

INSTITUTO TECNOLÓGICO SUPERIOR DE GUANAJUATO



“Brazo robótico clasificador de colores RGB controlado con FPGA.”

TESIS

Para obtener el título de
INGENIERO EN MECATRÓNICA

Presenta:

Juan Manuel Dueñez Silvestre

Con la asesoría de:

M.I. Ismael Urbina Salas

Guanajuato, Gto. 2 de Marzo de 2020



Agradecimientos

Agradezco en especial a mi madre que día a día se esforzó por que saliéramos adelante, su ejemplo me llevo a crecer como persona y me enseñó que las cosas se realizan con paciencia y mucho esfuerzo.

Agradezco a todas las personas que han estado en mi vida, su motivación ha sido de mucha ayuda para lograr mis proyectos.

Agradezco a mi asesor M.I. Ismael Urbina Salas que como profesor me transmitió mucho de su conocimiento y ahora me guía en el desarrollo de mi tesis.



Resumen

En este proyecto, se propone el control de un brazo robótico de 5 grados de libertad (GDL, grados de libertad). El diseño será por medio del software AutoCAD e Inventor y el control se realizara con la tarjeta FPGA Nexys 2. El robot será capaz de detectar, identificar y clasificar objetos de tres distintos colores en RGB, los cuales serán el color rojo, verde y azul también conocidos como colores primarios, por medio de los sensores HC-RS04 y TCRT5000 que estarán instalados en una banda transportadora. El robot detectara la posición del objeto a clasificar y enviara una señal cuando el objeto se encuentre dentro de su alcance. El robot utilizara como actuadores 5 servomotores MG995 para sus eslabones y 1 servomotor MG90S para el actuador final o *gripper*, todos los actuadores trabajaran de forma simultánea para realizar movimientos más rápidos y precisos, la clasificación de los colores primarios se realizara por medio del sensor RGB TCS3200 este sensor enviara una señal de onda cuadrada con un ciclo de trabajo del 50%, dependiendo de las características de esta señal el robot determinara el color del objeto que se va a clasificar y posteriormente será trasladado por el robot a su posición correspondiente. La principal característica de todo el sistema es que será completamente automático.

Para la automatización se desarrollaran los algoritmos de control en lenguaje VHDL y se utilizara una tarjeta FPGA para realizar el control de todo el sistema.



Contenido

<i>Agradecimientos</i>	II
<i>Resumen</i>	III
<i>Contenido</i>	IV
<i>Índice de figuras</i>	VII
<i>Índice de tablas</i>	XI
<i>Introducción</i>	XII
<i>Capítulo 1</i>	1
<i>Generalidades del proyecto</i>	1
1.1 <i>Planteamiento del problema</i>	1
1.2 <i>Justificación</i>	2
1.3 <i>Viabilidad</i>	3
1.4 <i>Objetivos</i>	4
1.4.1 <i>Objetivo general</i>	4
1.4.2 <i>Objetivo específicos</i>	4
1.5 <i>Organización de la tesis</i>	5
<i>Capítulo 2</i>	7
<i>Marco teórico</i>	7
2.1 <i>Robot</i>	7
2.1.1 <i>Tipos de robots</i>	8
2.1.2 <i>Robots industriales</i>	8
2.1.3 <i>Tipos de robots manipulables</i>	10
2.1.4 <i>Robot antropomórfico (RRR)</i>	13
2.2 <i>Cinemática del robot</i>	13
2.2.1 <i>Matriz de transformación homogénea</i>	14



2.2.2 Algoritmo de Denavit - Hartenberg.....	17
2.3 Control	19
2.3.1 FPGA.....	19
2.3.2 Diferencia entre FPGA y Procesador.	20
2.3.3 Servomotor	22
2.3.4 Motor a Pasos.....	25
2.3.5 Sensor RGB.....	27
2.3.6 Sensores de presencia	28
2.4 Antecedentes	31
Capítulo 3	33
Metodología.....	33
3.1 Materiales y Métodos.....	33
3.2 Diseño.....	34
3.2.1 Dimensiones	34
3.2.2 AutoCAD.....	39
3.2.3 Inventor.....	41
3.3 Cinemática	43
3.3.1 Cinemática Directa.....	44
3.4 Corte y Ensamble.....	48
3.4.1 Corte.....	48
3.4.2 Ensamble.....	49
3.5 Control	52
3.5.1 Sensor HC-SR04.....	53
3.5.2 Sensor TCRT5000.....	59
3.5.3 Motor a pasos y Driver A4988.....	61
3.5.3.1 Motor a pasos Nema 17	61
3.5.3.2 Driver A4988 y DRV8825	64
3.5.4 Servomotores MG995.....	71
3.5.4.1 Velocidad angular.....	73
3.5.4.2 Control de Servomotor MG995.....	77



3.5.5 Sensor TCS3200	86
3.5.6 Display 7 Segmentos	90
3.5.7 Divisores de frecuencia	94
Capítulo 4	96
Pruebas y Resultados	96
4.1 Pruebas	96
4.2 Resultados	100
Conclusiones y Recomendaciones	105
Conclusiones	105
Recomendaciones	106
Referencias	108
Anexos	110



Índice de figuras

<i>Figura 1. Tipo de articulaciones.</i>	<i>10</i>
<i>Figura 2. Clasificación de los robots manipuladores.</i>	<i>12</i>
<i>Figura 3. Parámetros D-H para un eslabón Giratorio.</i>	<i>19</i>
<i>Figura 4. Composición de un procesador.....</i>	<i>21</i>
<i>Figura 5. Composición de un FPGA.....</i>	<i>22</i>
<i>Figura 6. Componentes de un servomotor.</i>	<i>23</i>
<i>Figura 7. Combinación de colores más comunes para conexión de servomotor.</i>	<i>23</i>
<i>Figura 8. Diagrama de bloques del servomotor.....</i>	<i>24</i>
<i>Figura 9. Anchos de pulso y sus respectivas posiciones.....</i>	<i>25</i>
<i>Figura 10. Constitución interna de un motor a pasos.</i>	<i>26</i>
<i>Figura 11. Fotodiodos del sensor RGB.</i>	<i>28</i>
<i>Figura 12. Funcionamiento de sensor IR.</i>	<i>29</i>
<i>Figura 13. Esquema de funcionamiento sensor ultrasónico.</i>	<i>31</i>
<i>Figura 14. Prototipo del brazo robótico propuesto.....</i>	<i>36</i>
<i>Figura 15. Capacidad de carga de los actuadores.</i>	<i>36</i>
<i>Figura 16. Servomotores acoplados a eslabón.</i>	<i>37</i>
<i>Figura 17. Grafica de la relación kgf/cm.....</i>	<i>38</i>
<i>Figura 18. Capacidad de carga para un torque de 20 kgf/cm.....</i>	<i>39</i>
<i>Figura 19. Diseño de las piezas en AutoCAD.</i>	<i>40</i>
<i>Figura 20. Vista lateral del brazo robótico.</i>	<i>41</i>



<i>Figura 21. Vista de piezas en 3D.</i>	<i>42</i>
<i>Figura 22. Servomotores creados en inventor.....</i>	<i>42</i>
<i>Figura 23. Ensamble general del brazo robótico realizado en Inventor.</i>	<i>43</i>
<i>Figura 24. Asignación de ejes de referencia del brazo robótico.</i>	<i>44</i>
<i>Figura 25. Corte de acrílico de 0.3 cm de espesor.....</i>	<i>48</i>
<i>Figura 26. Corte de acrílico de 0.5 cm de espesor.....</i>	<i>49</i>
<i>Figura 27. Base giratoria construida en solera.....</i>	<i>50</i>
<i>Figura 28. Base giratoria construida en acrílico.</i>	<i>50</i>
<i>Figura 29. Ensamble físico del brazo robótico.....</i>	<i>51</i>
<i>Figura 30. Ensamble general del brazo robótico y la base.....</i>	<i>52</i>
<i>Figura 31. Sensor HC-SR04.</i>	<i>53</i>
<i>Figura 32. Señales generadas por el sensor HC-SR04.....</i>	<i>54</i>
<i>Figura 33. Sensor HC-RS04 mandando y recibiendo ultrasonido.</i>	<i>56</i>
<i>Figura 34. Conversión Decimal a Binario.....</i>	<i>57</i>
<i>Figura 35. Máquina de estados finitos del sensor HC-SR04.</i>	<i>57</i>
<i>Figura 36. Diagrama de bloques para el sensor HC-RS04.</i>	<i>58</i>
<i>Figura 37. Sensor TCRT5000.....</i>	<i>59</i>
<i>Figura 38. Sensor TCRT5000 en operación.....</i>	<i>61</i>
<i>Figura 39. Motor a pasos Nema 17.....</i>	<i>62</i>
<i>Figura 40. Diagrama motor a pasos bipolar.</i>	<i>62</i>
<i>Figura 41. Driver A4988 y DRV8825.....</i>	<i>64</i>
<i>Figura 42. Identificación de pines driver A4988.....</i>	<i>65</i>
<i>Figura 43. Máquina de estados finitos para el control del driver A4988.</i>	<i>69</i>
<i>Figura 44. Diagrama de bloques para el control del driver A4988.....</i>	<i>70</i>



<i>Figura 45. Servomotor de la marca Tower Pro Modelo MG995.</i>	<i>71</i>
<i>Figura 46. Componentes internos de servomotor.</i>	<i>73</i>
<i>Figura 47. PWM y sus respectivas posiciones angulares para el servomotor.</i>	<i>74</i>
<i>Figura 48. Multiplexor diseñado para controlar la velocidad del servomotor.</i>	<i>76</i>
<i>Figura 49. Diagrama de bloques para la velocidad de los servomotores.....</i>	<i>77</i>
<i>Figura 50. Diagrama para el control de Servomotores.</i>	<i>78</i>
<i>Figura 51. Diagrama de bloques para el control de servomotores.</i>	<i>81</i>
<i>Figura 52. Identificación de Servomotores.</i>	<i>83</i>
<i>Figura 53. Brazo robótico posicionado en Home.....</i>	<i>85</i>
<i>Figura 54. Nueva secuencia de movimiento con respecto de la señal M</i>	<i>86</i>
<i>Figura 55. Sensor TCS3200.</i>	<i>87</i>
<i>Figura 56. Diagrama general sensor TCS3200.....</i>	<i>87</i>
<i>Figura 57. Diagrama de bloques para el control del sensor RGB.....</i>	<i>89</i>
<i>Figura 58. Display 7 Segmentos de cuatro dígitos.</i>	<i>90</i>
<i>Figura 59. Display de 7 segmentos Nexys 2.</i>	<i>91</i>
<i>Figura 60. Diagrama de sincronización para display de 7 segmentos.....</i>	<i>91</i>
<i>Figura 61. Diagrama de bloques para la programación del display de 7 segmentos.....</i>	<i>93</i>
<i>Figura 62. Diagrama de bloques para control de divisores de frecuencia.</i>	<i>95</i>
<i>Figura 63. Brazo robótico ubicado en posición aleatoria.....</i>	<i>96</i>
<i>Figura 64. Brazo robótico en su posición de home.</i>	<i>97</i>
<i>Figura 65. Prueba de funcionamiento del sensor RGB.</i>	<i>98</i>
<i>Figura 66. Sensor TCRT5000 desactivado.</i>	<i>99</i>
<i>Figura 67. Sensor TCRT5000 activado.....</i>	<i>99</i>
<i>Figura 68. Brazo robótico realizando escaneo de objeto color rojo.</i>	<i>100</i>



<i>Figura 69. Frecuencia registrada para objeto de color rojo.</i>	<i>101</i>
<i>Figura 70. Brazo robótico realizando escaneo de objeto color Verde.....</i>	<i>102</i>
<i>Figura 71. Frecuencia registrada para objeto de color verde.</i>	<i>102</i>
<i>Figura 72. Brazo robótico realizando escaneo de objeto color Azul.</i>	<i>103</i>
<i>Figura 73. Frecuencia registrada para objeto de color azul.....</i>	<i>103</i>
<i>Figura 74. Resumen de recursos utilizados de tarjeta FPGA Nexys 2.</i>	<i>105</i>



Índice de tablas

<i>Tabla 1. Clasificación de robots manipuladores.</i>	11
<i>Tabla 2. Especificaciones del sensor RGB.</i>	27
<i>Tabla 3. Parámetros D-H para el robot antropomórfico propuesto</i>	45
<i>Tabla 4. Descripción de los pines del driver A4988.</i>	66
<i>Tabla 5. Conexión para resolución de pasos.</i>	66
<i>Tabla 6. Características Servomotor MG995.</i>	72
<i>Tabla 7. Relación para la velocidad, frecuencia y su valor en sistema binario.</i>	75
<i>Tabla 8. Asignación numérica para cada velocidad.</i>	76
<i>Tabla 9. Definición de señales para el programa Ctrl_A.</i>	79
<i>Tabla 10. Definición de señales para el programa Ctrl_B.</i>	80
<i>Tabla 11. Relación entre la frecuencia y el ángulo de movimiento.</i>	82
<i>Tabla 12. Función de la señal T y señal M.</i>	85
<i>Tabla 13. Configuración para la escala de la frecuencia.</i>	88
<i>Tabla 14. Configuración para la selección de filtro.</i>	88
<i>Tabla 15. Configuración para la activación de los ánodos.</i>	92
<i>Tabla 16. Configuración de cátodos para numeración decimal.</i>	92



Introducción

En la actualidad los avances tecnológicos enfocados en la robótica han crecido a pasos agigantados, hoy en día es normal ver y realizar proyectos de control con solo algunas líneas de programación, incluso sin tener amplios conocimientos en electrónica o programación se pueden desarrollar sistemas completamente autónomos, cosa que hace apenas algunas décadas solo se podía realizar con complejos circuitos electrónicos y grandes algoritmos de programación. Sin embargo, algunos de estos dispositivos de control solo se pueden utilizar en aplicaciones educativas, por su limitada capacidad para realizar procesos de control completos, hablando en términos educativos son una buena opción para introducirse al mundo del control y la programación.

