignar un offset en "A" para que al descargar los datos se empiecen a almacenar a partir de determinada dirección de la RAM.



Figura 91. Ventana para descargar posiciones deseadas a la RAM.

Pasando a la tercera pestaña de la figura 90 se tiene la ventana donde se lleva a cabo la ejecución del programa, como se muestra en la figura 92, la cual cuenta con 3 subpestañas.

En la subpestaña “3.1” de la figura 92 se asignan los parámetros de inicio en “B”, es decir, posiciones o velocidades y temperaturas deseadas con sus respectivas ganancias y la frecuencia a la que se moverá el motor a pasos para después en “A” ejecutar o detener código de control que se descargó en el microprocesador.



Figura 92. Ventana para inicializar parámetros.

En la subpestaña “3.2” de la figura 93, después de ejecutar el código de control, en “A” se tiene la opción de enviar comandos al microprocesador para poder llevar a cabo distintos procesos, ya sea leer puertos, mandar posiciones o temperaturas deseadas, variar ganancias, etc. Por ejemplo, si en el código de control al comando 20 se le asignó escribir posición deseada para el eje X, entonces en “A” se escribe el comando 20, en “B” se escribe el valor de la posición deseada y en “C” se envían al microprocesador para que ejecuten dicha acción.



Figura 93. Ventana para enviar comandos al microprocesador.

Finalmente, en la subpestaña “3.3” de la figura 94 se realiza la lectura de datos, es decir, si en el código de control se manda a leer la posición actual del eje X, en esta ventana se va almacenando la posición actual para poder guardarlos en un archivo, donde en “A” se le da nombre al archivo para posteriormente en “B” grabar dichos datos.



Figura 94. Ventana para la lectura de datos.

4.3 Conclusiones

El desarrollo del firmware embebido en la FPGA de la tarjeta DE0-CV que se implementó para llevar a cabo el control del sistema mecatrónico es de gran importancia ya que dicha tarjeta no cuenta con firmware propio para realizar algún tipo de control.

En cuanto a los algoritmos de control, se optó por implementar tangente hiperbólica ya que ofrece la ventaja, contra el control PD por ejemplo, de que a partir de la correcta sintonía de ganancias permite que el actuador no se sature y de esta manera pueda trabajar en la parte lineal.

Por otra parte, gracias al desarrollo del compilador de posiciones, es posible crear secuencias de posiciones sin la necesidad del diseño de una figura 3D que se tenga que pasar por un laminador para obtener el código G que proporciona dichas posiciones, ya que uno mismo en un bloc de notas puede crear su propia secuencia de movimientos y convertirlas mediante el compilador de posiciones para poder hacer pruebas en el robot.

Así también, a través de la interfaz de usuario es posible mandar a escribir datos a la memoria RAM, asignar parámetros de inicio, enviar comandos para variar posiciones deseadas, temperaturas deseadas o ganancias en distintos actuadores cuando se hacen pruebas e incluso leer y guardar datos en un archivo para poder graficarlos.

Capítulo 5: Resultados experimentales

5.1 Estructura mecánica de la impresora 3D

Al retomar el robot cartesiano tipo CNC de 3 grados de libertad (figura 95a) para implementarlo como impresora 3D, se tuvieron que realizar varias modificaciones y cambios en la electrónica, como se mencionó en los capítulos anteriores, donde se explica el cambio de encoders, el desarrollo de las etapas de potencia, el diseño del porta carrete, el cambio de extrusor y de cadenas para el recableado, quedando como se muestra en la figura 95b.

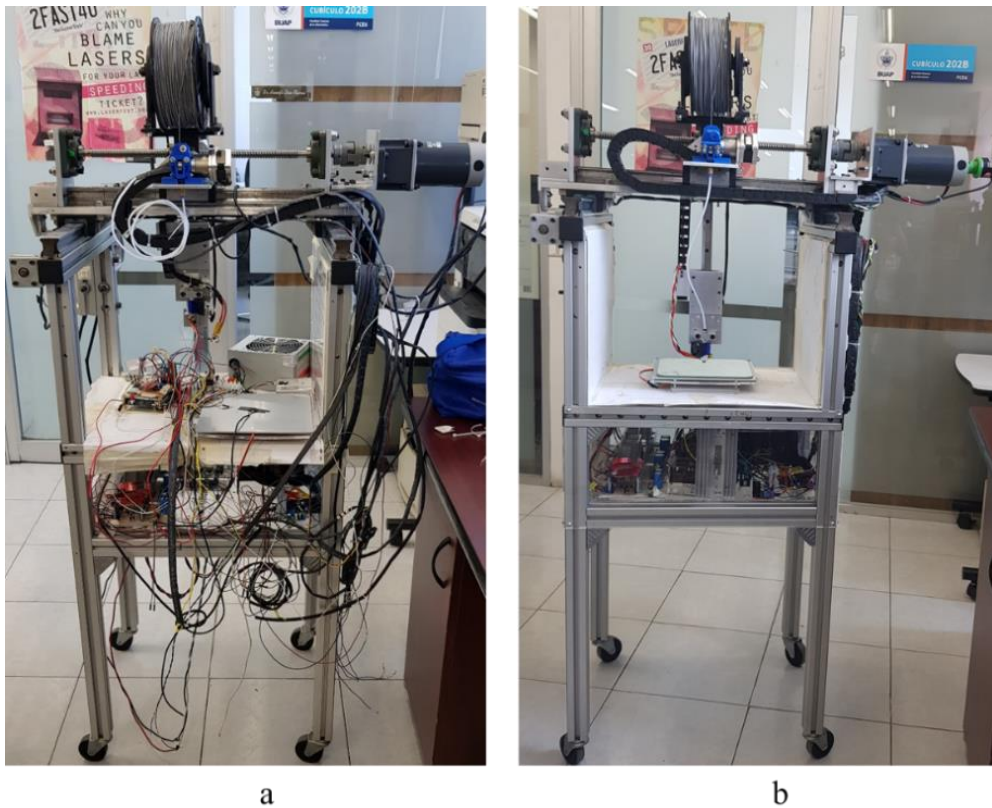


Figura 95. Sistema mecatrónico: a) Antes, b) Después.

5.2 Control de posición

Para llevar a cabo el control de posición en los 3 grados de libertad, se asignó una secuencia de 5 posiciones deseadas para cada eje, como se muestra en la tabla 22.

En las figuras 96, 97 y 98 se presentan las gráficas de posición actual, error de posición y velocidad obtenidas al realizar control de posición en los ejes X, Y y Z, respectivamente. Donde, para los ejes X y Y se implementó el algoritmo de control (52), para el eje Z el (54) y sus respectivas ganancias se muestran en la tabla 23, además la velocidad se determinó mediante la ecuación (53).

Tabla 22. Posiciones deseadas en X, Y, Z.

Parámetro (mm)	X	Y	Z
Posición deseada 1	84.75	95.6	70.05
Posición deseada 2	125.5	146.7	121.01
Posición deseada 3	225.9	230.5	140.05
Posición deseada 4	71.5	45.5	94.07
Posición deseada 5	0	0	0

Tabla 23. Ganancias para control de posición en X, Y, Z.

Ganancias	X	Y	Z
Kp	300	150	250
Ks	0.14	0.14	0.14
Kv	50	25	-

Como se puede observar en la figura 96, el error de posición en el eje X es menor a 400 micras y se mueve a una velocidad entre 33-35 mm/s. Cabe mencionar que para obtener los datos de cada gráfica se tuvieron que hacer distintas pruebas para poder guardar los datos de manera individual, ya que no se cuenta con la opción de poder almacenar distintas variables en diferentes archivos a la vez.

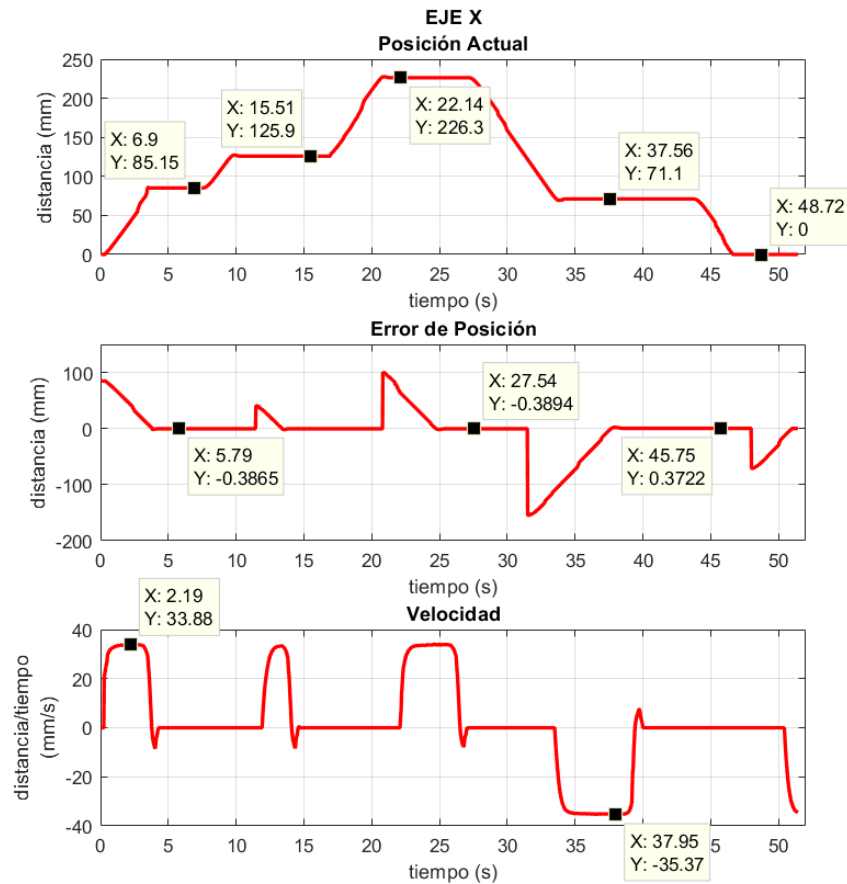


Figura 96. Control de posición en el eje X.

En cuanto al control de posición para el eje Y, se logra observar que el error de posición es mayor que el que se obtuvo en el eje X, con poco más de 400 micras y su velocidad de movimiento es de 24 mm/s.