Existen otras alternativas para control que si bien son utilizadas en aplicaciones educativas también lo son en aplicaciones industriales, por ejemplo, los PLC (*Programmable Logic Controller*, de sus siglas en inglés), FPGA (*Field Programmable Gate Array*, de sus siglas en inglés), LabVIEW (acrónimo de *Laboratory Virtual Instrument Engineering Workbench*) entre otros. Pero cual es mejor de ellos o cual presenta ventaja ante los demás, esta pregunta tal vez no tenga una respuesta precisa ya que cada uno puede tener una aplicación, en la cual su desempeño será mejor que el de los demás. Pero se tiene que reconocer que industrialmente los PLC tienen muchas más aplicaciones que otros dispositivos de control, esto debido a que en la industria es necesario estandarizar los procesos de producción y los PLC han sido desde su invención implementados para realizar estas tareas.

Como se mencionó anteriormente un dispositivo de control que actualmente ha ganado terreno en el control industrial son los FPGA, estos dispositivos son



utilizados a menudo conjuntamente con otros sistemas de control, para aumentar su capacidad y su rendimiento en los procesos de control.

En aplicaciones educativas los FPGA se pueden utilizar para el control de sistemas de modelismo, es decir, para recibir y enviar señales de sensores y actuadores a bajo costo, que en conjunto con un adecuado hardware de control se pueden obtener procesos de control complejos y sumamente aplicables industrialmente. Sin embargo, para llegar a realizar esto se deben de hacer los cambios adecuados en sensores y actuadores de mayor capacidad y residencia.

En el presente trabajo se tratará de dar difusión a los sistemas de control con FPGA y demostrar que son una buena alternativa para desarrollar sistemas completos y sumamente competitivos, que puedan sustituir a dispositivos ya existentes en algunas aplicaciones.



Capítulo 1

Generalidades del proyecto

1.1 Planteamiento del problema

Los sistemas de control tienen un amplio campo de aplicaciones y cada día se extienden abarcando más áreas donde es requerido un sistema que minimice los tiempos y costos de algún proceso. Por medio de la investigación surgen nuevas tecnologías que son capaces de desarrollar sistemas más complejos utilizando menos recursos.

En su artículo (Luis & Morales, 2017) [9] describen la importancia de la investigación en la robótica, y sus efectos positivos que tendrá en los campos de la salud, la seguridad y la educación, en esta última área es donde se enfoca este proyecto.

Para el correcto funcionamiento del sistema del brazo robótico se requería un dispositivo capaz de controlar todos los sensores y actuadores del sistema de forma eficiente. Por ello se determinó trabajar con una tarjeta FPGA, ya que una de sus principales características es su alta velocidad de procesamiento y la capacidad de realizar múltiples funciones simultáneamente, los sistemas de control basados en FPGA están ganando terreno en distintas campos de la investigación, la industria y la educación, como se mencionó anteriormente su amplio campo de aplicación es el principal motivo por el cual se utilizara esta tarjeta con la finalidad de dar una amplia difusión a las tecnologías basadas en FPGA.



Se requería un algoritmo capaz de controlar en su totalidad el sistema del brazo robótico, pero dada la limitada información que se pudo recabar en distintas fuentes de consulta sobre el uso de sensores y actuadores con FPGA, se tendrá que hacer una investigación más afondo sobre el funcionamiento de estos dispositivos implementados en otros controladores, para posteriormente realizar los algoritmos en VHDL (acrónimo proveniente de la combinación de dos acrónimos, VHSIC (*Very High Speed Integrated Circuit* y HDL *Hardware Description Language*) y comprobar su funcionamiento.

1.2 Justificación

Hoy en día existen un sinnúmero de dispositivos capaces de realizar el control de sistemas digitales, sin embargo, la mayoría de estos dispositivos tienen limitada su capacidad cuando son utilizados en sistemas de control complejos, partiendo de este punto y conociendo la capacidad de un FPGA, puede procesar gran cantidad de señales simultáneamente a gran velocidad característica que lo posicionan como una de las mejores opciones para el control digital.

La presente tesis tiene como finalidad la investigación e implementación de sistemas de control en FPGA. Dada la poca información que se tiene en libros o incluso en la web acerca del control por medio de esta tarjeta aplicado a actuadores y sensores de bajo costo que son normalmente utilizados por estudiantes que se inician en la electrónica digital, el presente trabajo será una fuente de consulta enfocada para todos aquellos que pretendan conocer el funcionamiento de estos dispositivos, a su vez se pretende dar difusión de las tarjetas FPGA y su amplio campo de aplicación.

Más específicamente aplicado a la carrera de ingeniería en mecatrónica en el Instituto Tecnológico Superior de Guanajuato (ITESG) los estudiantes podrán tener la



presente tesis como una fuente de consulta en distintas materias, por ejemplo en la materia de control digital el estudiante podrá entender el funcionamiento de sensores y actuadores utilizados en el brazo robótico que será descrito a detalle en los siguientes capítulos, también el estudiante podrá consultar la presente tesis en la materia de control para entender los parámetros de Denavit – Hartenberg para robot antropomórficos, en fin se pretende que este trabajo pueda ser una útil fuente de consulta en futuros proyectos desarrollados en el ITESG.

1.3 Viabilidad

El presente trabajo cuenta con varias ventajas con respecto a la viabilidad, y se describen en tres diferentes tipos:

- **Viabilidad técnica:** Hoy en día existen un sinnúmero de sensores y actuadores que pueden ser empleados en sistemas de control como el expuesto en esta tesis, cada uno con características propias, por ejemplo si se realiza una investigación con respecto a actuadores se podrá encontrar un sinnúmero de dispositivos con distintas características en capacidad, tamaño y forma sin embargo la mayoría de ellos poseen el mismo funcionamiento y son compatibles con el control por medio de PWM, de igual forma en el mercado hay varias opciones de tarjetas FPGA cada una con capacidades que se adecuan al tipo de aplicación en que se vayan a emplear.
- **Viabilidad económica:** Con respecto al costo los materiales, sensores y actuadores empleados en este trabajo son económicos y de fácil adquisición de igual forma como se mencionó anteriormente existen gran variedad de estos dispositivos que cumplen con las especificaciones requeridas para ser empleados en este sistema de control y su costo es prácticamente igual a los utilizados y descritos en los capítulos siguientes.



- **Viabilidad temporal:** La construcción de sistemas de control como el expuesto en el presente trabajo, se pueden realizar en un lapso de tiempo corto ya que no presentan gran dificultad en el diseño o control, para que el proyecto pudiera ser viable en este punto se utilizaron los software y herramientas adecuados para el diseño y control. De la misma forma se realizó una investigación completa sobre todos los componentes para minimizar el tiempo de construcción.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar, construir y controlar un brazo robótico de 5 grados de libertad que sea capaz de detectar, identificar y clasificar objetos a partir de los colores primarios rojo, verde y azul.

1.4.2 Objetivo específicos

Realizar por medio de software CAD (del inglés, Computer-Aided Design) el diseño de las piezas que conformaran el brazo robótico.

Simular en Software CAD 3D el ensamble general de las piezas del brazo robótico.

Construir la estructura del brazo robótico e instalar cada uno actuadores y sensores que se necesitan para su correcto funcionamiento.



Realizar una investigación sobre el funcionamiento y la implementación en VHDL de los sensores y actuadores utilizados en todo el sistema.

Elaborar un algoritmo en VHDL que realice el control de los sensores y actuadores que realizarán la detección, identificación y clasificación de los objetos de color rojo, verde y azul.

1.5 Organización de la tesis

El presente trabajo de tesis se reparte en 5 capítulos que a continuación se describen:

En el primer capítulo se agrega una introducción general sobre los temas que se abordarán en el transcurso del presente trabajo. En la descripción del problema se explica el por qué se optó por utilizar una tarjeta FPGA ante otros dispositivos de control, además se agregan trabajos de investigación relacionados con sistemas robóticos y RGB implementados en FPGA. Igualmente se describirán los objetivos a cumplir y la metodología que se deberá seguir.

En el capítulo dos se describirá toda la información que sustentará la tesis, es decir, todos los conceptos, definiciones, y sustento matemático que se deberán seguir. Se describirá el funcionamiento de sensores y actuadores utilizados, también dentro de este capítulo se realizará una breve explicación de lo que son los FPGA y su funcionamiento.

En el capítulo tres se realizará la explicación a detalle de todo el desarrollo del trabajo en base a los conceptos descritos en el capítulo dos. Se agregará el procedimiento que se realizó para el diseño y el ensamble del mecanismo.

En el capítulo cuatro se incluirán todas las pruebas y los resultados que se obtuvieron para determinar el correcto funcionamiento del sistema.



En el capítulo cinco se incluirán las conclusiones a las que se llegaron y las recomendaciones a seguir que pueden ayudar a mejorar el trabajo.

Finalmente, se incluirán las referencias de las fuentes de consulta que se utilizaron y por último se agregaran los anexos donde estará el código en VHDL de todo el sistema, así como los planos de diseño.



Capítulo 2

Marco teórico

2.1 Robot

De acuerdo a (Reyes Cortés, 2011) [14] existen varias definiciones para describir a un robot, entre ellas la que proporciona una mejor descripción es la adoptada por el *Robot Institute of America (RIA)* la cual establece:

- Un robot es un manipulador multifuncional reprogramable diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos programados para la ejecución de una variedad de tareas.

Esta definición que data desde 1980, refleja el estado actual de la tecnología en robótica.

La RAE (Real Academia de la Lengua) tiene dos definiciones para robot:

- Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas.
- Programa que explora automáticamente la red para encontrar información.

Los robots han tenido gran aceptación en la industria, al grado de que se han convertido en elemento clave del proceso de automatización industrial, debido a los beneficios que han traído consigo, tales como reducción de costos, incrementos de la productividad, mejoramiento de la calidad del producto y reducción de problemas en ambientes peligrosos al ser humano.



Entre las principales áreas que se cultivan en robótica se encuentran: control de movimiento, diseño de ambientes de programación, inteligencia artificial, sensores, visión, planificación y coordinación, entre otras. Los robots manipuladores representan nuevas fronteras para el desarrollo y bienestar de la sociedad, son piezas claves de la modernización tecnológica y han representado un factor clave de la economía mundial.

2.1.1 Tipos de robots

Actualmente existe una gran variedad de robots con diversas estructuras geométricas y mecánicas que definen su funcionalidad y aplicación. Sin embargo, de manera general pueden ser clasificados como se muestra en la Tabla 1.

Tabla 1. Clasificación de robots.

Clasificación de Robots		
Móvil	Terrestres: ruedas, patas	
	Submarinos, aéreo - espaciales	
Humanoides	Diseño complejo	
Industriales	Brazos mecánicos	Robots manipulables

Este trabajo se enfocará en los robots industriales específicamente en los robots manipulables, pasando los sistemas móviles y humanoides a segundo término.

2.1.2 Robots industriales

Oficialmente la *ISO* (del inglés, *International Standardization Organization*) define al robot industrial como un manipulador multipropósitos, reprogramable y controlado automáticamente en tres o más ejes.



Los robots industriales son el tipo de robots más populares, debido a la importancia que ocupan en el sector industrial como herramientas clave para la modernización de las empresas. Hoy en día, la automatización de procesos industriales es realizada por medio de robots y esto trae como consecuencia competitividad, productividad, eficiencia y rentabilidad de empresas.

Los robots industriales son conocidos como brazo robots o brazos mecánicos, por analogía con el brazo humano, y se componen de la base la cual puede rotar 360° o 180° alrededor de su eje de giro, además que poseen articulaciones para hombro y codo. Las principales aplicaciones que tienen los robots industriales son procesos de pintado de carrocerías automáticas, accesorios, cubetas, tinas, cajas, soldadura de punto y por arco en carrocerías automotrices, puertas y diversas piezas industriales; transado de herramientas, estibado y empaquetado de materiales, etc.

En aplicaciones más específicas por ejemplo pueden localizar objetos en el espacio de trabajo

En forma general, un robot industrial está formado por los siguientes elementos:

- **Articulación:** Formadas en la mayoría de las veces por servomotores que permiten la conexión y movimiento relativo entre dos eslabones consecutivos del robot. Dependiendo del tipo de movimiento que produzcan las articulaciones pueden ser de tipo rotacional o lineal también conocidas como prismáticas.
- **Actuadores:** Suministran las señales necesarias a las articulaciones para producir movimiento. Los actuadores empleados en robótica pueden ser servomotores, elementos neumáticos, electrónicos o hidráulicos.
- **Sensores:** Proporcionan información del estado interno del robot. Posición y velocidad articular son las variables más comunes en el sistema de sensores.

- **Estructura:** Consiste en una secuencia de eslabones rígidos conectados en cadena abierta por medio de articulaciones; un robot manipulador está caracterizado por un brazo que deberá asegurar la movilidad, una muñeca que confiere la destreza y un extremo final que realiza la tarea programada al robot.
- **Consola de control:** Se compone de un sistema electrónico que incluye los algoritmos de control y la etapa de potencia encargada de suministrar energía al robot para su movimiento.

2.1.3 Tipos de robots manipulables

Los robots manipulables son clasificados dependiendo de los tipos de articulaciones que poseen, en su libro (Ollero Baturone, 2001) [12] menciona que las articulaciones más utilizadas en robótica son las que se indican en la Figura 1.