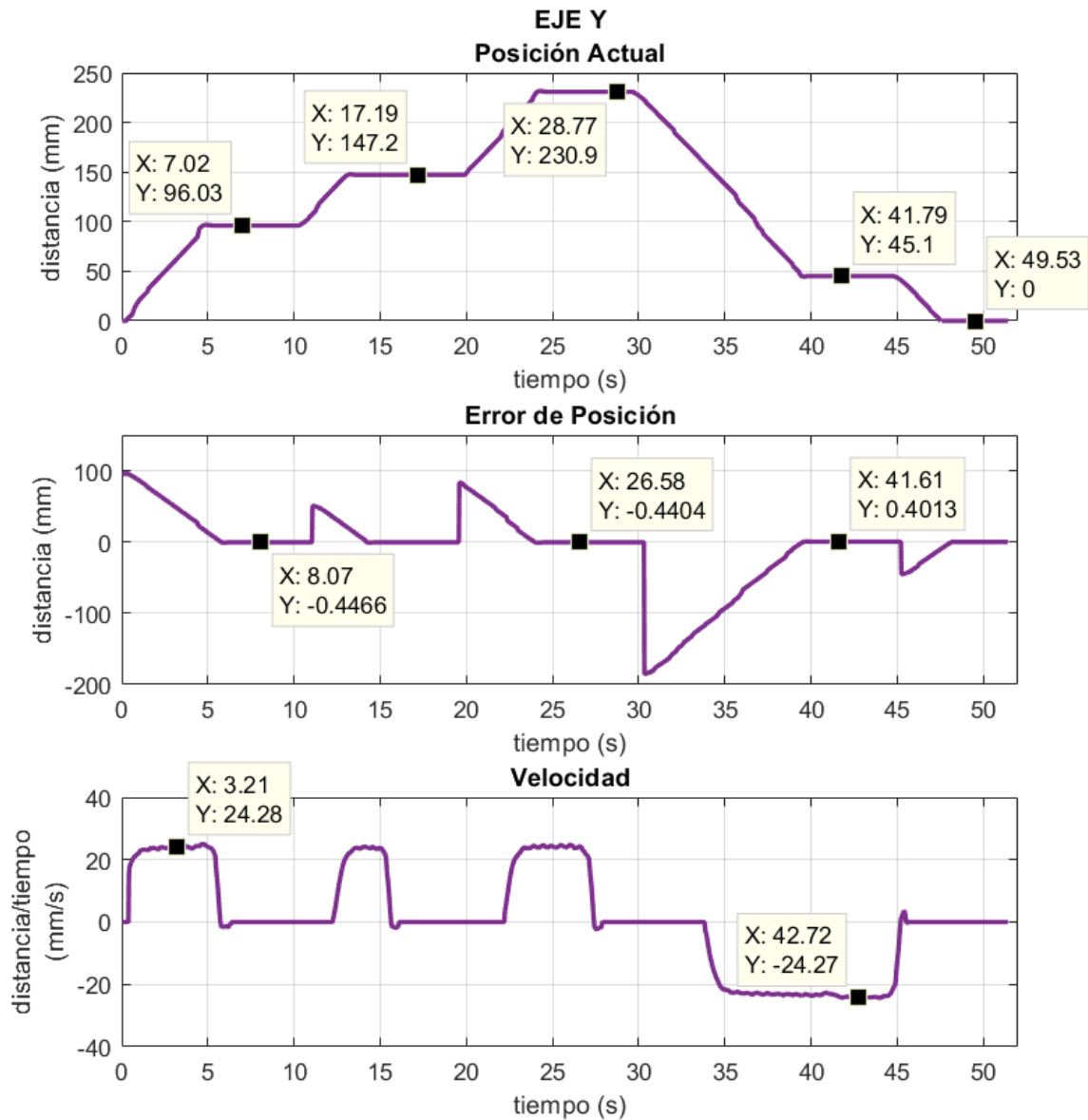


Figura 97. Control de posición en el eje Y.

Finalmente, el control de posición para el eje Z se logró realizar de manera más precisa alcanzando un error de hasta 10 micras, además su velocidad es de 21 mm/s.

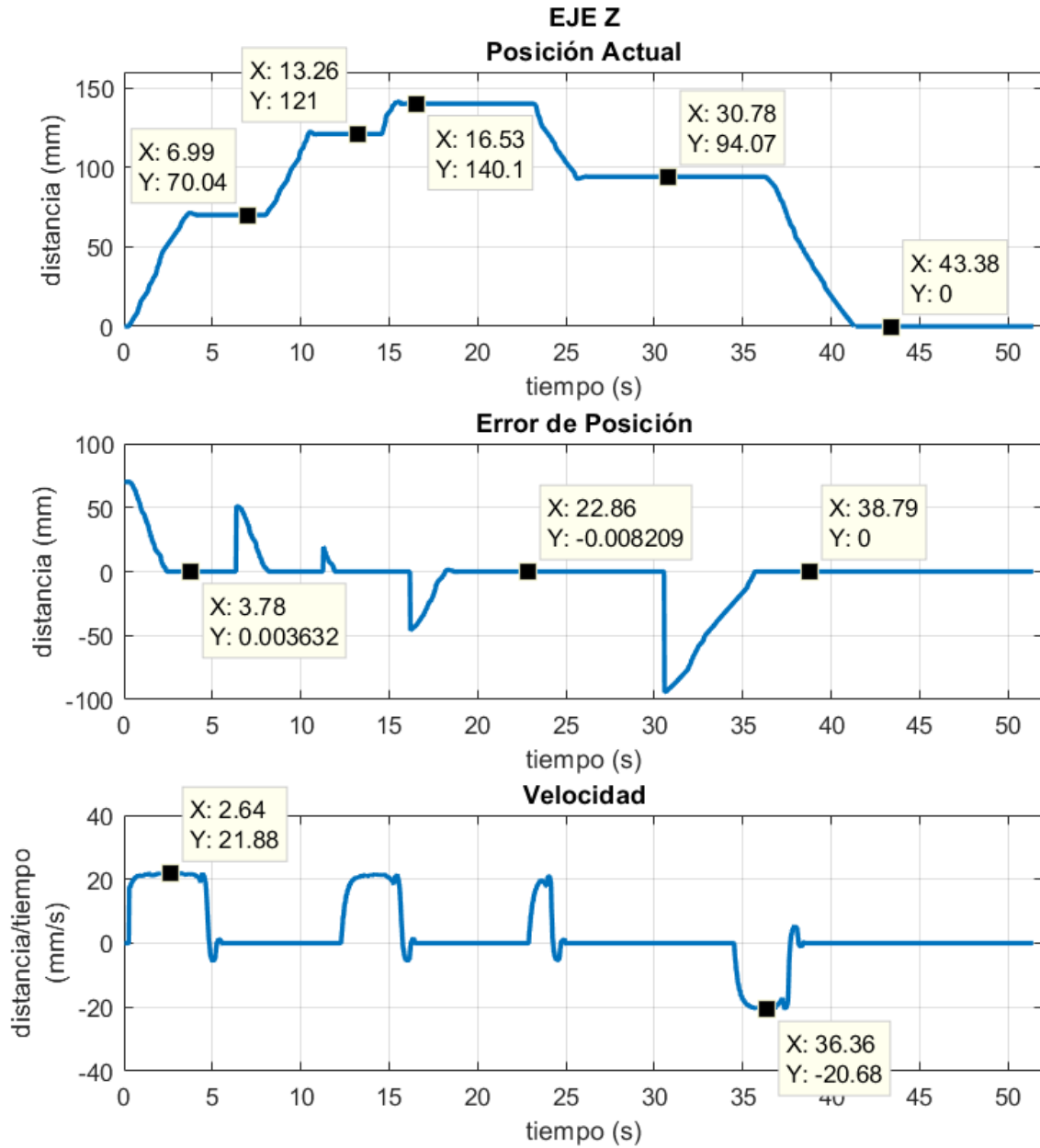


Figura 98. Control de posición en el eje Z.

5.3 Control de velocidad

El algoritmo de control implementado para el control de velocidad es el (55), el cual solo se aplicó para los ejes X y Y. Para cada eje se hicieron pruebas con 3 velocidades diferentes, 20 mm/s, 30 mm/s y 40 mm/s, cuyos resultados se muestran en la figura 99 para el eje X y en la figura 100 para el eje Y. De igual manera, sus respectivas ganancias se presentan en la tabla 24.

Tabla 24. Ganancias para control de velocidad en X y Y.

Ganancias	X	Y
Kp	5	5
Ks	0.1	0.1

En cuanto al eje X, se tiene una distancia de desplazamiento de 277 mm, por lo tanto, en la figura 99 se puede observar que a distintas velocidades se logra recorrer la misma distancia en menor tiempo.

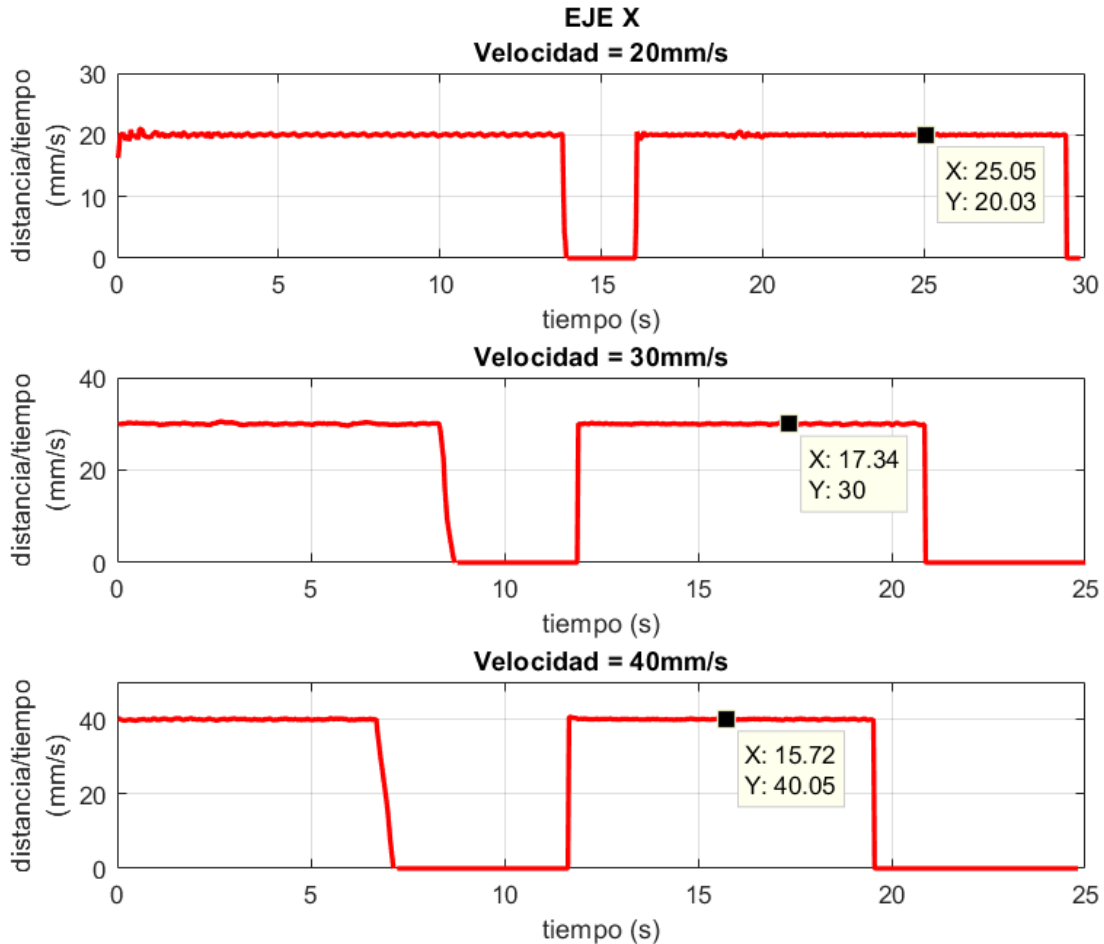


Figura 99. Control de velocidad en el eje X.

Para el eje Y se tiene una distancia de desplazamiento de 245 mm, por lo tanto, en la figura 100 se puede observar que a mayor velocidad es menor el tiempo que le toma recorrer dicha distancia.