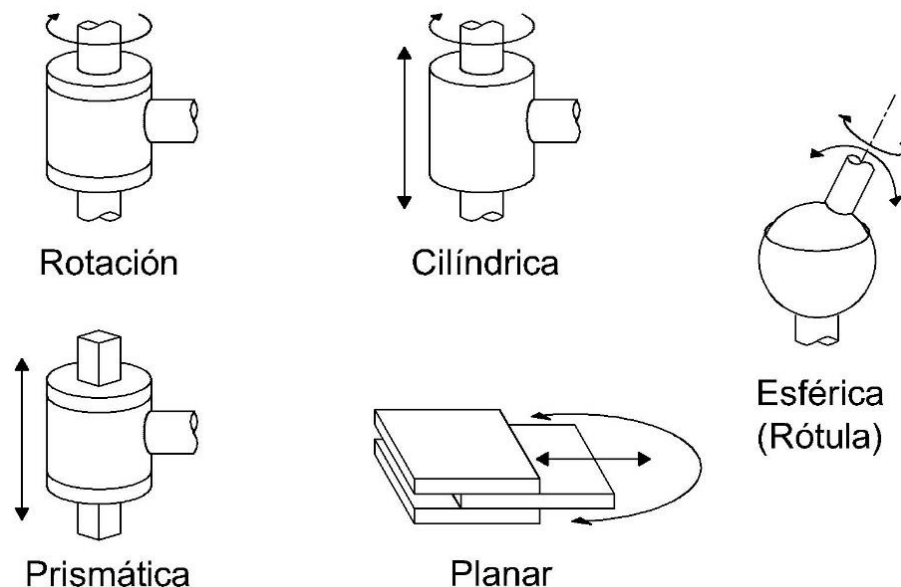


Figura 1. Tipo de articulaciones.



Cada una de las articulaciones se describen de la siguiente forma:

- La articulación de rotación suministra un grado de libertad consiste en una rotación alrededor del eje de la articulación. Esta articulación es, con diferencia, la más empleada.
- En la articulación prismática el grado de libertad consiste en una translación a lo largo del eje de la articulación.
- En la articulación cilíndrica existen dos grados de libertad: uno rotación y una translación.
- La articulación planar está caracterizada por el movimiento de desplazamiento en un plano, existiendo, por tanto, dos grados de libertad.
- Por último, la articulación esférica combina tres giros en tres direcciones perpendiculares en el espacio.

En su libro (Reyes Cortés, 2011) [14] describe que la estructura fundamental de los robots industriales es la cinemática abierta, la cual puede tener diferentes combinaciones de articulaciones rotacionales y prismáticas dando origen a diversos tipos de robots. La clasificación de robots manipuladores se realiza con base en las tres primeras articulaciones del robot, es decir sin incluir la muñeca o pinza colocada en su extremo final, en la Tabla 1 se realiza su clasificación de acuerdo con lo mencionado anteriormente.

Tabla 1. Clasificación de robots manipuladores.

Robot	Características
Antropomórfico (RRR)	Tres articulaciones rotacionales
Scara (RRP)	Dos articulaciones rotacionales y una prismática
Esférico (RRP)	Dos articulaciones rotacionales y una prismática
Cilíndrico (RPP)	Una articulación y dos prismáticas
Cartesiano (PPP)	Tres articulaciones prismáticas

De acuerdo a la tabla anterior se tiene que la letra R significa que es un tipo de articulación rotacional. Mientras que la letra P representa una articulación prismática. El orden como son presentadas corresponden a las articulaciones de la base, hombro y codo.

En la Figura 2 se muestra la clasificación de las 5 configuraciones de robots manipulables.

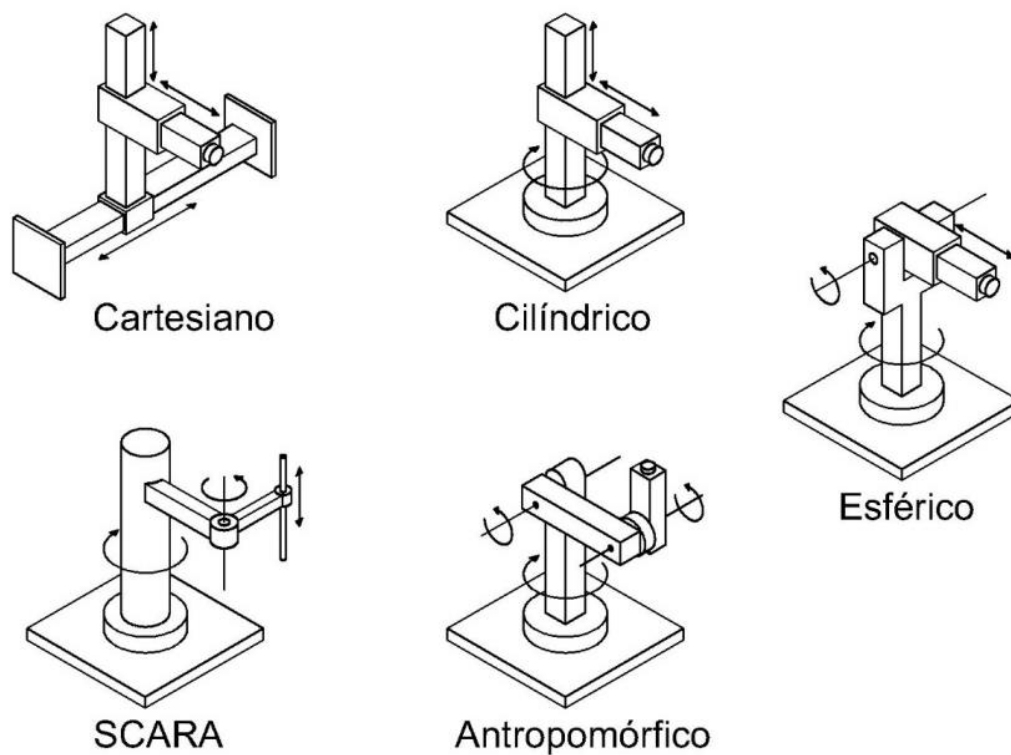


Figura 2. Clasificación de los robots manipuladores.

El robot manipulador utilizado en este proyecto será del tipo antropomórfico (RRR) de cinco grados de libertad (GDL).



2.1.4 Robot antropomórfico (RRR)

Los brazos mecánicos o robots manipuladores que tienen la configuración antropomórfica presentan mayor destreza en su espacio de trabajo, ya que sus eslabones están unidos por tres articulaciones rotacionales. El espacio de trabajo de la configuración antropomórfica corresponde a una esfera hueca, cuyo radio es igual a la suma de la longitud de sus eslabones.

En su libro (Craig, 2006) [4] detalla que como característica los robots antropomórficos minimizan la intrusión de la estructura del manipulador en el espacio de trabajo, haciéndolos capaces de llegar a espacios confinados. Requieren de una estructura total mucho menor que los robots cartesianos, lo que los hace menos costosos para aplicaciones que requieren de espacios de trabajo más pequeños.

2.2 Cinemática del robot

En su libro (Fu, Gonzalez, & Lee, 1988) [5] describen la cinemática del brazo manipulador como el estudio analítico de la geometría del movimiento de un robot con respecto a un sistema de coordenadas de referencia fijo como una función del tiempo sin considerar la fuerzas/momentos que originan dicho movimiento. Así pues, trata con la descripción analítica del desplazamiento espacial del robot como función del tiempo, en particular las relaciones entre las variables espaciales de tipo articulación como la posición y la orientación del efector final del robot. En la cinemática se estudia la posición, la velocidad, la aceleración y todas las demás derivadas de alto orden de las variables de posición (con respecto al tiempo o a cualquier otra variable). En consecuencia, el estudio de la cinemática de manipuladores se refiere a todas las propiedades geométricas y basadas en el tiempo del movimiento. Las relaciones entre estos movimientos y las fuerzas y momentos de torsión que los ocasionan constituyen el problema de la dinámica.



El problema cinemático directo se reduce a encontrar una matriz de transformación que relaciona el sistema de coordenadas ligado al cuerpo al sistema de coordenadas de referencia.

Existen varios métodos para obtener el modelo de cinemática directa, por ejemplo, por geometría aplicada al mecanismo del robot se puede deducir fácilmente las ecuaciones cinemáticas del robot. Sin embargo, cuando el número de grados de libertad crece, este método puede resultar tedioso. Otra alternativa es la metodología Denavit – Hartenberg, la cual es ampliamente conocida en el ambiente de ingeniería y ofrece un procedimiento sencillo para obtener el modelo cinemático directo cuya estructura queda en representación de transformaciones homogéneas.

2.2.1 Matriz de transformación homogénea

De acuerdo a (Fu, Gonzalez, & Lee, 1988) [5] la matriz de transformación homogénea es una matriz de 4×4 que transforma un vector de posición expresado en coordenadas homogéneas desde un sistema de coordenadas hasta otro sistema de coordenadas. Una matriz de transformación homogénea se puede considerar que consiste en cuatro submatrices, como se aprecia en la ecuación 1:

$$T = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ - & - \\ f_{1 \times 3} & 1 \times 1 \end{bmatrix} = \begin{bmatrix} \text{matriz de rotación} & \text{vector de posición} \\ - & - \\ \text{transformación de perspectiva} & \text{escalado} \end{bmatrix} \quad (1)$$

La sub-matriz 3×3 superior izquierda representa la matriz de rotación, la sub-matriz superior derecha 3×1 representa el vector de posición del origen del sistema de coordenadas rotado con respecto al sistema de referencia, la sub-matriz inferior izquierda



1×3 representa la transformación de perspectiva y el cuarto elemento diagonal es el factor de escala global que en aplicaciones en robótica siempre será igual a 1.

Las matrices de rotación y translación homogéneas se pueden multiplicar juntas para obtener una matriz de transformación homogénea compuesta (la llamada matriz T). Sin embargo, como la multiplicación de matrices no es conmutativa, se debe prestar una atención cuidadosa al orden en el cual se multiplican estas matrices.

Un robot de n grados de libertad está formado por n eslabones unidos por n articulaciones, de forma que cada par articulación (cada eslabón constituye un grado de libertad). A cada eslabón se le puede asociar un sistema de referencia solidario a él y utilizando las transformaciones homogéneas, es posible representar las rotaciones y traslaciones relativas entre los distintos eslabones que componen el robot. La matriz de transformación homogénea que representa la posición y orientación relativa entre los distintos sistemas asociados a dos eslabones consecutivos del robot se denomina ${}^{i-1}A_i$. Del mismo modo, la matriz 0A_k , resultante del producto de las matrices ${}^{i-1}A_i$ con i desde 1 hasta k , es la que representa de forma total o parcial la cadena cinemática que forma el robot con respecto al sistema de referencia inercial asociado a la base. Cuando se consideran todos los grados de libertad, a la matriz 0A_n se le denomina como se mencionó anteriormente matriz T , matriz de transformación que relaciona la posición y orientación del extremo final del robot respecto del sistema fijo situado en la base del mismo.

Para describir la relación que existe entre dos sistemas de referencia asociados a eslabones, se utiliza la representación Denavit- Hartenberg (D-H). En su libro (Barrientos, Peñin, Balaguer, & Aracil, 1997) [2] define que Denavit y Hartenberg propusieron en 1955 un método matricial que permite establecer de manera sistemática un sistema de coordenadas $\{S_i\}$ ligado a cada eslabón i de una cadena articulada. Además, la representación D-H permite pasar de un sistema de coordenadas a otro mediante cuatro transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón.



Estas transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que permiten relacionar el sistema de referencia del elemento i con el sistema del elemento $i - 1$. Las transformaciones en cuestión son las siguientes:

1. Rotación alrededor del eje Z_{i-1} un ángulo θ_i .
2. Traslación a lo largo de Z_{i-1} una distancia d_i .
3. Traslación a lo largo de x_i una distancia a_i .
4. Rotación alrededor del eje x_i un ángulo α_i .

Teniendo ya los valores de θ_i , d_i , a_i y α_i que son los denominados parámetros D-H del eslabón i , la matriz de transformación que relaciona los sistemas de referencia $\{S_{i-1}\}$ y $\{S_i\}$ se muestra en la ecuación 2.

$${}^{i-1}A_i = T(Z, \theta_i)T(0,0, d_i)T(a_i, 0,0)T(X, \alpha_i) \quad (2)$$

Desarrollando esta expresión en términos de los parámetros D-H, nos queda la ecuación 3.

$${}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\cos\theta_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\theta_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$



2.2.2 Algoritmo de Denavit - Hartenberg

A continuación se describe el procedimiento explicado en el libro de (Barrientos, Peñin, Balaguer, & Aracil, 1997) [2] para encontrar la cinemática directa a través de la convención Denavit – Hartenberg.

D-H 1 Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.

D-H 2 Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad) y acabando en n .

D-H 3 Localizar el eje de cada articulación. Si ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.

D-H 4 Para i de 0 a $n - 1$ situar el eje Z_i sobre el eje de la articulación $i + 1$.

D-H 5 Situar el origen del sistema de la base $\{S_0\}$ en cualquier punto del eje Z_0 . Los ejes X_0 e Y_0 se situarán de modo que formen un sistema dextrógiro con Z_0 .

D-H 6 Para i de 1 a $n - 1$, situar el sistema $\{S_i\}$ (solidario al eslabón i) en la intersección del eje Z_i con la línea normal común a Z_{i-1} y Z_i . Si ambos ejes se cortasen se situaría $\{S_i\}$ en el punto de corte. Si fuesen paralelos $\{S_i\}$ se situaría en la articulación $i + 1$.



D-H 7 Situar X_i en la línea normal común a Z_{i-1} y Z_i .

D-H 8 Situar Y_i de modo que forme un sistema dextrógiro con X_i y Z_i .

D-H 9 Situar el sistema $\{S_n\}$ en el extremo del robot de modo que Z_n coincida con la dirección de Z_{n-1} y X_n sea normal a Z_{n-1} y Z_n .

D-H 10 Obtener θ_i como el ángulo que hay que girar en torno a Z_{i-1} para que X_{i-1} y X_i queden paralelos.

D-H 11 Obtener d_i como la distancia, medida a lo largo de Z_{i-1} , que habría que desplazar $\{S_{i-1}\}$ para que X_{i-1} y X_i quedasen alineados.

D-H 12 Obtener a_i como la distancia medida a lo largo de X_i (que ahora coincidiría con X_{i-1}) que habría que desplazar el nuevo $\{S_{i-1}\}$ para que su origen coincidiese con $\{S_i\}$.

D-H 13 Obtener α_i como el ángulo que habría que girar en torno a X_i (que ahora coincidiría con X_{i-1}), para que el nuevo $\{S_{i-1}\}$ coincidiese totalmente con $\{S_i\}$.

D-H 14 Obtener las matrices de transformación ${}^{i-1}A_i$ definidas en (7).

D-H 15 Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot $T = {}^0A_1, {}^1A_2 \dots {}^{n-1}A_n$.

D-H 16 La matriz T define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base en función de las n coordenadas articulares.

En la Figura 3 se muestran los parámetros de D-H para un eslabón giratorio.

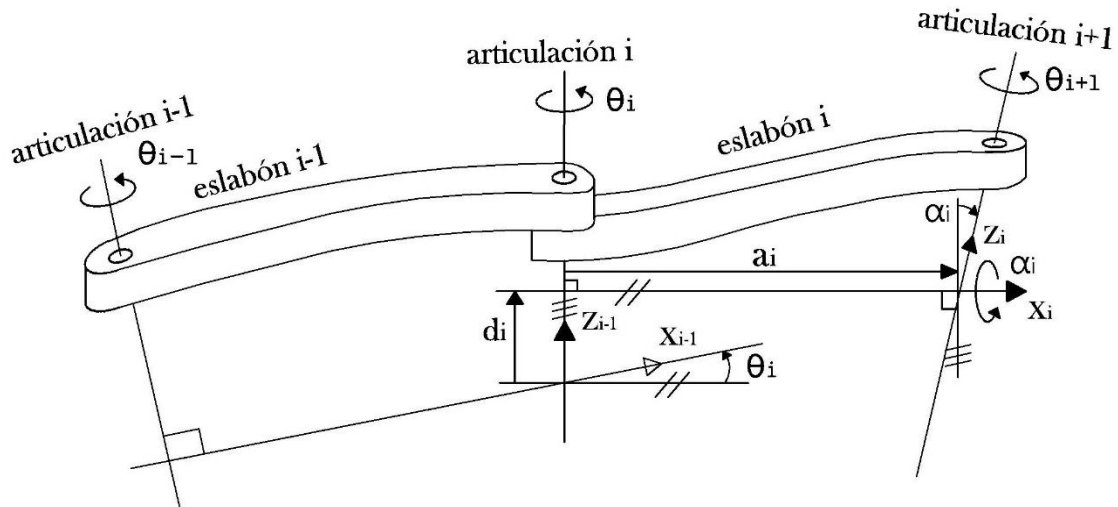


Figura 3. Parámetros D-H para un eslabón Giratorio.

2.3 Control

En esta sección se describe los dispositivos de control más comúnmente empleados en robótica. Del mismo modo se describen los principales dispositivos que articulan a un robot.

2.3.1 FPGA

Un FPGA (del inglés, *Field Programmable Gate Array*) es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada mediante un lenguaje de descripción especializado.



La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas en un chip.

Los FPGAs a diferencia de los microcontroladores y microprocesadores no se programan, se describen, no tienen lenguaje de programación, tienen lenguaje de descripción de hardware. Se llama *descripción* porque lo que en realidad se hace es describir las conexiones entre los elementos. Actualmente, existen dos software para describir: *VHDL* y *Verilog*.

Generalmente si se acude a un diseño con FPGA es porque se necesitan prestaciones críticas, que no es fácil asegurar por otra vía, mucho más simple de un procesador, un microprocesador, un microcontrolador, un DSP, o del tipo que sea. Es por ello que al final en el uso de FPGAs se suelen hacer especificaciones todavía cercanas al mundo del diseño hardware, aunque con la enorme ventaja de que se realiza por lenguaje y volcado al dispositivo, con lo que es posible el rediseño y la corrección de errores por reprogramación.

2.3.2 Diferencia entre FPGA y Procesador.

En su página web (Llamas, Ingeniería, informática y diseño, 2017) [8] describe que un procesador y un FPGA como dispositivos similares, sin embargo al profundizar es casi más fácil encontrar diferencias que similitudes, por ejemplo, un procesador contiene una serie de instrucciones (funciones) que realizan operaciones sobre operadores binarios (sumar, incrementar, leer y escribir de la memoria). Algunos procesadores tienen más instrucciones que otras (asociados a circuitería interna del procesador) y es uno de los factores que determinan su rendimiento, mostrado en la Figura 4.

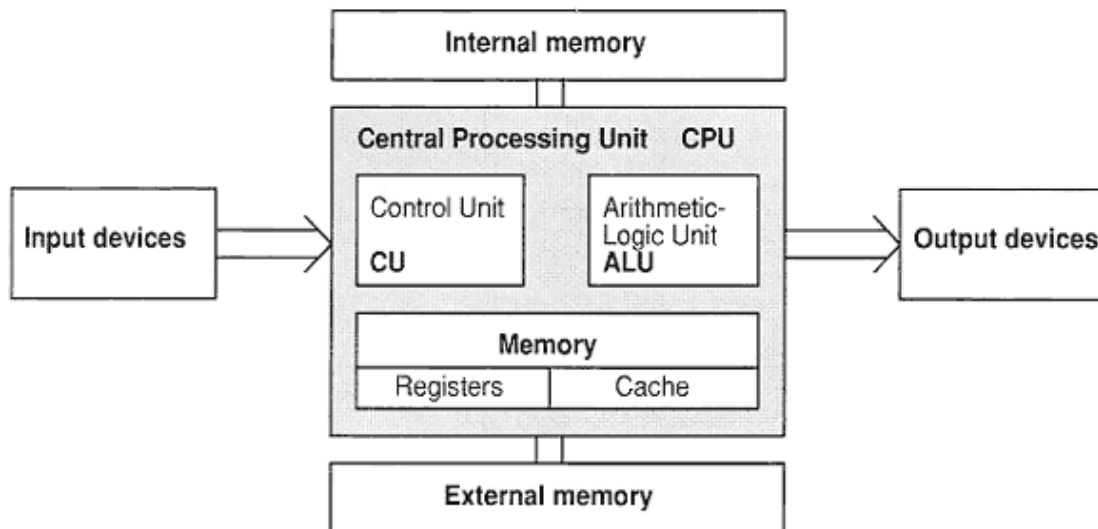


Figura 4. Composición de un procesador.

Por otro lado, contiene una serie de registros, que contienen los datos de entrada y salida en las operaciones del procesador. Además, se dispone de memoria para almacenar información. Finalmente, un procesador contiene una pila de instrucciones que contienen el programa que va a ejecutarse en código máquina, y un reloj, en cada ciclo de reloj, el procesador lee de la pila de instrucciones los valores necesarios, llama a la instrucción oportuna, y ejecuta el cálculo, cuando programamos el procesador empleamos uno de los muchos lenguajes disponibles, en un formato entendible y cómodo. En el proceso de enlazado y compilación, el código se traduce a código máquina, que se graba en la memoria del procesador.

Sin embargo, al programar un FPGA lo que se hace es modificar una matriz de conexiones. Los bloques individuales están constituidos por elementos que les permiten adoptar distintas funciones de transferencia, mostrado en la Figura 5.

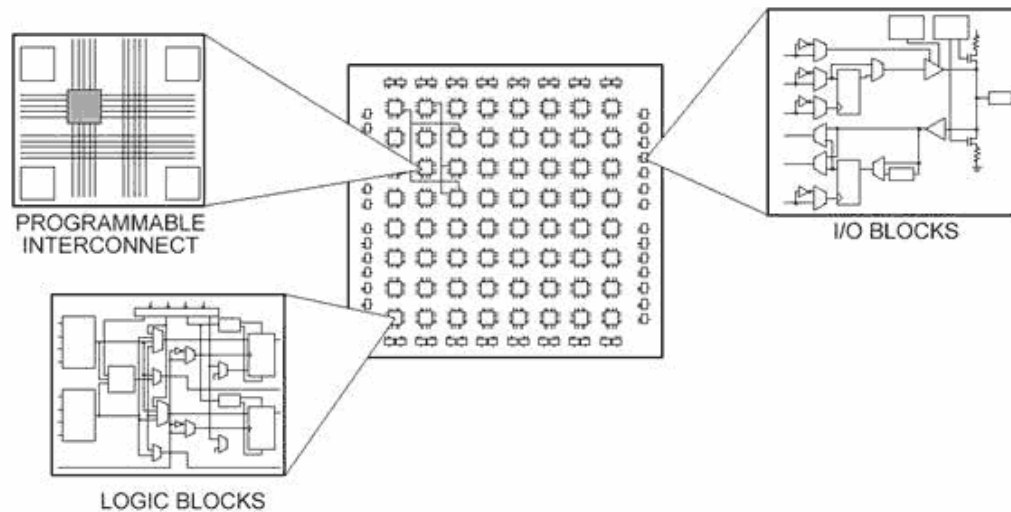


Figura 5. Composición de un FPGA.

Juntos, los distintos bloques, unidos por las conexiones que programan, hacen que físicamente se constituya un circuito electrónico, de forma similar a como se realizaría en una placa de entrenamiento (*protoboard*) o al fabricar un chip propio. Como se observa, la diferencia es sustancial. Un procesador (en sus muchas variantes) tiene una estructura fija y se modifica su comportamiento a través del programa que realizamos, traducido en código máquina, y ejecutado de forma secuencial, sin embargo, en un FPGA se varía la estructura interna, sintetizando uno o varios circuitos electrónicos en su interior. Al describir el FPGA definiremos los circuitos electrónicos que queremos que se configuren en su interior.

2.3.3 Servomotor

Un servomotor es un tipo especial de motor que permite controlar la posición del eje en un momento dado, está diseñado para moverse determinada cantidad de grados y luego mantenerse fijo en una posición. Al hablar de un servomotor se hace referencia a un sistema compuesto por componentes electromecánicos y electrónicos, mostrados en la Figura 6.

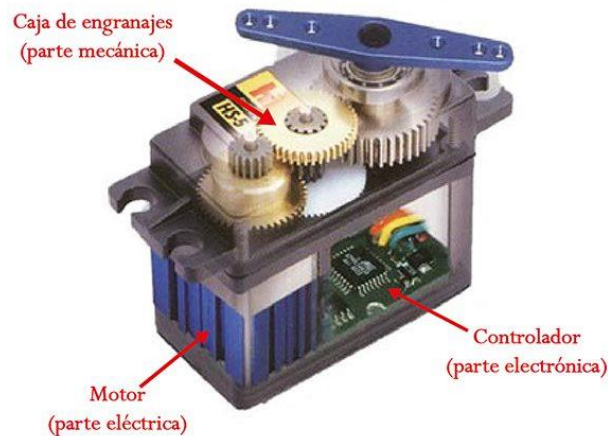


Figura 6. Componentes de un servomotor.

El motor en el interior de un servomotor es un motor DC (corriente directa). El eje del motor se acopla a una caja de engranajes similar a una transmisión. Esto se hace para potenciar el torque del motor y permitir mantener una posición fija cuando se requiera. El circuito electrónico es el encargado de manejar el movimiento y la posición del motor.

Los servomotores poseen tres cables, a diferencia de los motores comunes que sólo tienen dos. Estos tres cables casi siempre tienen los mismos colores, por lo que son fácilmente reconocibles, en la Figura 7 se muestran los colores más comunes para su conexión.

Voltaje positivo	Tierra (ground)	Señal de control

Figura 7. Combinación de colores más comunes para conexión de servomotor.

Los colores con los que contarán el cable para la conexión de los servomotores dependerán del fabricante.

En la Figura 8 se muestra el diagrama de bloques de los componentes interno de un servomotor.

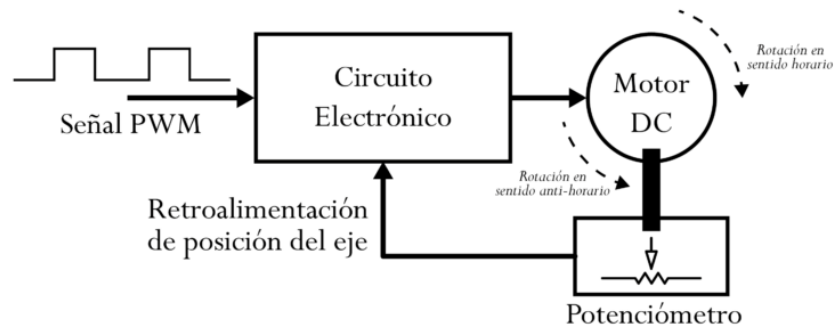


Figura 8. Diagrama de bloques del servomotor.

El diagrama de bloque del servomotor representa de forma visual el servomotor como un sistema. El circuito electrónico es el encargado de recibir la señal PWM (*Pulse-Width Modulation*, por sus siglas en inglés) y traducirla en movimiento del motor DC. El eje del motor DC está acoplado a un potenciómetro, el cual permite formar un divisor de voltaje. El voltaje en la salida del divisor varía en función de la posición del eje del motor DC.

Los servomotores como se menciona anteriormente requieren de una señal de control para realizar los movimientos de su eje, esta señal es normalmente conocida como PWM o ancho de pulso, dependiendo de la duración de este pulso el servomotor se posicionará en un ángulo específico, cuando se realice un cambio en el PWM el servomotor se posicionara en otro ángulo correspondiente al nuevo ancho de pulso.

En la Figura 9 se muestra los anchos de pulso con sus respectivas posiciones para un servomotor.

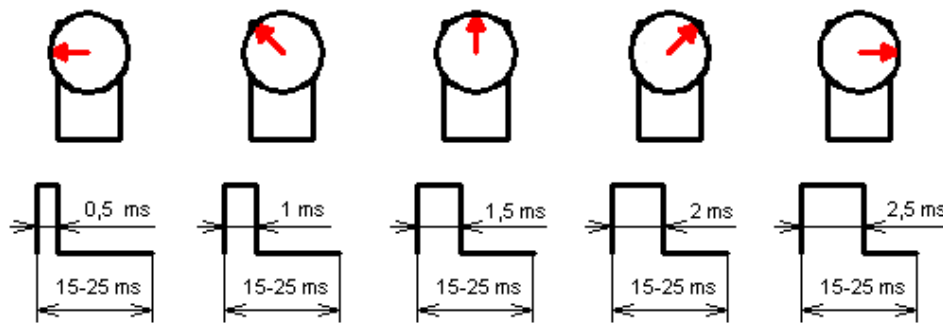


Figura 9. Anchos de pulso y sus respectivas posiciones.