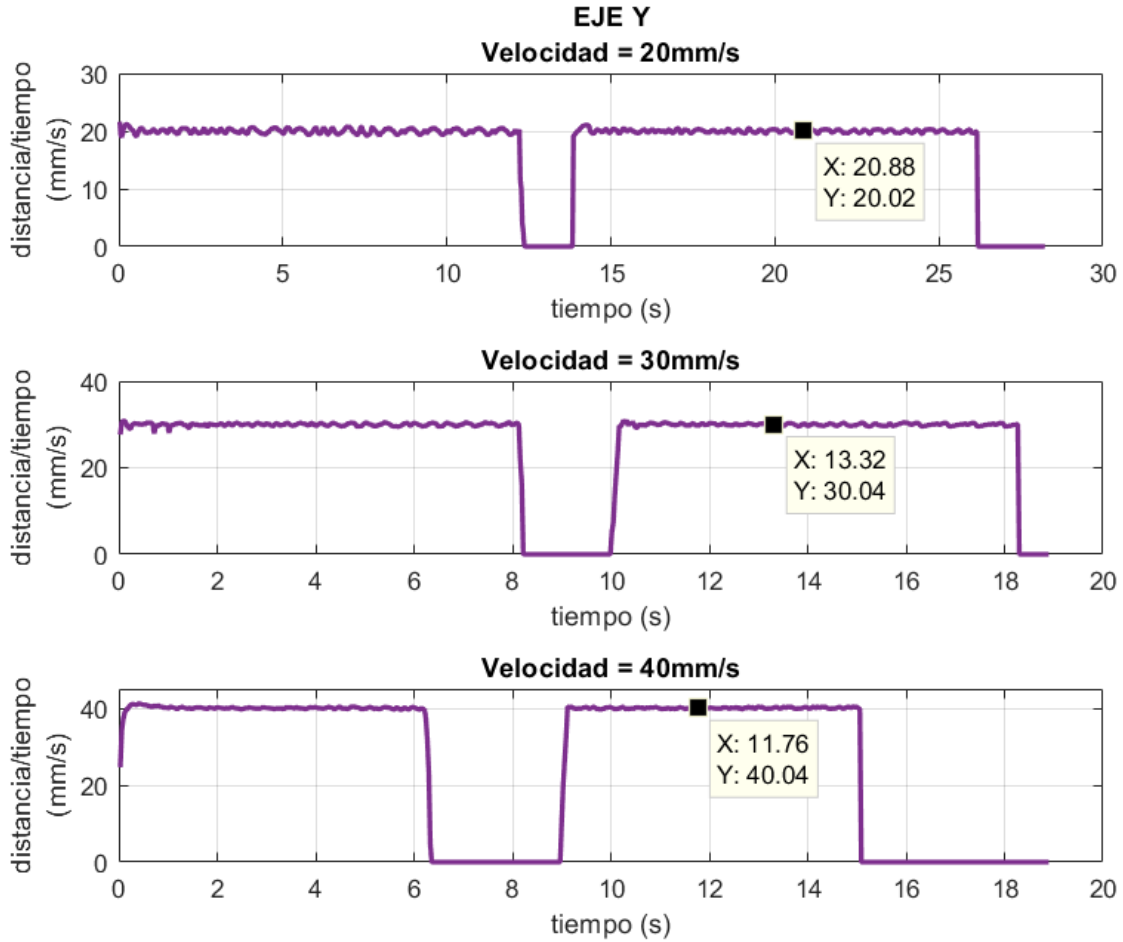


Figura 100. Control de velocidad en el eje Y.

5.4 Control de temperatura

En las figuras 101 y 102 se presenta la respuesta de la base y el extrusor, respectivamente. Las ganancias que se asignaron para lograr un control de temperatura estable se muestran en la tabla 25.

Tabla 25. Ganancias para control de temperatura.

Ganancias	Base	Extrusor
Kp	800	900
Ks	1.47	1.47

Para el caso de la base, esta se tiene que controlar a 60° C ya que eso es lo necesario para que el filamento extruido se mantenga adherido a la cama caliente y no se seque de momento ya que podría provocar grumos o pandeos en la pieza a imprimir.

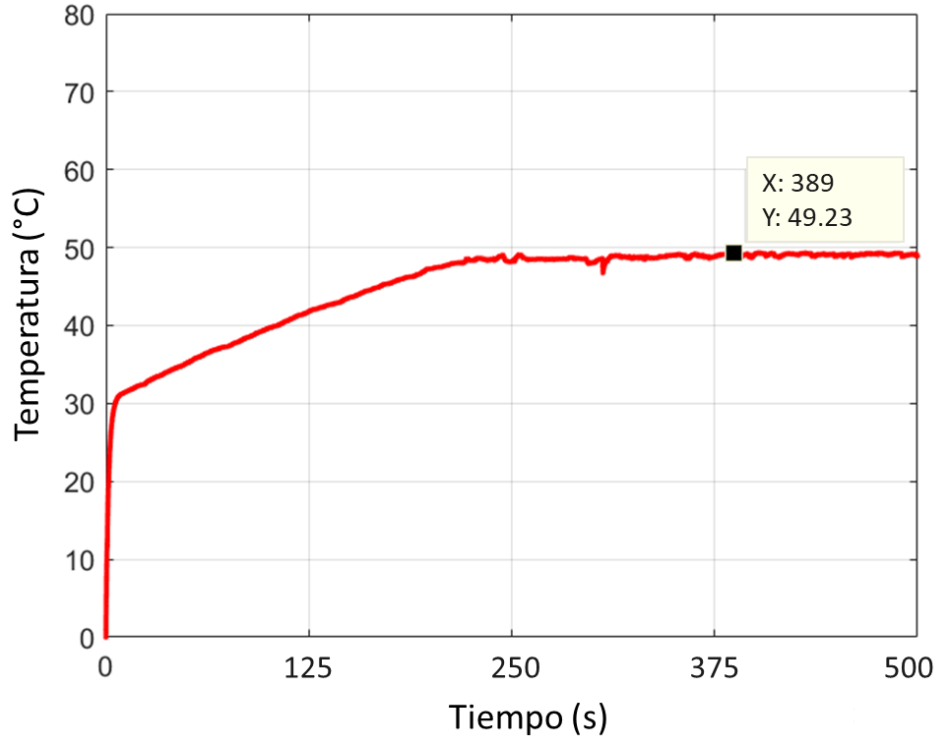


Figura 101. Control de temperatura de la base a 60° C.

Por otra parte, ya que el filamento utilizado para realizar la impresión 3D es PLA y su rango de temperatura va de los 190-210° C, se decidió controlarlo el extrusor a 200° C.

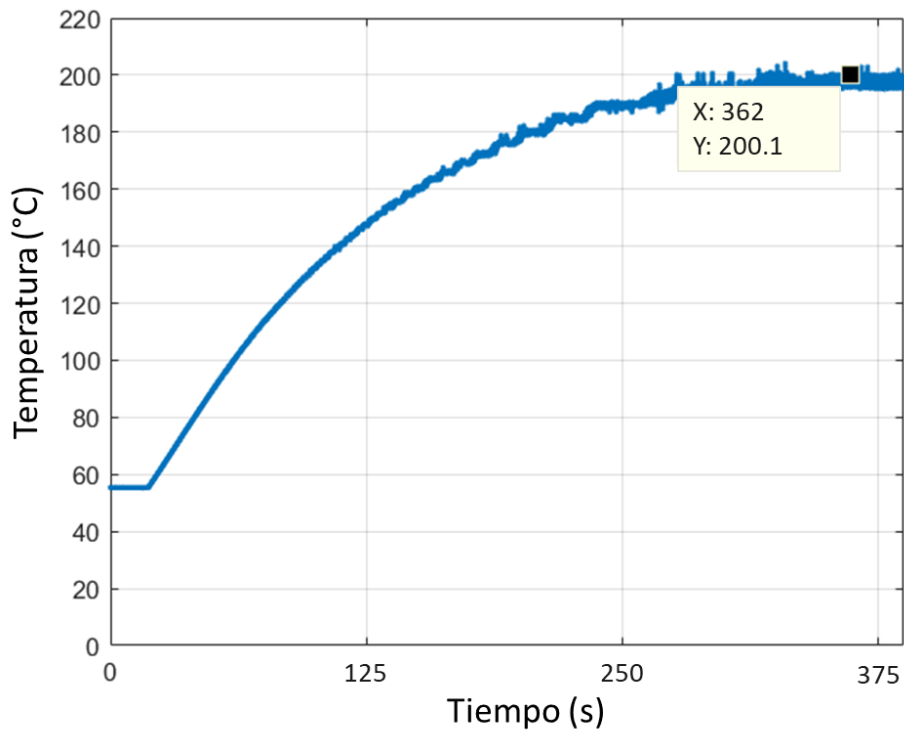


Figura 102. Control de temperatura del extrusor a 200° C.

5.5 Impresión 3D

Para realizar la impresión 3D se sigue la lógica mostrada en el diagrama de flujo de la figura 103. En este caso, se desarrolló un código que se explicará mediante 5 pasos (bloques encerrados en azul).

- 1) Se hace la lectura de los parámetros de inicio que el usuario asigna en la interfaz de usuario en labview para ser mandados al microprocesador.
- 2) Se realiza el control de temperatura de la base y del extrusor, en el cual la base debe calentar a 60° C y el extrusor a 200° C; una vez que ambos hayan llegado al rango de temperatura deseada se procede al siguiente paso.
- 3) El robot se posiciona a casa. Para este caso, se toma como posición de casa cuando el interruptor de atrás del eje X, el interruptor izquierdo del eje Y y el interruptor de arriba del eje Z están activados.
- 4) Una vez que se llegó a posición de casa, se enciende el motor a pasos, se vuelve a controlar la temperatura de la base y el extrusor a las temperaturas antes mencionadas para que el filamento pueda caer con ayuda del motor a pasos y se inicia la secuencia de posiciones para formar la pieza en 3D. Estas posiciones serán comandadas a través del archivo que se extrajo del compilador de posiciones las cuales fueron almacenadas en la memoria RAM. En la figura 104 se explica a detalle el procedimiento que se sigue para la secuencia de posiciones.
- 5) Se apagan los actuadores y se sale del código de control.

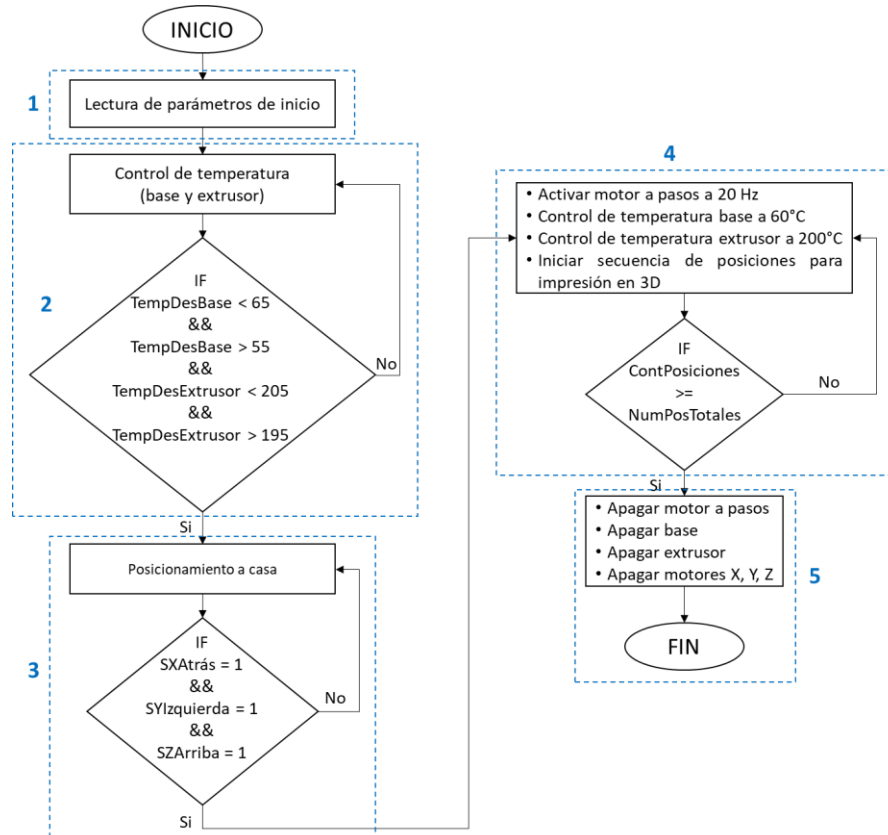


Figura 103. Diagrama de flujo para impresión 3D.