2.3.4 Motor a Pasos

Un motor paso a paso es un dispositivo electromecánico que convierte una serie de pulsos eléctricos en desplazamientos angulares, lo que significa que es capaz de girar una cantidad de grados (paso o medio paso) dependiendo de sus entradas de control.

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos. La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde 90° hasta pequeños movimientos de 1.8° .

Este tipo de motores son muy utilizados, ya que pueden moverse a deseo del usuario según la secuencia que se les indique a través de un microcontrolador.

Estos motores poseen la habilidad de quedar enclavados en una posición si una o más de sus bobinas están energizadas o bien totalmente libres de corriente.

En la Figura 10 se muestra la constitución interna de un motor a pasos.



Figura 10. Constitución interna de un motor a pasos.

El principio de su funcionamiento está basado en un estator construido por varios bobinados en un material ferromagnético y un rotor que puede girar libremente en el estator. Estos diferentes bobinados son alimentados uno a continuación del otro y causan un determinado desplazamiento angular que se denomina “paso angular” y es la principal característica del motor.

Existen dos tipos de motores de imán permanente que son los más utilizados en robótica:

Unipolares: Estos motores suelen tener 5 o 6 cables de salida dependiendo de su conexionado interno. Suelen ser 4 cables por los cuales se recibe los pulsos que indican la secuencia y duración de los pasos y los restantes sirven como alimentación del motor. Este tipo se caracteriza por ser más simple de controlar.

Bipolares: Este tipo de motores por lo general tienen 4 cables de salida, necesitan ciertas manipulaciones para poder ser controlados, debido a que requieren del cambio de dirección de flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento, es necesario un puente H por cada bobina del motor, es decir que para controlar un motor paso a paso de 4 cables (dos bobinas), se necesitan usar dos puentes H. Esto hace que la tarjeta controladora se vuelva más compleja y costosa. Su uso no es tan común como en el caso de los de tipo unipolar.



2.3.5 Sensor RGB

El sensor de color RGB (de sus siglas en inglés *Red, Green y Blue*) es un convertidor de luz a frecuencia que combina fotodiodos de silicio reconfigurables y una corriente de frecuencia en un solo circuito integrado. La salida es una onda cuadrada (ciclo de trabajo 50%) con una frecuencia directamente proporcional a la intensidad de luz. Las entradas y salidas digitales permiten una interfaz directa con un microcontrolador u otro conjunto de circuitos lógicos, por esta razón el sensor de color RGB es ideal para líneas de producción, domótica, robótica, etc.

En la Tabla 2 se muestra las especificaciones del sensor de color RGB utilizados en proyectos robóticos.

Tabla 2. Especificaciones del sensor RGB.

Especificaciones	
1	Alta resolución de conversión de luz a frecuencia.
2	Frecuencia de salida programable en color y escala completa.
3	Se comunica directamente con un microcontrolador.
4	Voltaje de funcionamiento: 2.7-5.5 V.
5	Rango de error típicamente de 0,2% a 50 kHz.

El sensor RGB utilizado en este proyecto combina 64 foto-diodos de silicio configurables y un convertidor de corriente a frecuencia en un solo chip. La salida es una onda cuadrada con una frecuencia directamente proporcional a la intensidad de la luz (irradiación). El convertidor de luz a frecuencia lee una matriz de diodos de 8 X 8 de los cuales 16 tienen filtro azul, 16 filtro verde, 16 filtro rojo y 16 sin filtro.

El funcionamiento del sensor es simple de entender y funciona de la siguiente forma, si se escanea un objeto de color rojo con los tres filtro rojo, verde y azul (los foto-diodos sin filtro no se utilizaran en este proyecto) por un segundo para cada filtro y se realiza una cuenta de los pulsos recibidos por la frecuencia correspondiente a cada filtro, y posteriormente estas cuentas se comparan entre sí, la frecuencia más alta registrada en la operación corresponderá al filtro que tenga el mismo color que el objeto, dicho de otra forma si se analiza un objeto de color azul la frecuencia más alta registrada será correspondiente al filtro azul, si el objeto es color verde entonces el filtro de color verde enviara la frecuencia más alta, lo mismo pasara con el filtro verde respectivamente. Los cuatro tipos de colores están distribuidos uniformemente por toda la matriz para minimizar el efecto de la no-uniformidad, en la Figura 11 se muestra el arreglo de fotodiodos con sus respectivos filtros.

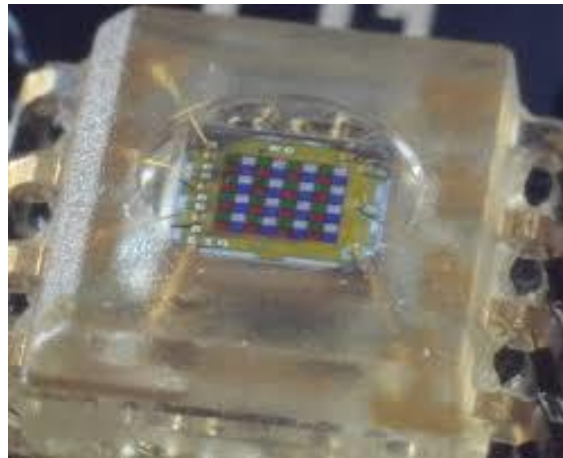


Figura 11. Fotodiodos del sensor RGB.

2.3.6 Sensores de presencia

Los sensores de presencia utilizados en este proyecto son de dos tipos mostrados a continuación:

Sensor IR: Un detector de obstáculos infrarrojo es un dispositivo que detecta la presencia de un objeto mediante la reflexión que produce en la luz. El uso de luz infrarroja (IR) es simplemente para que esta no sea visible para los humanos. Se dispone de un LED emisor de luz infrarroja y de un fotodiodo (tipo BPV10NF o similar) que recibe la luz reflejada por un posible obstáculo. Los detectores de obstáculo suelen proporcionarse con una placa de medición estándar con el comparador LM393, que permite obtener la lectura como un valor digital cuando se supera un cierto umbral, que se regula a través de un potenciómetro ubicado en la placa. Este tipo de sensores actúan a distancias cortas, típicamente de 5 mm a 20 mm. Además, la cantidad de luz infrarroja recibida depende del color, material, forma y posición del obstáculo, por lo que no disponen de una precisión suficiente para proporcionar una estimación de la distancia al obstáculo. Pese a esta limitación son ampliamente utilizados para la detección de obstáculos en pequeños vehículos o robots. Su bajo costo hace que sea frecuente ubicarlos en el perímetro, de forma que detectemos obstáculos en varias direcciones.

En la Figura 12 se muestra el principio básico de su funcionamiento que consta de un led infrarrojo como emisor que será el encargado de enviar la señal luminosa, si esta señal encuentra un objeto en su camino y se encuentra dentro de su alcance se reflejara hacia el foto-diodo receptor y enviara una señal digital al dispositivo de control.

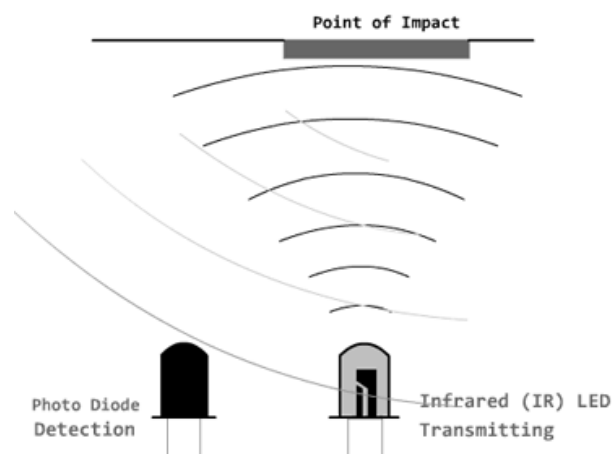


Figura 12. Funcionamiento de sensor IR.



Sensor Ultrasónico: Un sensor de ultra sonidos es un dispositivo para medir distancias. Su funcionamiento se base en el envío de un pulso de alta frecuencia, no audible por el ser humano. Este pulso rebota en los objetos cercanos y es reflejado hacia el sensor, que dispone de un micrófono adecuado para esa frecuencia. Midiendo el tiempo entre pulsos, conociendo la velocidad del sonido, se puede estimar la distancia del objeto contra cuya superficie impacto el impulso de ultrasonidos. Los sensores de ultrasonidos son sensores baratos, y sencillos de usar. El rango de medición teórico del sensor es de 2 cm a 400 cm, con una resolución de 0.3 cm. En la práctica, sin embargo, el rango de medición real es mucho más limitado, en torno a 20 cm a 2 metros. Los sensores de ultrasonidos son sensores de baja precisión. La orientación de la superficie a medir puede provocar que la onda se refleje, falseando la medición. Además, no resultan adecuados en entornos con gran número de objetos, dado que el sonido rebota en las superficies generando ecos y falsas mediciones. Tampoco son apropiados para el funcionamiento en el exterior y al aire libre. Pese a esta baja precisión, que impide conocer con precisión la distancia a un objeto, los sensores de ultrasonidos son ampliamente empleados.

En robótica es habitual montar uno o varios de estos sensores, por ejemplo, para detección de obstáculos, determinar la posición del robot, crear mapas de entorno, o resolver laberintos. En aplicaciones en que se requiera una precisión superior en la medición de la distancia, suelen acompañarse de medidores de distancia por infrarrojos y sensores ópticos.

En la Figura 13 se muestra el esquema de su funcionamiento, estos sensores envían una señal ultrasónica a través de una bocina emisora, esta señal viaja por el aire a una velocidad aproximada de 340 m/s, si la señal ultrasónica encuentra un objeto en su camino esta rebotara en varias direcciones, y parte de esta señal será captada por la bocina receptora, haciendo un cálculo de conversión entre velocidad y tiempo se obtiene la distancia aproximada a la que se encuentra el obstáculo en el que reboto la señal.

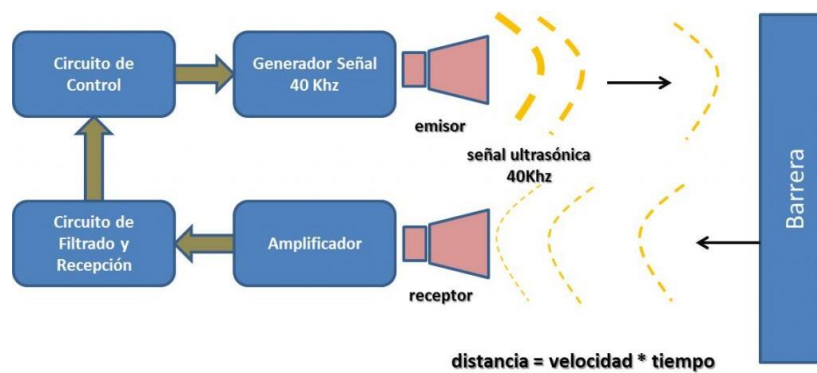


Figura 13. Esquema de funcionamiento sensor ultrasónico.

2.4 Antecedentes

Desde que se inventaron los sistemas FPGAs en 1984 por Xilinx han estado evolucionando hasta convertirse en parte importante de las industrias de medición, aeroespacial, de procesamiento de imágenes, de consumo, de comunicaciones y de procesamiento de señales. Una de las principales razones por la cual los FPGAs se han vuelto tan populares, es la cantidad de instrucciones por segundo que son capaces de ejecutar [6].

Hoy en día existe un sinnúmero de dispositivos capaces de realizar el control de sistemas robóticos, sin embargo, los FPGAs presentan ventaja ante otros sistemas por ejemplo la capacidad de personalizar la arquitectura en función del comportamiento deseado para el sistema, es decir, optimizar los recursos o aumentar el desempeño.

(Jiménez Hernández, 2015) [6] Presenta una tesis utilizando un manipulador de 5 grados de libertad y la implementación de un sistema de visión artificial utilizando una cámara que captura y transmite las imágenes a la tarjeta, como componente central se utiliza la tarjeta de desarrollo Nexys 4 y su programación se realiza en lenguaje descriptor de hardware o HDL (*Hardware Description Language*, de sus siglas en inglés).



(Cipres Torres, 2016) [3] Presenta una tesis para desarrollar un control alterno con tecnología FPGA para el robot *scorbot-er 4u*. Este manipulador cuenta con 5 grados de libertad y como actuadores utiliza motores de corriente directa fijados en la base del robot estos transmiten el movimiento a sus articulaciones por medio de bandas, en este trabajo se demuestra cómo se puede sustituir el control original del robot por un sistema basado en FPGA de menor costo.

(Nhivekar & Mudholkar, 2012) [11] Presenta un artículo sobre un sensor de color de bajo costo creado a partir de una fotorresistencia y un led, utilizando un convertidor analógico – Digital (ADC) para convertir la señal analógica de la fotorresistencia y ser enviada en forma digital al FPGA Cyclone-II, el color identificado se muestra través de un monitor VGA también conectado al FPGA, demostrando el correcto funcionamiento de sensores de bajo costo implementados en sistemas FPGA.

(Andueza & Aguirre, 2009) [1] Presentan un artículo donde se describen los principales aspectos del modelo, diseño y construcción de un manipulador robótico de tres grados de libertad. En este trabajo se calcula la cinemática directa e inversa del manipulador, haciendo uso de los parámetros de Denavit-Hartenberg, y se calcula el modelo dinámico del manipulador. Se realizan pruebas de simulación del comportamiento del manipulador mediante la creación de rutinas en Matlab y se comparan con los resultados obtenidos mediante el uso de la Toolbox de Robótica de Matlab desarrollada por Peter Corke.

(Luis & Morales, 2017) [9] Presentan un artículo llamado "Los Robots del Futuro y su Importancia para México" describen la importancia de la investigación en la robótica, y sus efectos positivos que tendrá en los campos de la salud, la seguridad y la educación en México.