La estrategia que se implementó para realizar la secuencia de pasos que dará lugar a la figura en 3D se explica en el diagrama de flujo de la figura 104. Para esto, primero se leen las etiquetas de los ejes X, Y y Z, las cuales se ponen en alto cuando el error de posición en cada eje es cercano a 0, y se suman los valores, si la suma da 3 quiere decir que los 3 ejes han llegado a la posición deseada, de ser así se resetean las etiquetas, se incrementa el contador de posiciones totales (ContPosiciones) y el contador que se utiliza como offset para asignar la dirección de la memoria RAM que se quiera leer para la siguiente posición deseada (ContDirRam), de lo contrario, se pasa a la siguiente condición.

Si el contador de posiciones totales es mayor o igual al número de posiciones totales quiere decir que ya no hay más posiciones por hacer, por lo tanto, el programa termina, de lo contrario se sigue con el control. Finalmente, para cada eje se lee la posición deseada, se realiza el control (control de posición o velocidad) y se calcula el error, si el error se encuentra en el rango determinado se activa la etiqueta, si no, continua con el control.

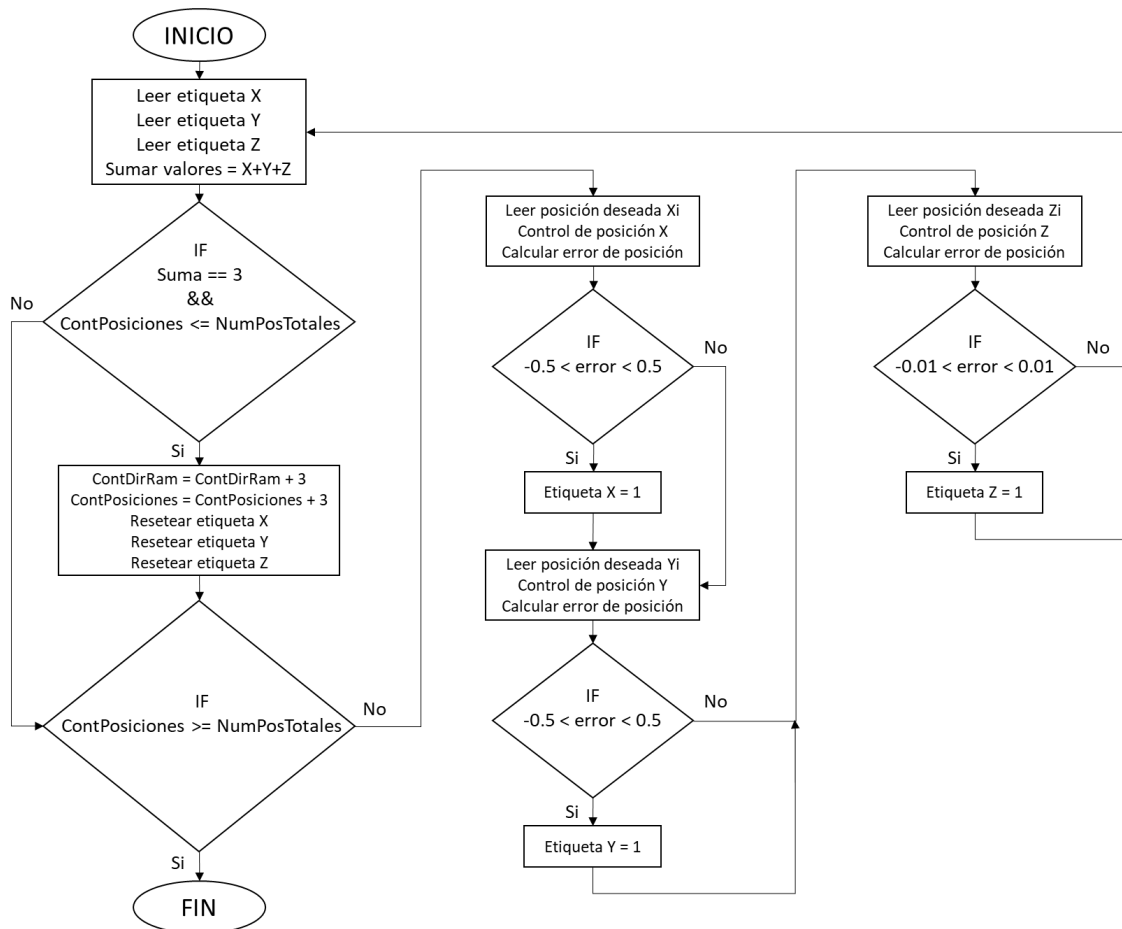


Figura 104. Secuencia 3D para leer posiciones.

Para realizar las pruebas de impresión 3D y verificar si la lógica implementada para la secuencia de posiciones deseadas es ideal, se diseñó una malla de 6x6cm con separaciones de 1cm, como se muestra en la figura 105. A partir de los puntos tomados para formar esta figura se procedió a implementar las estrategias mostradas en los diagramas de flujo de las figuras 103 y 104, utilizando primero control de velocidad y posteriormente control de posición.

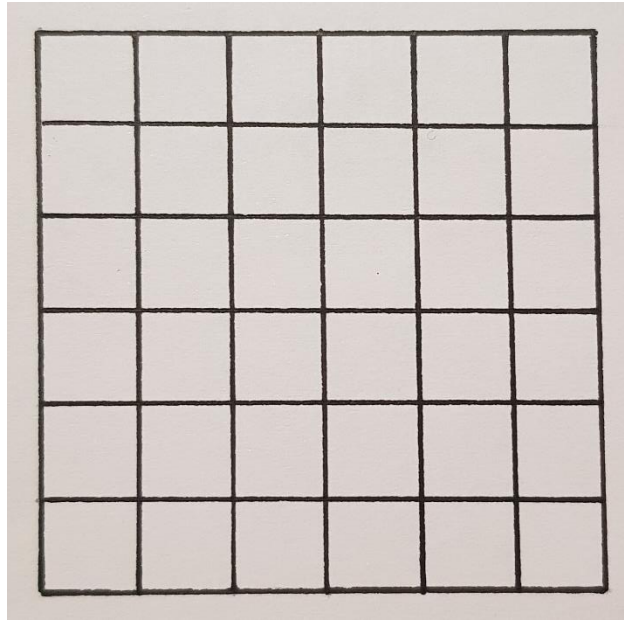


Figura 105. Boceto de malla como ejemplo.

5.5.1 Impresión 3D con control de velocidad

Primero se hizo la prueba de impresión con velocidad de 40 mm/s haciendo 2 capas, cuyo resultado se muestra en la figura 106.

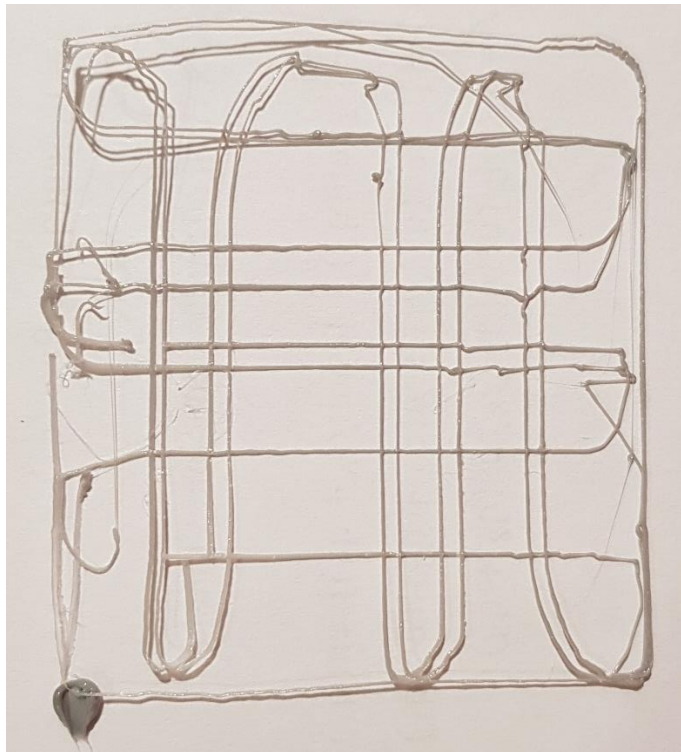


Figura 106. Impresión de malla a 40 mm/s.

Al observar que la impresión salió muy desfasada se decidió reducir la velocidad a 30 mm/s, lo cual se puede observar en la figura 107.

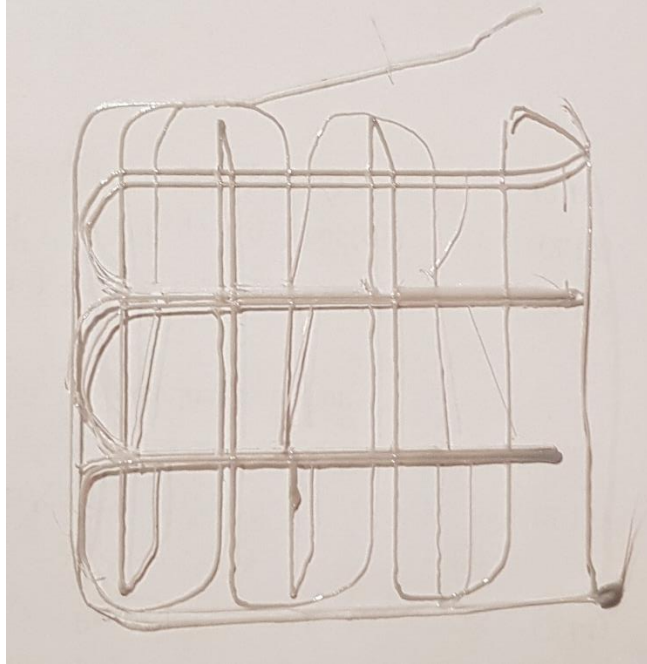


Figura 107. Impresión de malla a 30 mm/s.

Finalmente, se hizo la prueba a velocidad de 20 mm/s para ver si se obtenía alguna mejora en el resultado de la impresión, ver figura 108.



Figura 108. Impresión de malla a 20 mm/s.

5.5.2 Impresión 3D con control de posición

Pasando a la impresión 3D de la malla (figura 105) con control de posición se logra tener una mejor precisión, de modo que, la impresión de 2 capas mostrada en la figura 109 logra salir mucho mejor comparándola con las obtenidas con control de velocidad.

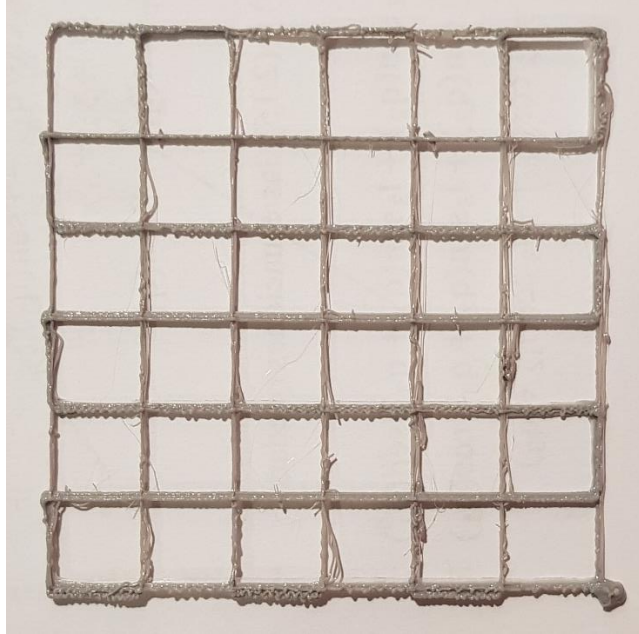


Figura 109. Impresión de malla con control de posición (2 capas).