Capítulo 3

Metodología

3.1 Materiales y Métodos

La metodología propuesta para la realización de la presente tesis constara de 4 etapas a seguir.

En la primera etapa, se encuentra el diseño del brazo robótico, la investigación y selección de actuadores y sensores, en base a la investigación se realizara el diseño conforme a las capacidades de trabajo de los actuadores y se aplicaran diversos conocimientos prácticos en software de diseño CAD para la creación de cada pieza que conformara el sistema robótico, cada componente utilizado deberá de ser económico y de fácil adquisición para que el proyecto sea viable.

La segunda etapa, consistirá en la simulación del brazo robótico en software CAD 3D en base a la investigación se determinará el software que mejor se acople a las necesidades de trabajo. El programa de simulación estará destinado a predecir posibles fallas en el diseño o limitaciones en sus movimientos. En base a los resultados obtenidos en la simulación se deberán corregir los errores para que el sistema robótico cumpla con los objetivos establecidos.

La tercera etapa, consistirá en el corte y ensamble físico del brazo robótico. El material en el que se cortara cada pieza deberá ser resistente, de bajo costo y que pueda ser seccionado por medio de una cortadora laser. En el ensamble se acoplará cada pieza cortada y sus respectivos generadores de movimiento para finalizar esta etapa se realizaran pruebas básicas de funcionalidad.



La cuarta etapa, describirá la automatización completa del sistema robótico, iniciando con algoritmos básicos para el control de cada actuador o sensor y así determinar su correcto funcionamiento. Por lo tanto, cada componente del sistema tendrá una programación de control independiente que conformar al algoritmo final, la complejidad de este aumentará una vez se haya complete el objetivo del sistema robótico.

3.2 Diseño

En esta sección se describe la metodología utilizada para el diseño y construcción, así como los software utilizados para la creación y simulación del prototipo propuesto.

3.2.1 Dimensiones

Las principales consideraciones que se tomaran para definir las dimensiones de los eslabones que conformaran el brazo robótico fueron el torque de trabajo de sus actuadores y la distancia total del brazo, los actuadores que se utilizaran son servomotores de la marca TowerPro modelo MG995 que proporcionan un torque de trabajo de 10 kgf/cm cuando son alimentados a 5 volts. Con respecto a la dimensión del mecanismo se utilizó una distancia total entre 50 cm a 55 cm esto se determinó a partir de la distribución de las medidas de cada eslabón y del peso total del mecanismo, siendo de gran importancia la consideración anterior para que el peso total se encuentre dentro de las capacidades de carga de cada actuador.

Los eslabones que conforman el mecanismo serán construidos con acrílico de 0.3 cm de espesor, conforme a sus especificaciones de este material una sección de 10,000 cm² (1 m²) tiene un peso de 3.598 kg, dada esta información se calculara el



área y el peso aproximado del brazo robótico y posteriormente se comparara con el diseño final de cada eslabón, en la Figura 14 se muestra el prototipo con las dimensiones y pesos de cada eslabón y actuador utilizados.

Para obtener el peso del acrílico se suman las áreas de cada eslabón, esto se presenta en la ecuación 4.

$$\Sigma = 432 \text{ cm}^2 + 217 \text{ cm}^2 + 144 \text{ cm}^2 \quad \therefore \quad \Sigma = 793 \text{ cm}^2 \quad (4)$$

Resolviendo la ecuación 5 se obtiene el peso del acrílico utilizado.

$$\begin{aligned} 10,000 \text{ cm}^2 &\rightarrow 3.598 \text{ kg} \\ 793 \text{ cm}^2 &\rightarrow X \quad \therefore \quad X = 0.285 \text{ kg} \end{aligned} \quad (5)$$

Se suma el peso de los tres actuadores para obtener el peso total del brazo como se muestra en la ecuación 6.

$$\begin{aligned} \Sigma &= 0.285 \text{ kg} + 0.062 \text{ kg} + 0.062 \text{ kg} + 0.014 \text{ kg} \\ \Sigma &= 0.423 \text{ kg} \end{aligned} \quad (6)$$

El peso total del brazo será de 0.423 kg, como se muestra en la Figura 14 con esta información se determinara si los actuadores tienen el torque necesario para mover el mecanismo sin problemas.

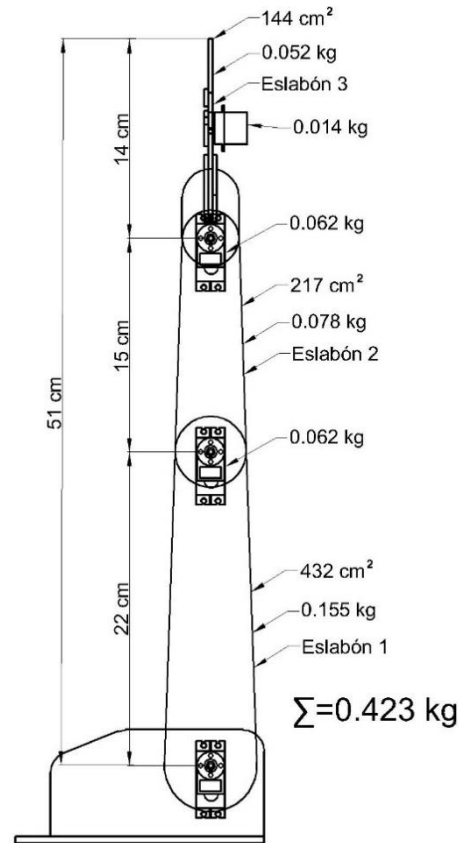


Figura 14. Prototipo del brazo robótico propuesto.

Como se mencionó anteriormente los servomotores utilizados proveen un torque de 10 kgf/cm alimentados a 5 volts, en la Figura 15 se explica a qué se refiere esta unidad de medida.

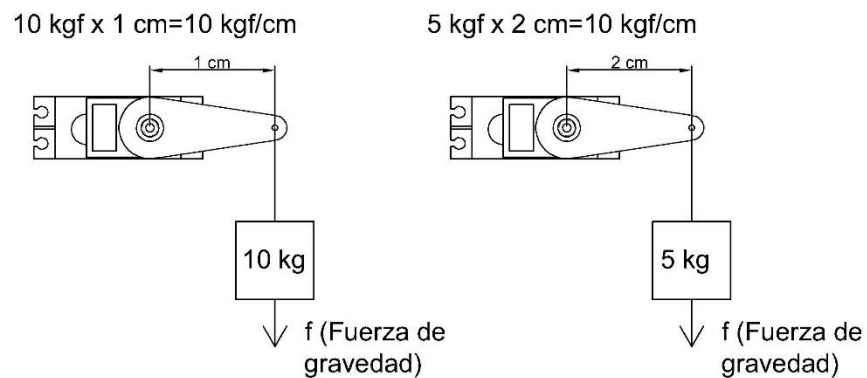


Figura 15. Capacidad de carga de los actuadores.

En la figura anterior se observa que conforme se aleja la carga del eje del motor menor es su fuerza, por ejemplo, si la carga está a un centímetro de distancia del eje del motor se aplicará un torque total de 10 kgf/cm, pero si la carga se desplaza a dos centímetros del eje del motor el torque aplicado será de 5 kgf/cm.

En el prototipo de la Figura 14 se estableció el primer eslabón de 22 cm, este será movido por dos servomotores ambos de 10 kgf/cm como se observa en la

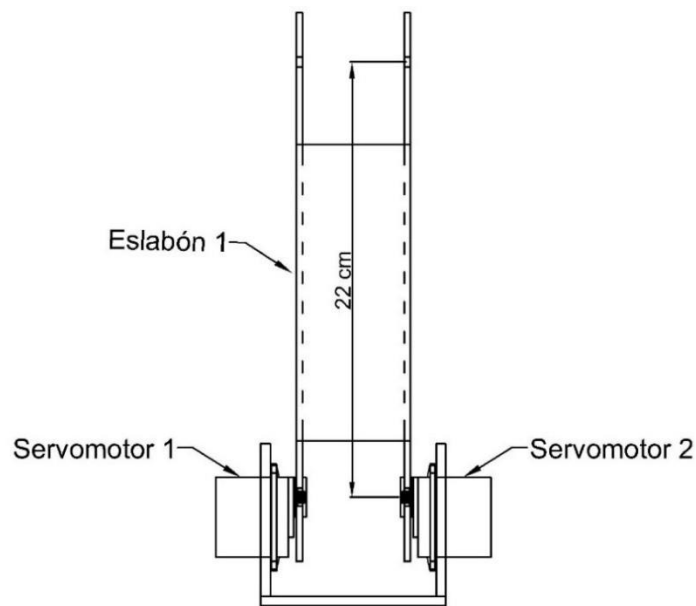


Figura 16. Servomotores acoplados a eslabón.

El torque aplicado al eslabón de la Figura 16 será de 20 kgf/cm, la carga estará a 22 cm y el peso que deberá cargar ambos actuadores será el peso total del brazo robótico 0.423 kg.

En la Figura 17 muestra la relación kgf/cm para un torque de 20 kgf/cm a 22 cm.

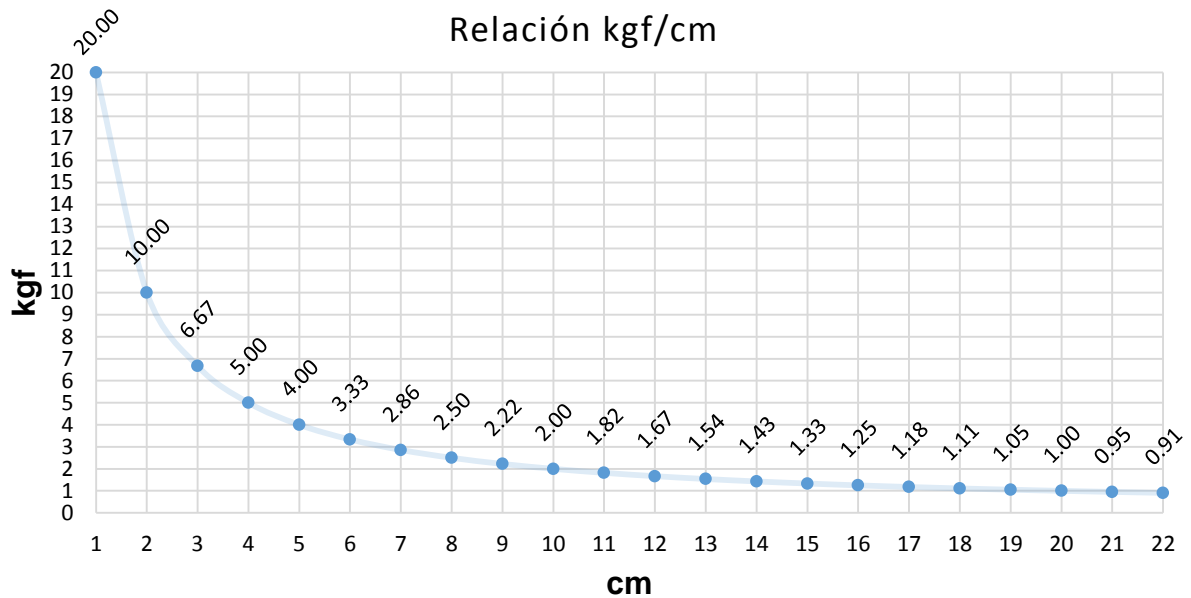


Figura 17. Grafica de la relación kgf/cm.

En la gráfica se observa que para una distancia de 22 cm aplicando un torque de 20 kgf/cm los servomotores serán capaces de cargar 0.91 kg.

Como se mostró en la ecuación 6 el peso total del brazo robótico es de 0.423 kg por lo tanto en la ecuación 7 se muestra que la capacidad de carga de los servomotores es mayor que el peso total del brazo robótico.

$$0.910 \text{ kg} > 0.423 \text{ kg} \quad (7)$$

En la Figura 18 se muestra la capacidad de carga que tendrá un servomotor de 20 kgf/cm si el peso se ubica a 22 centímetros del eje del motor.

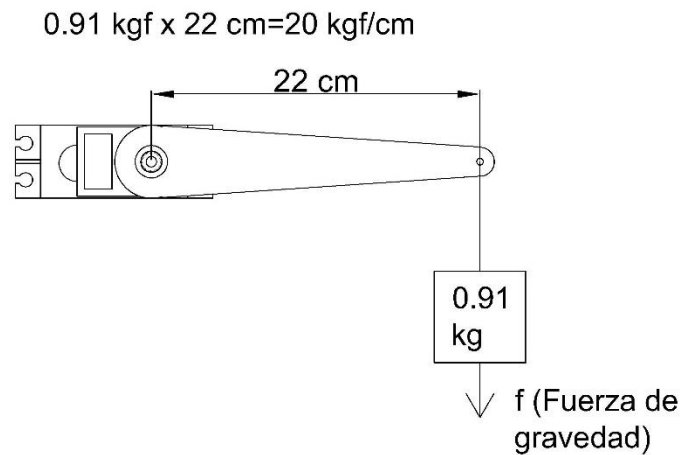


Figura 18. Capacidad de carga para un torque de 20 kgf/cm.

Los actuadores del primer eslabón tienen la capacidad de cargar aproximadamente del doble del peso total del brazo de acuerdo a sus hojas de datos, en base a esto se determina que las dimensiones que se proponen son adecuadas para este tipo de servomotores.

3.2.2 AutoCAD

El diseño de cada una de las piezas del robot se realizó por medio de software AutoCAD, las dimensiones de cada una de las piezas que formaran los eslabones se diseñaran a partir de las medidas propuestas en la Figura 19 se muestra cada una de las piezas que conformaran el brazo robótico. Estas se clasifican en 2 colores distintos según el espesor en el que será cortada cada pieza, para el acrílico de 0.5 cm serán las piezas de color azul con las que se construirá la base y para el acrílico de 0.3 cm

serán las piezas de color rojo que formarán los eslabones y el *gripper*, cada una de las piezas de este dibujo serán guardadas en un archivo independiente en formato *.dxf para posteriormente ser importadas y seccionadas en una cortadora laser.

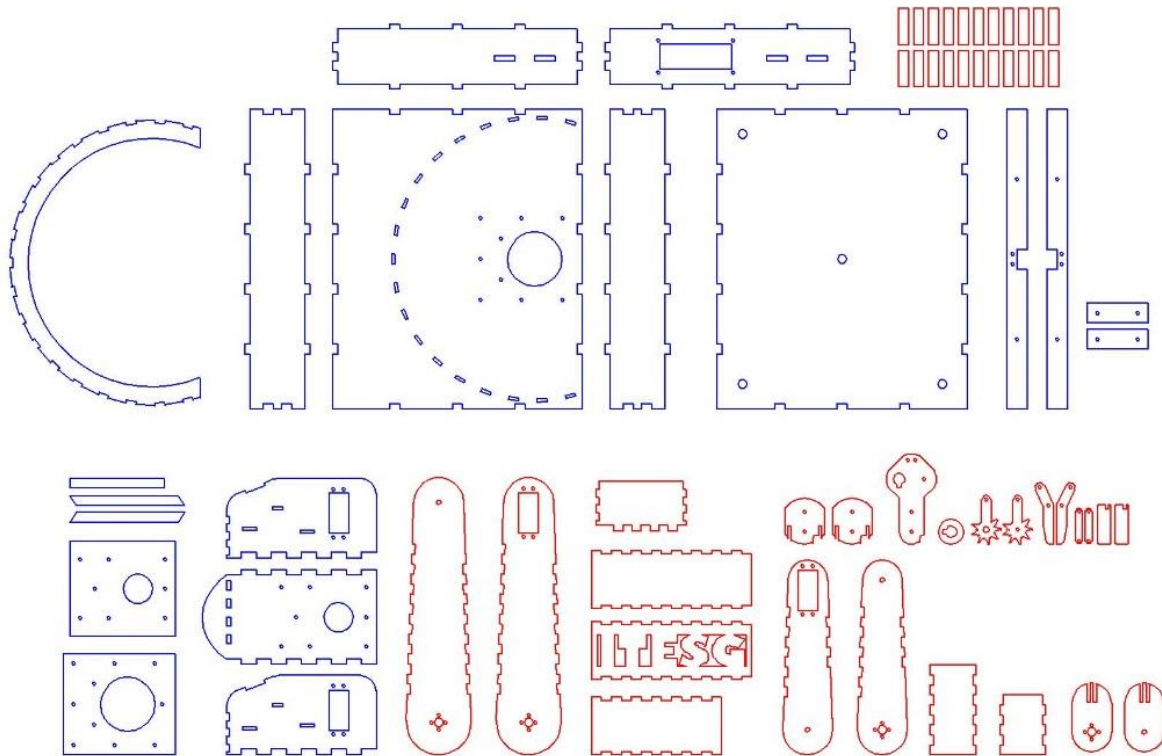


Figura 19. Diseño de las piezas en AutoCAD.

En la Figura 20 se muestra la vista lateral del ensamble dividido en dos secciones, la base en color azul (acrílico de 0.5 cm) y los eslabones en color rojo (acrílico de 0.3 cm) el acrílico de mayor espesor será utilizado en general para la base y el acrílico de menor espesor será el que constituirá los eslabones y *gripper*.

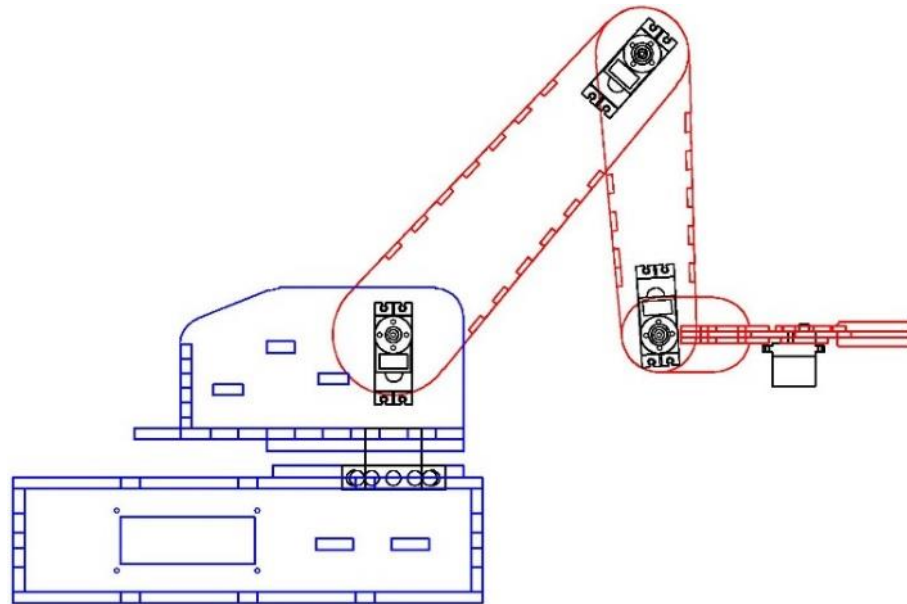


Figura 20. Vista lateral del brazo robótico.

3.2.3 Inventor

Cada una de las piezas dibujadas en AutoCAD serán importadas al software de diseño 3D Inventor, esto con el fin de realizar un ensamble de todos los componentes para corregir posibles errores que se hayan generado en el diseño y tener un vista general del brazo robótico totalmente ensamblado. Posteriormente, serán seccionadas en la cortadora laser, en la Figura 21 se muestran todas las piezas en 3D importadas en Inventor.

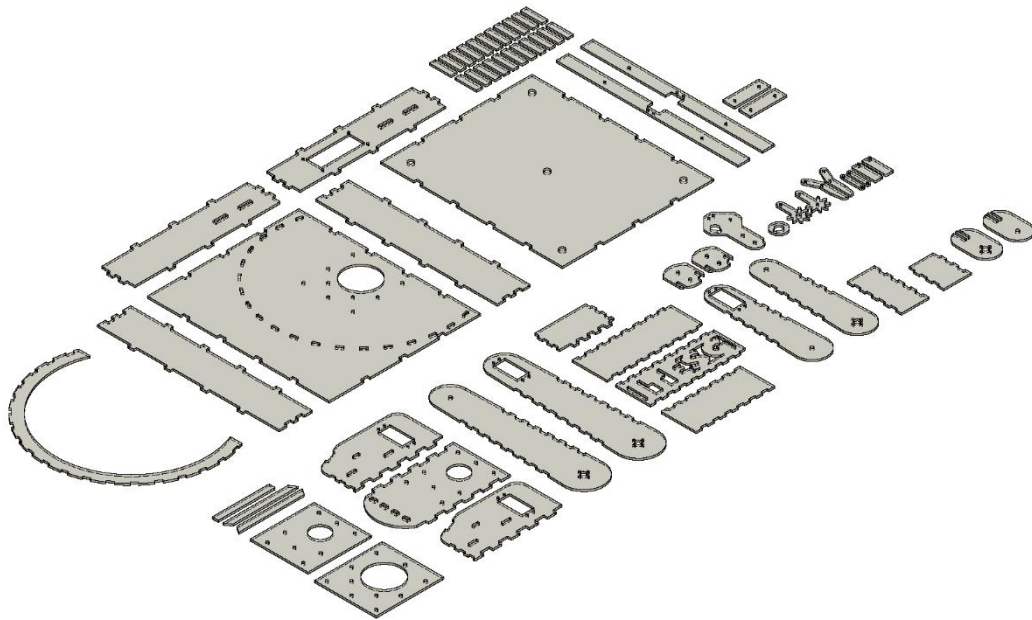


Figura 21. Vista de piezas en 3D.

Los actuadores también deberán ser importados al software de diseño en la Figura 22 se muestran todos los servomotores que conformaran robot, cinco son del modelo MG995 y uno del modelo MG90S.

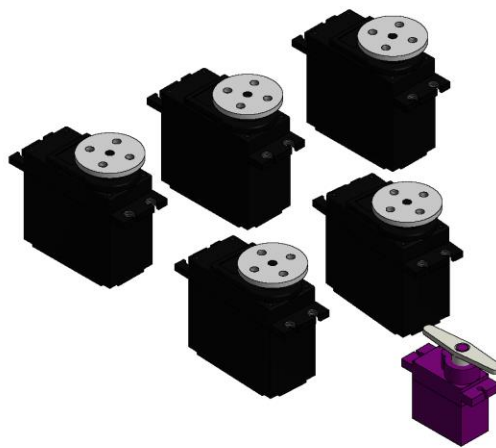


Figura 22. Servomotores creados en inventor.

Entre las principales características que posee Inventor y lo posicionan entre uno de los mejores software de diseño 3D, es que dentro de su entorno se pueden realizar ensambles de mecanismos complejos y hacer la simulación de sus movimientos, este procedimiento minimiza los errores de diseño y ofrece una vista preliminar del mecanismo que se está creando, en la Figura 23 se muestra el ensamble general del brazo robótico utilizando Inventor.

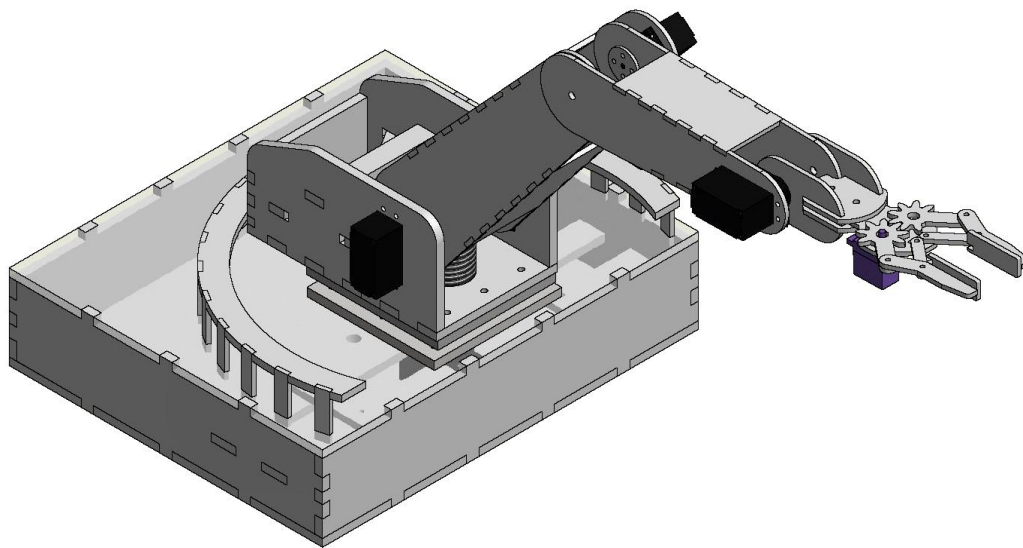


Figura 23. Ensamble general del brazo robótico realizado en Inventor.

3.3 Cinemática

En este subcapítulo se explicará la resolución de la cinemática inversa del brazo robótico por medio de los parámetros de Denavit – Hartenberg.

3.3.1 Cinemática Directa

En la Figura 24 se asignan los ejes de referencia, la identificación de las articulaciones, y la numeración de cada eslabón del brazo robótico.

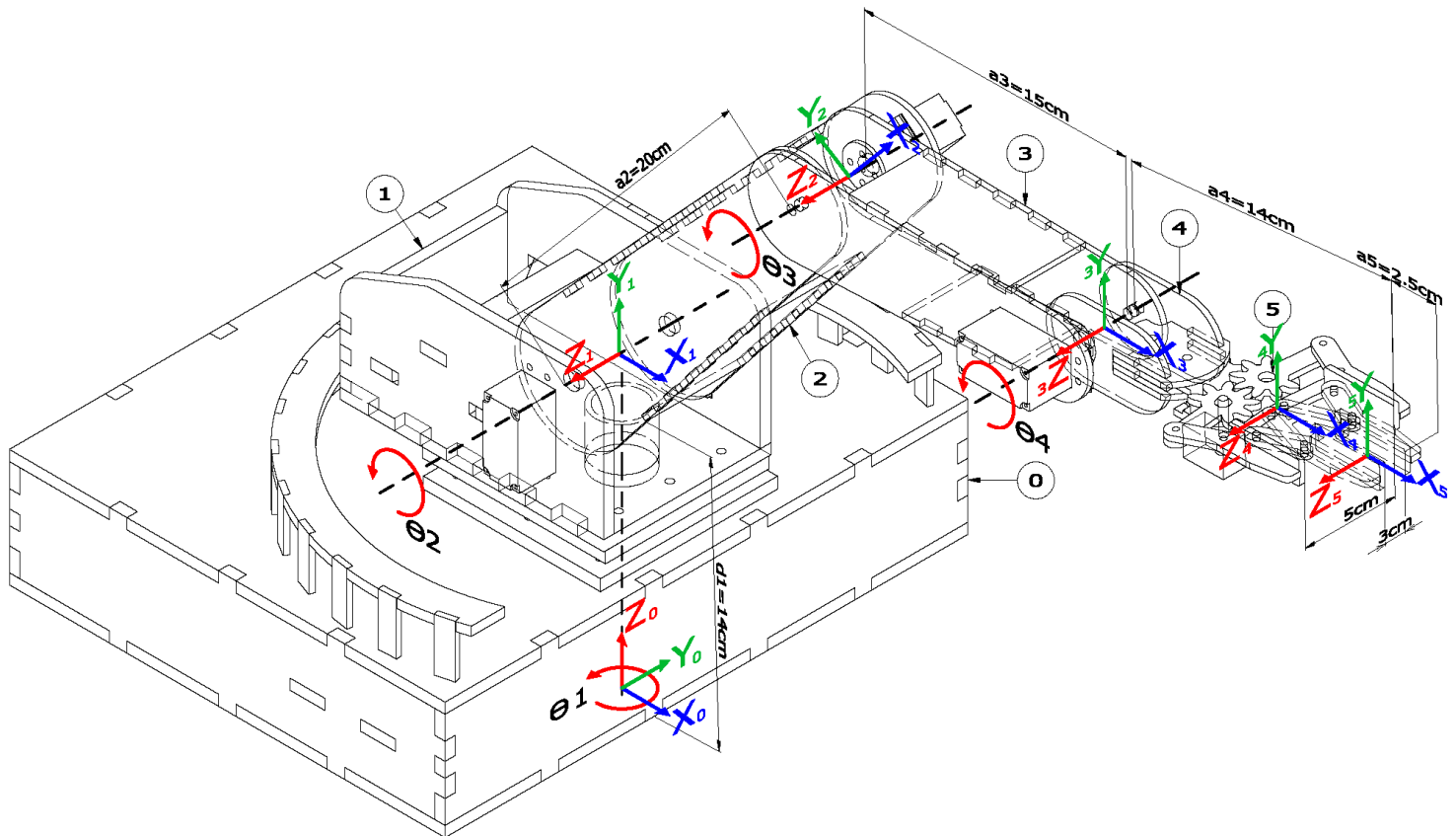


Figura 24. Asignación de ejes de referencia del brazo robótico.