Al obtener mejores resultados de impresión mediante control de posición, se decidió aumentar el número de capas a 5 y posteriormente a 10, quedando como las mostradas en las figuras 110a y 110b, respectivamente.

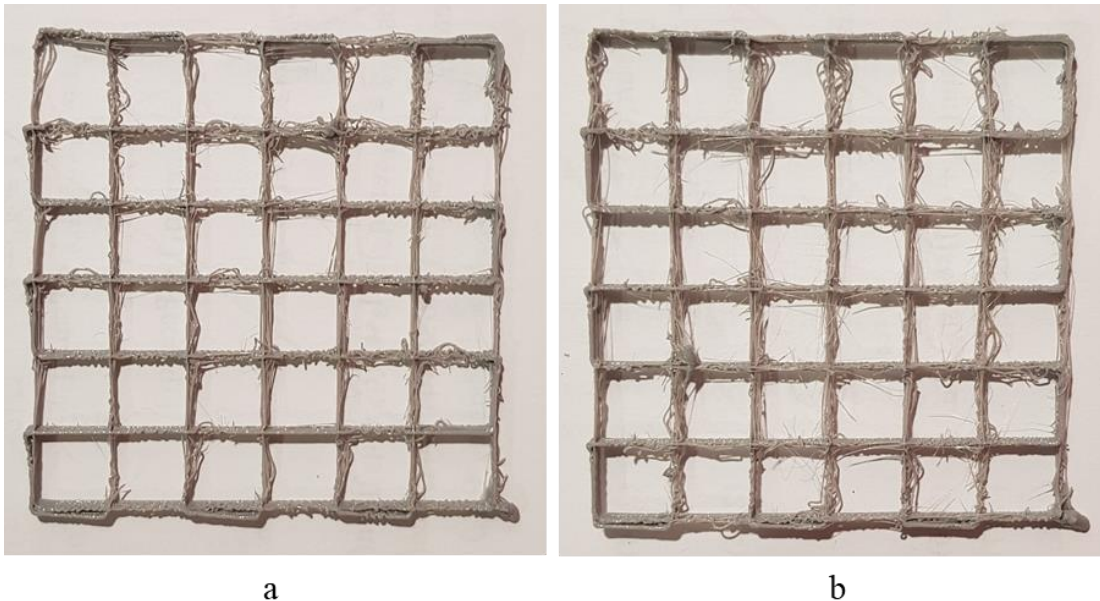


Figura 110. Impresión de malla con control de posición: a) 5 capas, b) 10 capas.

5.5.3 Tabla comparativa de impresoras 3D industriales

En la tabla 26 se presenta una tabla comparativa de nuestra impresora con otras impresoras 3D industriales más recientes que se pueden encontrar en el mercado [8].

Tabla 26. Comparación de impresoras 3D industriales.

Impresora Descripción	I3D MCEA	BigRep Studio	Industry F340	DT 600+	Mark X7
Volumen de impresión (mm)	214 × 214 × 170	500 × 1000 × 500	260 × 300 × 340	600 × 450 × 450	250 × 250 × 250
Resolución (μm)	XY: 400 Z: 20	100	XY: 60 Z: 40	30	50
Velocidad (mm/s)	40	140	-	-	-
Cama caliente	Si	Si	Si	Si	No
Extrusor	1	2	2	2	1
Materiales de impresión	PLA	PLA, filamentos especiales	PLA, ABS, plástico	PLA, ABS, filamentos especiales, polyamida PA12	Resina, plástico
Tamaño (mm)	950 × 860 × 1470	1022 × 1660 × 1500	945 × 748 × 918	1140 × 740 × 970	787 × 737 × 1829
Software	Abierto	Simplify3D, Cura, Slic3r	3DGence Slicer	Simplify3D	Eiger
Conectividad	Wifi	Cable USB, ethernet	Cable USB, tarjeta SD, USB	Wifi, ethernet, cable USB	Ethernet
Sistemas operativos	Abierto	Windows, Mac OSX, Linux	Windows, MacOS	Windows, Mac OS	Windows
Precio (MXN)	\$30,000	\$725,000	\$875,000	\$750,000	\$2.5-6.25 millones

Conclusiones generales

De acuerdo a los resultados obtenidos en el desempeño del robot desarrollado podemos concluir que funciona adecuadamente para el propósito planteado en esta tesis.

La instrumentación de una estrategia segmentada del algoritmo de control permitió alcanzar una precisión más alta en el eje Z, esto fue debido a que cuando la posición deseada varía unas cuantas micras de la posición actual, el error es tan pequeño que para producir un movimiento la ganancia del algoritmo de control debe ser muy alta para producir el movimiento, esto originó que tuviéramos que segmentar el algoritmo de control en tres secciones mejorando por mucho el desempeño de este eje.

En cuanto al control de velocidad en los ejes X y Y, se logra alcanzar una velocidad de hasta 40 mm/s en cada eje. Por otra parte, con control de posición, en el eje Z se alcanzó una elevada precisión de 20 micrómetros, sin embargo, en X y Y no se logró obtener una precisión tan alta, alcanzando cerca de 400 micras. Sería importante continuar a futuro con este trabajo para lograr una precisión en X y Y del mismo orden que Z, ya que las aplicaciones con un robot con estas características permitirían producir máquinas para recoger y colocar circuitos electrónicos (conocidas como pick and place), máquinas para alinear circuitos integrados con empaque BGA, etc.

Este robot puede utilizarse como plataforma experimental para realizar pruebas con distintos algoritmos de control. Gracias a los interruptores de fin de carrera que se le adaptaron para evitar que el carro de precisión sobrepase los límites físicos de cada eje se evitará provocar daños en la estructura mecánica. Además, por tratarse de motores de transmisión directa, el sistema mecatrónico se controla mediante lazo cerrado, lo cual es de gran ventaja ya que después de varios usos el movimiento seguirá siendo igual, no habrá desgaste mecánico ni pérdida en la posición de los ejes, haciendo al robot de uso robusto.

La implementación del compilador de posiciones deseadas desarrollado en LabView facilita el hecho de poder crear secuencias de posiciones o rutinas para realizar pruebas de control sin necesidad de tener que imprimir una pieza en 3D. Además, es de gran ayuda a la hora de convertir archivos con demasiadas posiciones, como lo son los códigos G que se obtienen de algún software laminador.

Mediante la interfaz de usuario, que a través de protocolo de comunicación Wi-Fi permite la comunicación entre la computadora y el microprocesador embebido en la tarjeta de adquisición (FPGA), se logra llevar a cabo el control de todo el sistema. De igual manera, dicha interfaz da la posibilidad de mandar comandos para variar ganancias o asignar parámetros deseados (posición, velocidad, temperatura), leer puertos en tiempo real e incluso almacenar datos para posteriormente procesarlos y poder observar el comportamiento del sistema.

El control de temperatura, tanto del extrusor como de la base, fue muy laborioso de realizar ya que se trata de control no lineal, pero no porque la respuesta del control sea no lineal, sino por la naturaleza de los termistores que al caracterizarlos se pudo observar que su comportamiento es de forma logarítmica. Debido a esto, se tuvieron que hacer muchas

pruebas y ajustes en el control de temperatura de cada planta, sobre todo para el control del extrusor, ya que el termistor presenta una respuesta en la que a mayor temperatura se tiene menor resolución, es por eso que para controlar a 200 °C se tuvo que amplificar el voltaje de salida para que el ADC pudiera leerlo con mayor facilidad.

Finalmente, nuestro robot en el futuro también podrá utilizarse como una máquina multiherramienta, gracias a los motores de transmisión directa que se le implementaron en los ejes X, Y y Z, los cuales le darán la fuerza de empuje necesaria para poder devastar madera y gracias a esto, se podrán crear tarjetas de circuito impreso. De igual manera, otra de las ventajas que presenta este robot es que su arquitectura es abierta en comparación con los que se encuentran en el mercado que son de secreto industrial. Esto da la ventaja de poder desarrollar e implementar software y hardware propio o hasta incluso poder crear una aplicación para Android que permita hacer la configuración del robot para realizar el proceso deseado.

Apéndices

Apéndice A: Constancia de Ponencia



Apéndice B: Artículo publicado

Robótica y Computación. Investigación y Desarrollo.

Desarrollo de un sistema de control para un robot cartesiano de 3 grados de libertad mediante un sistema embebido para que funcione como impresora 3D.

C. Jiménez, S. Vergara, M. A. Vargas, F. Reyes.
Facultad de Ciencias de la Electrónica opción en Automatización.
Benemérita Universidad Autónoma de Puebla.
Puebla, Pue. México.
caro.jimenez.martinez@hotmail.com

Resumen—En este trabajo se presenta el desarrollo mecánico, la cinemática directa y el modelo dinámico para un robot cartesiano de 3 grados de libertad (gdl). Se describe el desarrollo del sistema de control para dicho robot mediante un sistema embebido basado en una tarjeta FPGA (Field Programmable Gate Array) de la familia Cyclone V de Altera, el cual le permite controlar la posición de cada motor, al igual que le permite controlar la temperatura deseada de la cama caliente y del extrusor de material, de modo tal que el robot cartesiano de 3 gdl pueda funcionar como impresora 3D. Además, se presenta la caracterización del termistor que trae integrado tanto el extrusor de material como la cama caliente mediante el cual se puede determinar la temperatura actual del sistema para después cerrar el lazo de control de temperatura empleando el algoritmo de control tangente hiperbólica.

Palabras clave—FPGA, impresora 3D, control robot cartesiano.

Abstract—This paper presents the mechanical development, the direct kinematics and the dynamic model for a cartesian robot with 3 degrees of freedom (dof). The development of the control system for said robot is described by means of an embedded system based on an Altera Cyclone V FPGA (Field Programmable Gate Array), which allows it to control the position of each motor, as well as to control the temperature desired from the heatbed and the material extruder, so that the 3 dof cartesian robot can function as a 3D printer. In addition, the characterization of each thermistor is presented, which includes both the material extruder and the heatbed through which the current temperature of the system can be determined to then close the temperature control loop using a hyperbolic tangent control.

Keywords—FPGA, 3D printer, cartesian robot controller.

I. INTRODUCCIÓN

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, esto se debe a que mientras las empresas nacionales tienen que importar tecnología en muchos casos obsoleta, las empresas extranjeras usan tecnología de punta en sus procesos de producción. Por lo

anterior, es muy importante ganar experiencia en el desarrollo de máquinas que en un futuro puedan usarse en sistemas automatizados de producción.

Los robots manipuladores se encuentran mayormente en lo que es el área industrial ya que ofrecen ventajas considerables por el hecho de que pueden realizar tareas difíciles o peligrosas para el ser humano, así como también pueden realizar dichas tareas en menor tiempo y con mayor calidad y precisión. Dentro de los robots manipuladores se encuentra el robot cartesiano, el cual es uno de los dispositivos de mayor uso en las máquinas automatizadas gracias a que puede generar trayectorias complejas en un espacio tridimensional a partir de movimientos lineales independientes. De esta forma, pueden ser empleados para manipular objetos o en máquinas-herramienta de control numérico computarizado (CNC) como las cortadoras láser, perforadoras de placas, soldadura, centros de mecanizado o incluso como impresora 3D.

Las impresoras 3D se han vuelto muy populares hoy en día, gracias a las ventajas que ofrecen al poder realizar réplicas de modelos diseñados por ordenador convirtiéndolos en piezas 3D de distintos tamaños y formas según la aplicación para la cual se requieran. Debido a la versatilidad de este tipo de máquinas se les puede emplear casi en cualquier sector, por ejemplo, en los hospitales se ha empleado esta técnica de impresión 3D para la fabricación de órganos mediante materiales especiales, por otra parte, incluso se han desarrollado impresoras 3D para construir casas pequeñas de concreto.

Uno de los problemas más comunes que suelen tener las impresoras 3D de escritorio es que la posición de sus ejes se descalibra fácilmente debido a que funcionan mediante motores a pasos, es decir, en lazo abierto. Es por eso que se decidió realizar una impresora 3D que funcione mediante motores DC con encoders de cuadratura para determinar su posición actual (lazo cerrado), además de que se empleará una tarjeta FPGA para llevar a cabo el control, lo cual la vuelve única en su tipo, de modo que, mediante este sistema embebido se logren controlar los 3 gdl y la temperatura tanto de la cama caliente

como del extrusor de material, al igual que hacer monitoreo y diagnósticos del sistema en tiempo real.

II. DIAGRAMA GENERAL DEL SISTEMA MECATRÓNICO

En la Fig. 1 se muestra el diagrama general del sistema mecatrónico, el cual se compone de un robot cartesiano de 3 gdl que puede ser manipulado por el usuario mediante comunicación wifi a través de una interfaz desarrollada en LabView donde se pueden modificar y visualizar los parámetros del sistema, como posiciones deseadas y ganancias de ajuste para los controladores.

Para que el robot se comunique con el sistema digital, se implementó una FPGA Cyclone V de la familia altera, la cual ejecutará los algoritmos de control para controlar la posición de los 3 gdl del robot cartesiano (x, y, z), así como también se encargará de mantener la temperatura deseada del extrusor de material y de la cama caliente donde se irá depositando el material extruido llevando a cabo la acción de impresora 3D.

En el caso de los 3 gdl, la FPGA se encargará de leer los encoders de cuadratura de cada servomotor para determinar su posición actual, aplicará la ley de control y mediante el generador de PWM (Pulse Width Modulation) mandará la señal a la etapa de potencia (puente H) de cada motor el cual proporcionará la energía necesaria a cada actuador para que se muevan a la posición deseada por el usuario.

En cuanto al extrusor de material y la cama caliente, estos traen integrado una resistencia eléctrica para calentarse y un termistor mediante el cual se determina la temperatura actual del dispositivo. Para que la FPGA pueda leer estas señales y procesarlas se debe hacer una conversión de la señal analógica a digital para lo cual se implementaron ADC0804 de 8 bits (uno para cada uno), se aplica la ley de control y mediante el generador de PWM se manda la señal de activación a la etapa de potencia (transistores de potencia), tanto del extrusor como de la cama caliente, proporcionando la energía necesaria para mantener cada actuador a la temperatura deseada por el usuario.

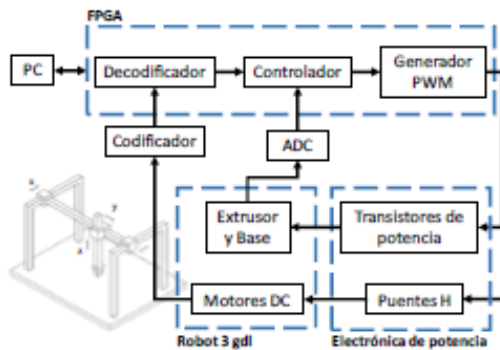


Fig. 1. Diagrama general del sistema mecatrónico.

III. MECÁNICA DEL ROBOT

Se cuenta con un robot cartesiano de 3 gdl en el cual para poder realizar los movimientos prismáticos de este sistema con suma precisión se utilizaron rieles de alta precisión apoyados con carros lineales de precisión (Fig. 2a), esto con el fin de poder obtener el mejor desplazamiento lineal con mínima fricción en el sistema.

De igual manera, cuenta con tornillos embalados (Fig. 2b) para transformar el desplazamiento giratorio a lineal, ya que 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.

También se le adaptaron interruptores de fin de carrera (Fig. 2c) a cada eje para evitar que el carro de alta precisión sobrepase los límites del área de trabajo y dañe el sistema mecánico.



Fig. 2. Componentes mecánicos: a) Riel de alta precisión y carro de alta precisión, b) Tornillo embalado, c) Interruptor de fin de carrera.

Para los actuadores de los 3 gdl (ejes X, Y y Z) se utilizan motores reductores Z5D120-12 (Fig. 3a) que se alimentan a 12V, cuentan con la característica de proporcionar 3100 rpm (revoluciones por minuto) y generan un torque de 3-4 Nm [1]. Para poder ocupar los motores DC como servomotores tienen acoplado un encoder giratorio GTS06-VE-RA 1000 (Fig. 3b) de 1000 pulsos/revolución en la parte posterior de cada motor, mediante el cual se puede determinar la posición actual del actuador.



Fig. 3. Actuadores para los 3 gdl: a) Motor reductor, b) Encoder giratorio.

En cuanto a la herramienta de extrusión se implementó un extrusor de material Metal J-Head (Fig. 4a) el cual es alimentado con filamento de plástico (material a extruir, PLA) mediante un motor a pasos bipolar de cambio rápido de filamento, modelo EXT V22SMBU (Fig. 4b). Dicho extrusor cuenta con un termistor como sensor de temperatura de 100 K tipo NTC (Negative Temperature Coefficient), es decir, que al aumentar la temperatura su resistencia disminuye [2].

Finalmente, se le agregó una cama caliente MK3 (Fig. 4c) en la cual se irá depositando el filamento extruido para ir formando la pieza en 3D, esto con el fin de mantener el material caliente durante la impresión y así evitar pandeo en la pieza. Esta cama

tiene la característica de contar con fuente de alimentación doble, es decir, que se puede operar con 12V o 24V [3].

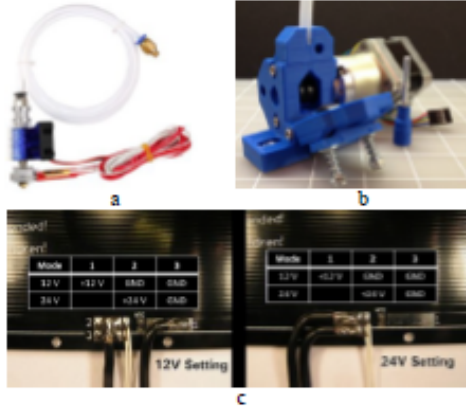


Fig. 4. Herramienta de extrusión: a) Extrusor de material, b) Alimentador de material, c) Cama caliente.

IV. DISEÑO DE LA CINEMÁTICA Y DINÁMICA

Es importante determinar el modelo dinámico para realizar un control más apropiado del robot y que además presente un mejor desempeño, de igual manera, es muy útil para analizar su comportamiento con diferentes algoritmos de control.

En la Fig. 5 se muestra la representación geométrica del robot cartesiano de 3 gdl, el cual permitirá visualizar el comportamiento del manipulador. Tanto el modelo de cinemática directa como el modelo dinámico se tomaron de [4].

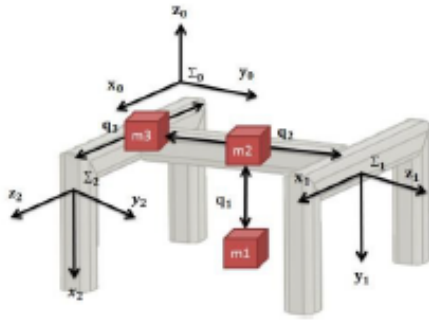


Fig. 5. Representación del robot cartesiano de 3 grados de libertad.

A. Cinemática

Para determinar la cinemática directa del robot cartesiano de 3 gdl se implementó la metodología de Denavit-Hartenberg [5], la cual ofrece un procedimiento sencillo para obtener la mínima representación de la orientación y la traslación de robots manipuladores y cuya estructura queda en representación de transformaciones homogéneas.

En la tabla I se presentan los parámetros de Denavit-Hartenberg.

TABLE I. PARÁMETROS DE DENAVIT-HARTENBERG.

Eslabon	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

En donde, los cuatro parámetros Denavit-Hartenberg se definen de la siguiente manera:

- l_i es la longitud del i -ésimo eslabón.
- α_i es el ángulo de torsión.
- d_i es el desplazamiento de la articulación (offset).
- θ_i determina el ángulo de la articulación.

En la convención Denavit-Hartenberg cada transformación homogénea se representa mediante el producto de cuatro transformaciones básicas, como se muestra en (1).

$$H_{i-1}^i = H_{Rz_{i-1}}(\theta_i)H_{Tx_{i-1}}(d_i)H_{Rz_{i-1}}(l_i)H_{Rx_{i-1}}(\alpha_i) \quad (1)$$

La matriz de transformación homogénea para el robot cartesiano de 3 gdl queda como en (2).

$$H_0^3 = H_0^1 R_{z_0}^T H_1^2 H_2^3 = \begin{bmatrix} 0 & 0 & 1 & d_3 \\ 0 & -1 & 0 & d_2 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Por lo tanto, se tiene que el modelo de cinemática directa para el robot cartesiano de 3 gdl queda como se muestra en (3).

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} d_3 \\ d_2 \\ d_1 \end{bmatrix} \quad (3)$$

B. Dinámica

El modelo dinámico cartesiano está dado por (4).

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + f_f(\dot{q}, f_e) \quad (4)$$

donde,

- $q \in \mathbb{R}^n$ es el vector de coordenadas generalizadas o posiciones articulares.
- $\dot{q} \in \mathbb{R}^n$ es el vector de velocidades articulares.
- $\ddot{q} \in \mathbb{R}^n$ es el vector de aceleraciones articulares.
- $M(q) \in \mathbb{R}^{n \times n}$ es la matriz de inercia, es simétrica y definida positiva.
- $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ es la matriz de fuerzas centrípetas y de Coriolis.
- $g(q) \in \mathbb{R}^n$ es el vector de fuerzas o pares gravitacionales.
- $f_f(\dot{q}, f_e) \in \mathbb{R}^n$ es el vector de pares de fricción que incluye la fricción viscosa, de Coulomb y estática.

Para este caso no se cuenta con la matriz de fuerzas centrípetas y de Coriolis ya que es igual a cero. Por lo tanto, el modelo dinámico para el robot de 3 gdl queda definido por (5).

$$\tau = \begin{bmatrix} m_1 + m_2 + m_3 & 0 & 0 \\ 0 & m_2 + m_3 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{d}_1 \\ \ddot{d}_2 \\ \ddot{d}_3 \end{bmatrix} + g \begin{bmatrix} m_1 + m_2 + m_3 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} + \begin{bmatrix} f_{c1} & 0 & 0 \\ 0 & f_{c2} & 0 \\ 0 & 0 & f_{c3} \end{bmatrix} \begin{bmatrix} \text{signo}(d_1) \\ \text{signo}(d_2) \\ \text{signo}(d_3) \end{bmatrix} + \begin{bmatrix} f_{e1}[1 - |\text{signo}(d_1)|] \\ f_{e2}[1 - |\text{signo}(d_2)|] \\ f_{e3}[1 - |\text{signo}(d_3)|] \end{bmatrix} \quad (5)$$

donde:

- m_1, m_2, m_3 son las masas de los servomotores.
- b_1, b_2, b_3 son los coeficientes de fricción viscosa.
- f_{c1}, f_{c2}, f_{c3} representan los coeficientes de fricción de Coulomb.
- f_{e1}, f_{e2}, f_{e3} son los coeficientes de fricción estática.

V. ELECTRONICA DEL ROBOT

Para la etapa de potencia de cada motor se les diseñó un puente H en configuración Darlington que permita drenar 15A en modo saturación. Para la lectura de los encoders se hizo un divisor de voltaje, ya que los encoders manejan un voltaje de 5V en sus señales de salida y la FPGA solo maneja 3.3V, de esta manera, no se daña la FPGA y la lectura de las señales se logra hacer satisfactoriamente.

En la Fig. 6 se muestra el diagrama a bloques de la electrónica de control que se implementó tanto para el extrusor de material como para la cama caliente. Las etapas de potencia se diseñaron en base a transistores NPN en configuración Darlington para drenar 2.5A en modo saturación para el extrusor de material y 11.6A para la cama caliente.

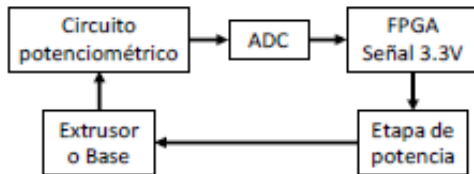


Fig. 6. Electrónica de control para el extrusor de material y la cama caliente.

Para determinar la temperatura actual del extrusor o de la cama caliente se hizo un circuito potenciométrico, formado por el termistor que trae integrado como sensor de temperatura y una resistencia colocados en serie a modo de divisor de voltaje, tomando el punto medio como medida de referencia. Después se caracterizó el termistor de ambos dispositivos variando el ciclo de trabajo de la señal PWM a una frecuencia de 100Hz y pasando 10 minutos de funcionamiento se registraba la temperatura actual del dispositivo (extrusor o base). Al graficar la respuesta de temperatura contra ciclo de trabajo, se puede observar que el comportamiento tanto del extrusor (Fig. 7) como de la cama caliente (Fig. 8) presenta una respuesta lineal.

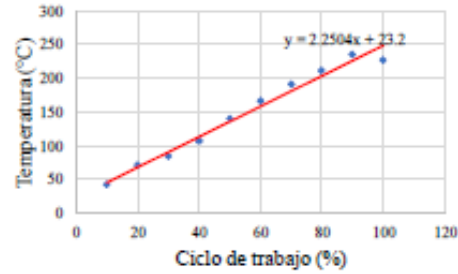


Fig. 7. Gráfica de linealidad del extrusor de material.

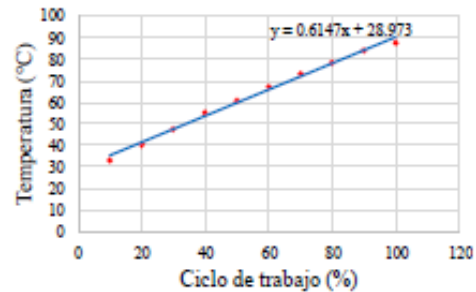


Fig. 8. Gráfica de linealidad de la cama caliente.

El circuito potenciométrico es alimentado por un voltaje que provoca una tensión de salida en el punto medio del divisor de voltaje, mediante el cual, al graficarlo con respecto a la temperatura se obtuvo la ecuación característica a la que trabaja el termistor y con la cual se puede determinar la temperatura a la que se encuentra el extrusor o la base.

La respuesta del termistor, tanto del extrusor de material como del de la cama caliente, Fig. 9 y Fig. 10 respectivamente, presentan un comportamiento no lineal al tener que su ecuación característica se adapta mejor con la curva de ajuste logarítmica.

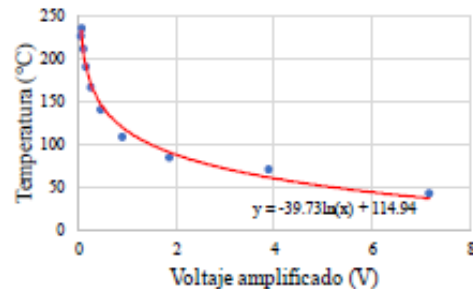


Fig. 9. Gráfica de respuesta logarítmica del termistor del extrusor de material.

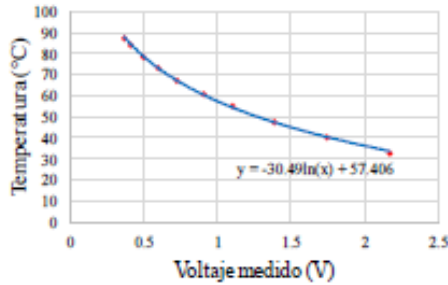


Fig. 10. Gráfica de respuesta logarítmica del termistor de la cama caliente.

Finalmente, para que la FPGA pueda interpretar la señal de los termistores, se utilizaron ADC0804 de 8 bits de resolución, cuyo circuito se diseñó con conexión en modo continuo [6].

VI. IMPLEMENTACIÓN DEL CONTROL EN EL SISTEMA EMBEBIDO

A. Firmware

Para llevar a cabo el control del robot se utilizó una tarjeta de desarrollo DE0-CV con FPGA de la familia Cyclone V de Altera, cuyo firmware necesario se desarrolló en el software Quartus II mediante lenguaje AHDL. Este firmware cuenta con protocolo de comunicación WIFI para poder tener comunicación entre una computadora y la tarjeta de adquisición de datos. Para realizar el control del robot se utilizó un microprocesador embebido en la FPGA desarrollado en la Maestría en Ciencias de la Electrónica opción en Automatización de la Facultad de Ciencias de la Electrónica de la Benemérita Universidad Autónoma de Puebla [7].

Este microprocesador se diseñó en base a arquitectura MIPS (Microprocessor without Interlocked Pipeline Stages), el cual es un procesador de 32 bits con arquitectura RISC (Reduced Instruction Set Computer) y cuenta con un banco de 32 registros de 32 bits. De igual manera, permite procesar las instrucciones recibidas en lenguaje ensamblador y realizar las correspondientes operaciones aritméticas, con el fin de llevar a cabo el control del robot. Además, a dicho firmware se le agregaron los bloques necesarios para procesar las señales de los encodes, de los ADC (Analog-Digital Converter) y también para generar las señales de control para cada actuador.

Para interpretar las señales A y B de los encoders, se le agregaron bloques para decodificar dichas señales, de igual manera, llevan a cabo el conteo de los pulsos de los encoders para poder determinar la posición actual de los actuadores que representan los 3 gdl y mediante dichos conteos se puede determinar el sentido de giro de cada motor.

Para interpretar y procesar las señales que se obtienen de los ADC y así determinar la temperatura actual de cada dispositivo (extrusor o base), se almacenan los 8 bits en un bus de 32 bits que posteriormente se convierten a punto flotante ya que el microprocesador solo puede leer datos en precisión simple.

Por otra parte, se le agregó el firmware necesario para realizar tanto los controles de posición como los controles de temperatura mediante PWM. En el caso de los 3 gdl, se manda la señal de PWM hacia los puentes H para activar los motores y según el conteo que se esté haciendo será el sentido de giro en cada motor. Para el extrusor de material y la cama caliente, la señal de PWM se manda hacia sus respectivas etapas de potencia y de esta manera, permitir que cada dispositivo empiece a calentarse.

B. Software de control

1) Control de posición.

Para realizar el control de posición de los 3 ejes del robot, se implementó un control cartesiano proporcional-derivativo (PD) más compensación de gravedad, como se muestra en la Fig. 11, ya que es el más conocido y empleado en el área de control automático. En este tipo de control, se tiene que la acción de control proporcional es directamente proporcional al error de posición y en cuanto a la acción de control derivativa, tiene el efecto de amortiguamiento o freno mecánico a través de la inyección de la velocidad articular, el cual tiene la finalidad de mejorar el desempeño del control proporcional [8].

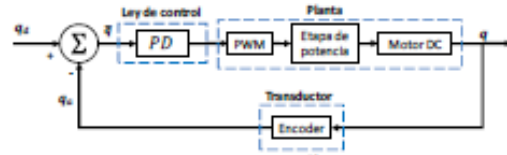


Fig. 11. Lazo cerrado para el control de posición.

Donde:

- q_d es la posición cartesiana deseada (cm).
- q_a es la posición cartesiana actual (cm).
- \tilde{q} es el error de posición cartesiano ($\tilde{q} = q_d - q_a$).

El control cartesiano PD está dado por (6):

$$\tau = K_p \tilde{q} - K_v \dot{q} + g(q) \quad (6)$$

donde,

- K_p es la ganancia proporcional.
- K_v es la ganancia derivativa.
- \dot{q} es la velocidad de movimiento que se emplea para inyección de amortiguamiento.

La ecuación en lazo cerrado formada por el modelo dinámico cartesiano del robot de 3 gdl (5) y el algoritmo de control PD (6) está dada por (7):

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{q}} \\ M^{-1}(q) [K_p \tilde{q} - K_v \dot{\tilde{q}} - C(q, \dot{q}) \dot{\tilde{q}} - B \dot{\tilde{q}}] \end{bmatrix} \quad (7)$$

2) Control de temperatura.

Para el control de temperatura, tanto del extrusor de material como de la cama caliente, se implementó un algoritmo de control tangente hiperbólico (*tanh*), ya que además de tener

sustento científico, una de las cualidades de este esquema de control es que permite operar al actuador (base o extrusor) en la parte lineal lejos de los límites de saturación.

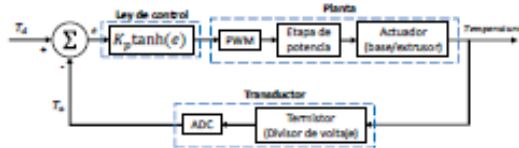


Fig. 12. Lazo cerrado para el control de temperatura.

Donde:

- T_d es la temperatura deseada ($^{\circ}\text{C}$).
- T_a es la temperatura actual ($^{\circ}\text{C}$).
- e es el error de temperatura $e = T_d - T_a$.

Su algoritmo de control está dado por (8).

$$\tau = K_p \tanh(e) \quad (8)$$

donde $K_p > 0$ y la función $\tanh(e) = 0$ cuando $e = 0$.

C. Software de usuario

1) *Compilador de posiciones deseadas.*

El software necesario para que el robot cartesiano pueda realizar el proceso de impresión en 3D se describe en el diagrama de flujo de la Fig. 13.



Fig. 13. Diagrama del software para impresión 3D.

El diseño que se desee imprimir en 3D se descarga de internet o se realiza mediante diseño asistido por computadora (CAD) y el archivo se guarda bajo la extensión *.stl*. Después, mediante un programa laminador, a partir del archivo *.stl* se genera su código G el cual será introducido al compilador de posiciones deseadas (Fig. 14). Mediante la interfaz gráfica de LabView se diseñó dicho compilador, en el cual se carga el archivo en código G "A" y cuyo código se puede visualizar en "B". Después, hace una depuración para extraer las coordenadas de cada eje "C" y las va almacenando en vectores "D". Finalmente, estos vectores se mandarían a RAM para llevar a cabo la acción de impresión en 3D.

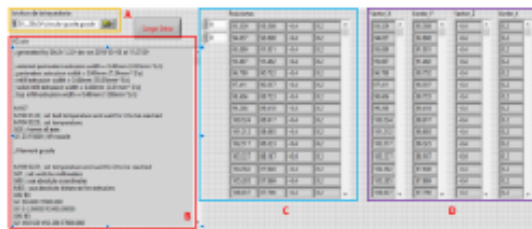


Fig. 14. Interfaz gráfica del compilador de posiciones deseadas.

2) *Interfaz de usuario.*

Para llevar a cabo el control del robot, es decir, el control de posición de sus 3 gdl y los controles de temperatura del extrusor y la cama caliente, se utilizó una interfaz gráfica (Fig. 15) desarrollada en LabView [7], la cual mediante protocolo de comunicación wifi permite la comunicación entre la computadora y el microprocesador "A" embebido en la tarjeta de adquisición (FPGA), además trabaja a una velocidad de muestreo de 2.5ms.

En esta interfaz se carga un archivo que contiene el código "B" mediante el cual se realizará el control de todo el sistema, es decir, donde se cierran todos los lazos de control. Este código está basado en instrucciones en ensamblador para que el microprocesador las pueda interpretar y realizar las correspondientes operaciones aritméticas. En "C" se asignan las posiciones deseadas para cada gdl con sus respectivas ganancias proporcional y derivativa, así como también en "D" se asignan las temperaturas deseadas para el extrusor de material y la cama caliente con sus ganancias proporcionales. Finalmente, en "E" se ejecuta o se detiene el código de control.



Fig. 15. Interfaz de usuario para el control del robot.

VII. RESULTADOS

En la Fig. 16 se presenta el robot cartesiano de 3 gdl que funcionará como impresora 3D, en el cual se muestra la estructura mecánica y la electrónica de potencia.

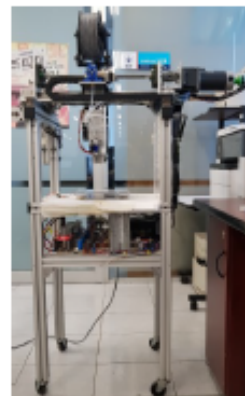


Fig. 16. Sistema mecánico para impresión en 3D.

Ya que el propósito de este robot es funcionar como impresora 3D, el método de impresión que llevará a cabo este robot será del tipo modelado por deposición fundida (FDM), el cual consiste en extruir en caliente un filamento de plástico, que para este caso se utilizará PLA. Este termoplástico presenta un rango de temperatura de impresión de 190-210 °C y para el caso de la cama caliente, se necesita que la cama se mantenga a temperaturas bajas, entre 30-50 °C.

A partir de estos rangos de temperatura, es decir, de 190-210 °C para el extrusor y de 30-50 °C para la cama caliente, se presentan las gráficas obtenidas al ejecutar los controles de temperatura para cada dispositivo. En la Fig. 17 se presenta la gráfica del comportamiento del extrusor al asignarle una temperatura deseada de 200 °C con una ganancia proporcional de 190.

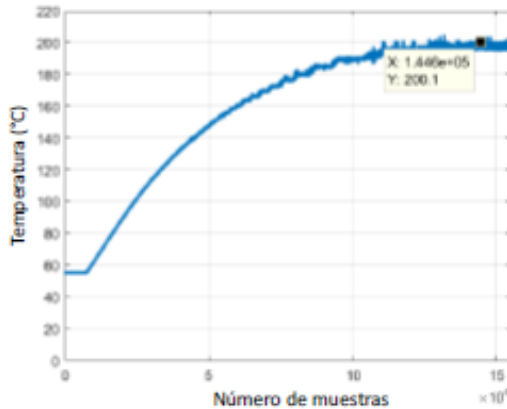


Fig. 17. Control de temperatura del extrusor de material a 200 °C.

En la Fig. 18 se muestra el comportamiento de la cama caliente al asignarle una temperatura deseada de 50 °C con una ganancia proporcional de 180.

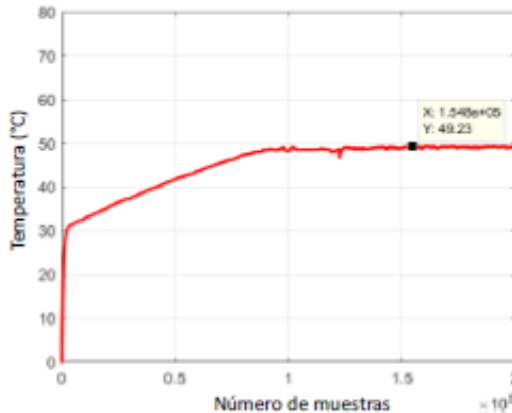


Fig. 18. Control de temperatura de la cama caliente a 50 °C.

CONCLUSIONES

Realizar el control de temperatura, tanto para el extrusor de material como para la cama caliente, fue todo un reto ya que se trata de control no lineal, pero no porque la respuesta del control sea no lineal sino porque la naturaleza de los termistores es no lineal gracias a la respuesta logarítmica que presentan. Debido a esto, se tuvieron que hacer muchas pruebas y ajustes en el control de temperatura de cada planta, sobre todo para el control del extrusor, ya que el termistor presenta una respuesta en la que a mayor temperatura se tiene menor resolución, es por eso que para controlar a 200 °C se tuvo que amplificar el voltaje de salida para que el ADC pudiera leerlo con mayor facilidad, aún así se logró mantener ambos controles en el rango de temperatura que se necesita para cada uno (30-50 °C para la cama caliente, 190-210 °C para el extrusor de material).

La implementación de los circuitos de potencia, son de gran importancia ya que permiten drenar la corriente necesaria según la aplicación solicitada, en este caso, la etapa de potencia que se implementó para cada planta (base y extrusor) funciona de manera adecuada para el control de temperatura.

En cuanto al algoritmo de control empleado (*tanh*), al poner en funcionamiento cada sistema (extrusor o base), se logró mantener controlada la planta a la temperatura deseada teniendo un rango de error de ± 1 °C, lo cual no afecta al sistema ya que se encuentra dentro de los rangos de temperatura deseados.

Como trabajo a futuro se van a hacer pruebas para empezar a extruir el filamento y de esta manera poder concluir con la etapa de impresión en 3D.

REFERENCIAS

- [1] Motor DC [online]. Disponible en: <https://longkang-motor.en.made-in-china.com/product/lozPcJbOKYg/China-24V-DC-Gear-Motor-Z3D120.html>
- [2] Extrusor Metal J-Head long distance [online]. Disponible en: http://www.geeotech.com/wiki/index.php/Metal_J-head_long-distance
- [3] Cama caliente PCB Heatbed [online]. Disponible en: http://reprap.org/wiki/PCB_Heatbed
- [4] A. López, S. Vergara, M. A. Vargas, F. Reyes, A. Palomino, "Design and construction of an acquisition system to control a Cartesian robot as a 3D printer," CIINDET, México, 2016.
- [5] Subir Kumar Saha., "Introducción a la Robotica," Mc Graw Hill, 1er ed., México, 2010, pp.100-106.
- [6] ADC0804 datasheet [online]. Disponible en: <https://www.intersil.com/content/dam/intersil/documents/adc0/adc0803-04.pdf>.
- [7] G. Zepeda, S. Vergara, M. A. Vargas, F. Reyes, D. Pinto, D. Vilarillo, "Diseño e implementación de un microprocesador RISC de 32 bits en una tarjeta con FPGA y comunicación inalámbrica," vol. 1, No. 1, Revista de Ciencias de la Ingeniería y Tecnología (RECIT), México, 2017.
- [8] F. R. Cortés, "Robotica, Control de robots manipuladores," Alfaomega Grupo Editor, 1st ed., México, 2011, pp.336-354.

Bibliografía

- [1] F. Reyes, *Robótica. Control de robots manipuladores*. México: Alfaomega, 2011.
- [2] A. Barrientos, *Fundamentos de robótica*. 1996.
- [3] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, *Robotics. Modeling, Planning and Control*. London: Springer, 2009.
- [4] J. J. Craig, *Robótica*. México: Pearson Educación, 2006.
- [5] M. W. Spong, M. Vidyasagar, *Robot Dynamics and Control*. Canadá: John Wiley & Sons, Inc., 1989.
- [6] “Automatización y Tecnología CNC”. [En línea]. Disponible en: <http://innovacion.uas.edu.mx/automatizacion-y-tecnologia-cnc/>. [Accedido: junio 2019].
- [7] “Historia de las impresoras 3D”. [En línea]. Disponible en: <http://impresorad3d.com/impresoras-3d-historia/>. [Accedido: junio 2019].
- [8] “3D natives, el sitio web de la impresión 3D”. [En línea]. Disponible en: <https://www.3dnatives.com/es/3D-compare>. [Accedido: noviembre 2018].
- [9] W. Wolf, *Computers as Components: Principles of Embedded Computer Systems Design*. USA: Morgan Kaufman, 2008.
- [10] R. Dubey, *Introduction to Embedded System Design Using Field Programmable Gate Arrays*. India: Springer, 2009.
- [11] Manual de usuario, “*DE0-CV User Manual*”. [En línea]. Disponible en: https://www2.pcs.usp.br/~labdig/material/DE0_CV_User_Manual.pdf. [Accedido: octubre 2018].
- [12] A. López Beltrán, “Robot CNC para impresora 3D”, trabajo de fin de máster, Benemérita Universidad Autónoma de Puebla, Puebla, 2016.
- [13] G. Zepeda Arce, “Diseño de un microprocesador de arquitectura RISC de 32 bits”, trabajo de fin de máster, Benemérita Universidad Autónoma de Puebla, Puebla, 2017.
- [14] S. Kumar Saha, *Introducción a la Robótica*, México: Mc Graw Hill, 2010.
- [15] R. Kelly, V. Santibáñez, *Control de Movimiento de Robots Manipuladores*, Madrid: Pearson Education, 2003.
- [16] “Riel de alta precisión”. [En línea]. Disponible en: <http://www.preciolandia.com/mx/cnc-riel-guia-lineal-20mm-para-routers-c-1bc6ap-a.html>. [Accedido: marzo 2018].
- [17] “Concepto de pick and place”. [En línea]. Disponible en: <http://www.motoman.lt/es/Solutions/Applications/Pick--place/>. [Accedido: marzo 2018].

- [18] “Información motor”. [En línea]. Disponible en: http://www.crestmi.com.au/DC_precision_gearmotors.php. [Accedido: marzo 2018].
- [19] “Motor Z4D40-12GN-30S con reductor 4GN15K”. [En línea]. Disponible en: <https://es.aliexpress.com/item/32285141026.html>. [Accedido: mayo 2019].
- [20] “Metal J-Head”. [En línea]. Disponible en: http://www.geeetech.com/wiki/index.php/Metal_J-head_long-distance. [Accedido: marzo 2018].
- [21] “1040220 Kysan Electronics”. [En línea]. Disponible en: http://www.kysanelectronics.com/Products/datasheet_display.php?recordID=8072. [Accedido: mayo 2018].
- [22] “PCB Heatbed”. [En línea]. Disponible en: http://reprap.org/wiki/PCB_Heatbed. [Accedido: abril 2018].
- [23] A. P. Malvino, *Principios de Electrónica*. Madrid: McGRAW-HILL, 2000.
- [24] R. F. Coughlin, F. F. Driscoll, *Amplificadores Operacionales y Circuitos Integrados Lineales*. México: Prentice-Hall, 1993.
- [25] Hoja técnica, “*Power Converter 150-kHz 3-A Step-Down Voltage Regulator*”, Texas Instrument, SNVS124D-NOVEMBER 1999.
- [26] Hoja técnica, “*ADC080x 8-Bit, μ P-Compatible, Analog-to-Digital Converters*”, Texas Instrumente, SNOSBI1C-NOVEMBER 2009.
- [27] Hoja técnica, “*RN-XV Data Sheet*”, Roving Networks, RN-XV-DS v0.3 8/18/2011.
- [28] A. Ernst, J. Heidenhain, *Digital Linear and Angular Metrology*. Landsberg: Verlag Moderne Industrie, 1992.
- [29] R. Kelly, V. Santibáñez D. A. Loría (2005) PD Control with Gravity Compensation. In: *Control of Robot Manipulators in Joint Space*. Advanced Textbooks in Control and Signal Processing. Springer, London. https://doi.org/10.1007/1-85233-999-3_9.