La Tabla 3 muestra los parámetros de algoritmo de Denavit – Hartenberg para el robot antropomórfico de cinco grados de libertad, el parámetro θ_i representa el ángulo de rotación de cada eslabón, se mide con respecto al eje Z , d_i representa la distancia para este caso desde el inicio de la base hasta el próximo eje de referencia se mide a lo largo del eje Z y como referencia los ejes X_0 y X_1 deben quedar alineados, a_i



representa la distancia de cada eslabón y se mide respecto al eje X, α_i representa el ángulo que deberá girar el nuevo eje de referencia con respecto al anterior sobre el eje X.

Tabla 3. Parámetros D-H para el robot antropomórfico propuesto

ARTICULACIÓN	θ_i	$d_i(\text{cm})$	$a_i(\text{cm})$	α_i
1	θ_1	14 cm	0 cm	90°
2	θ_2	0 cm	22 cm	0°
3	θ_3	0 cm	15 cm	0°
4	θ_4	0 cm	14 cm	0°
5	0°	0 cm	2.5 cm	0°

Como ya se mencionó anteriormente el robot es del tipo antropomórfico y se caracteriza por tener todas sus articulaciones del tipo rotacional, por lo tanto, las matrices de transformación homogénea adquieren la siguiente forma:

$${}^0A_1 = \begin{bmatrix} \cos\theta_1 & 0 & \text{sen}\theta_1 & 0 \\ \text{sen}\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & 14 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$${}^1A_2 = \begin{bmatrix} \cos\theta_2 & -\text{sen}\theta_2 & 0 & 22\cos\theta_2 \\ \text{sen}\theta_2 & \cos\theta_2 & 0 & 22\text{sen}\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$${}^2A_3 = \begin{bmatrix} \cos\theta_3 & -\text{sen}\theta_3 & 0 & 15\cos\theta_3 \\ \text{sen}\theta_3 & \cos\theta_3 & 0 & 15\text{sen}\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$



$${}^3A_4 = \begin{bmatrix} \cos\theta_4 & -\text{sen}\theta_4 & 0 & 14\cos\theta_4 \\ \text{sen}\theta_4 & \cos\theta_4 & 0 & 14\text{sen}\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$${}^4A_5 = \begin{bmatrix} 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

Para determinar la matriz T se multiplican todas las matrices:

$$T = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 \quad (13)$$

Quedando de la siguiente forma:

$$T = \begin{bmatrix} \cos\theta_1 & 0 & \text{sen}\theta_1 & 0 \\ \text{sen}\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & 14 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\text{sen}\theta_2 & 0 & 22\cos\theta_2 \\ \text{sen}\theta_2 & \cos\theta_2 & 0 & 22\text{sen}\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

$$\begin{bmatrix} \cos\theta_3 & -\text{sen}\theta_3 & 0 & 15\cos\theta_3 \\ \text{sen}\theta_3 & \cos\theta_3 & 0 & 15\text{sen}\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_4 & -\text{sen}\theta_4 & 0 & 14\cos\theta_4 \\ \text{sen}\theta_4 & \cos\theta_4 & 0 & 14\text{sen}\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Se sustituirá *cos* por *c* y *sen* por *s*. Así mismo, se utilizará la identidad para la suma y resta de ángulos obteniendo los siguientes resultados:

$${}^0A_1 {}^1A_2 = \begin{bmatrix} c\theta_1 c\theta_2 & -c\theta_1 s\theta_2 & s\theta_1 & 22c\theta_1 c\theta_2 \\ s\theta_1 c\theta_2 & -s\theta_1 s\theta_2 & -c\theta_1 & 22s\theta_1 c\theta_2 \\ s\theta_2 & c\theta_1 & 0 & 22s\theta_2 + 14 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

$${}^0A_2 {}^2A_3 = \begin{bmatrix} c\theta_1 c(\theta_2 + \theta_3) & -c\theta_1 s(\theta_2 + \theta_3) & s\theta_1 & 15c\theta_1 c(\theta_2 + \theta_3) + 22c\theta_1 c\theta_2 \\ s\theta_1 c(\theta_2 + \theta_3) & -s\theta_1 s(\theta_2 + \theta_3) & -c\theta_1 & 15s\theta_1 c(\theta_2 + \theta_3) + 22s\theta_1 c\theta_2 \\ s(\theta_2 + \theta_3) & c(\theta_2 + \theta_3) & 0 & 15s(\theta_2 + \theta_3) + 22s\theta_2 + 14 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$



$${}^0A_3{}^3A_4 = \begin{bmatrix} c\theta_1c(\theta_2 + \theta_3 + \theta_4) & -c\theta_1s(\theta_2 + \theta_3 + \theta_4) & s\theta_1 & 14c\theta_1c(\theta_2 + \theta_3 + \theta_4) + 15c\theta_1c(\theta_2 + \theta_3) + 22c\theta_1c\theta_2 \\ s\theta_1c(\theta_2 + \theta_3 + \theta_4) & -s\theta_1s(\theta_2 + \theta_3 + \theta_4) & -c\theta_1 & 14s\theta_1c(\theta_2 + \theta_3 + \theta_4) + 15s\theta_1c(\theta_2 + \theta_3) + 22s\theta_1c\theta_2 \\ s(\theta_2 + \theta_3 + \theta_4) & c(\theta_2 + \theta_3 + \theta_4) & 0 & 14s(\theta_2 + \theta_3 + \theta_4) + 15s(\theta_2 + \theta_3) + 22s\theta_2 + 14 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

$${}^0A_4{}^4A_5 = T =$$

$$\begin{bmatrix} c\theta_1c(\theta_2 + \theta_3 + \theta_4) & -c\theta_1s(\theta_2 + \theta_3 + \theta_4) & s\theta_1 & 2.5[c\theta_1c(\theta_2 + \theta_3 + \theta_4)] + 14c\theta_1c(\theta_2 + \theta_3 + \theta_4) + 15c\theta_1c(\theta_2 + \theta_3) + 22c\theta_1c\theta_2 \\ s\theta_1c(\theta_2 + \theta_3 + \theta_4) & -s\theta_1s(\theta_2 + \theta_3 + \theta_4) & -c\theta_1 & 2.5[s\theta_1c(\theta_2 + \theta_3 + \theta_4)] + 14s\theta_1c(\theta_2 + \theta_3 + \theta_4) + 15s\theta_1c(\theta_2 + \theta_3) + 22s\theta_1c\theta_2 \\ s(\theta_2 + \theta_3 + \theta_4) & c(\theta_2 + \theta_3 + \theta_4) & 0 & 2.5[s(\theta_2 + \theta_3 + \theta_4)] + 14s(\theta_2 + \theta_3 + \theta_4) + 15s(\theta_2 + \theta_3) + 22s\theta_2 + 14 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(18)

$$\text{Dada la matriz: } T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$n_x = c\theta_1c(\theta_2 + \theta_3 + \theta_4) \quad o_x = -c\theta_1s(\theta_2 + \theta_3 + \theta_4) \quad a_x = s\theta_1$$

$$p_x = 2.5[c\theta_1c(\theta_2 + \theta_3 + \theta_4)] + 14c\theta_1c(\theta_2 + \theta_3 + \theta_4) + 15c\theta_1c(\theta_2 + \theta_3) + 22c\theta_1c\theta_2$$

$$n_y = s\theta_1c(\theta_2 + \theta_3 + \theta_4) \quad o_y = -s\theta_1s(\theta_2 + \theta_3 + \theta_4) \quad a_y = -c\theta_1$$

(19)

$$p_y = 2.5[s\theta_1c(\theta_2 + \theta_3 + \theta_4)] + 14s\theta_1c(\theta_2 + \theta_3 + \theta_4) + 15s\theta_1c(\theta_2 + \theta_3) + 22s\theta_1c\theta_2$$

$$n_z = s(\theta_2 + \theta_3 + \theta_4) \quad o_z = c(\theta_2 + \theta_3 + \theta_4) \quad a_z = 0$$

$$p_z = 2.5[s(\theta_2 + \theta_3 + \theta_4)] + 14s(\theta_2 + \theta_3 + \theta_4) + 15s(\theta_2 + \theta_3) + 22s\theta_2 + 14$$



3.4 Corte y Ensamble

En este subcapítulo se explicará los métodos y procedimiento utilizados para el corte de las piezas y ensamble del brazo robótico.

3.4.1 Corte

Las piezas que conformaran el brazo robótico serán seccionadas en cortadora laser de la marca LaserMex modelo LX-4060SE.

El archivo de diseño generado en AutoCAD fue exportado e importado a la cortadora en formato *.dxf, esta extensión es compactible con este tipo de sistemas de corte, una vez realizado esto se cortaron las piezas en sus respectivos espesores de acrílico, en la Figura 25 se muestra el corte de las piezas de acrílico de 0.3 cm de espesor.



Figura 25. Corte de acrílico de 0.3 cm de espesor.

En la Figura 26 se muestra el corte de las piezas de acrílico de 0.5 cm de espesor



Figura 26. Corte de acrílico de 0.5 cm de espesor.

3.4.2 Ensamble

Una vez cortadas todas las piezas se continuará con el ensamble completo del mecanismo comenzado con la base principal, esta tendrá la función de contrapeso. Por este motivo será construida en acrílico de 0.5 cm de espesor, en esta parte se instalará el primer actuador que realizara la función de rotación de todo el mecanismo, dicho actuador en primera instancia fue unido a una base giratoria hecha en solera de 4" de ancho. Sin embargo, el balero instalado en esta se dañaba por el sobrecalentamiento que se producía al realizar la soldadura por arco eléctrico por tal motivo se optó por construir la base giratoria en acrílico de 0.5 cm de espesor obteniendo un excelente resultado en el giro, en la Figura 27 se muestra el primer eslabón acoplado a la primera base creada en solera.

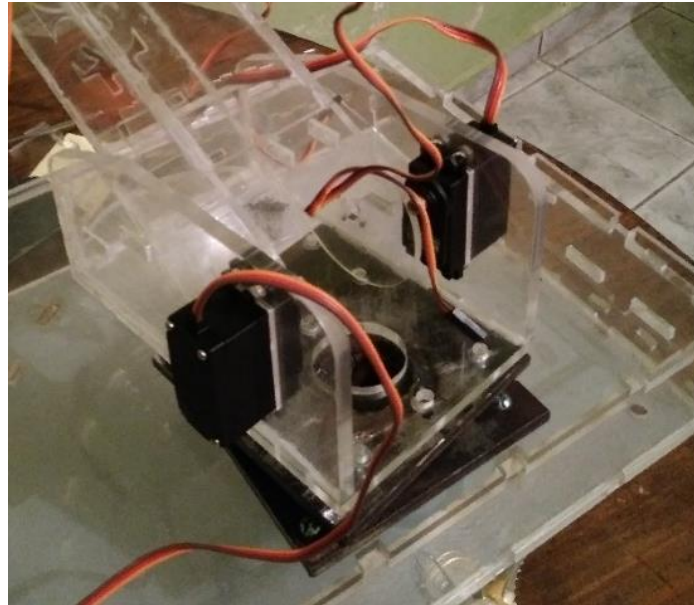


Figura 27. Base giratoria construida en solera.

En la Figura 28 se muestra la base final construida en acrílico.

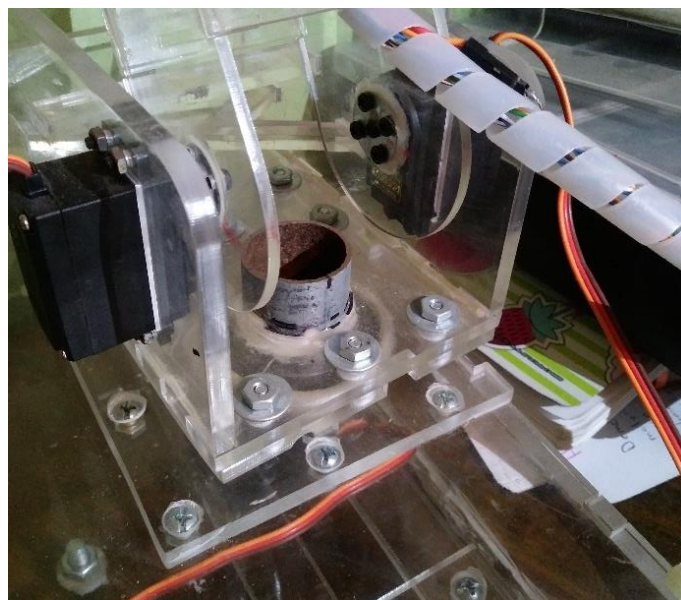


Figura 28. Base giratoria construida en acrílico.

Ensamblada la base principal y la base giratoria se continuó con la construcción del brazo y *gripper* en esta parte se colocaron el resto de los actuadores, los dos servomotores del primer eslabón que cargaran el peso completo del brazo se ajustaron para sincronizar sus giros ya que uno girara en sentido de las manecillas del reloj y el otro en sentido contrario en todo momento que el mecanismo este trabajando, los actuadores restantes no presentaron ningún inconveniente en su instalación, en la Figura 29 se muestra el ensamble del brazo.

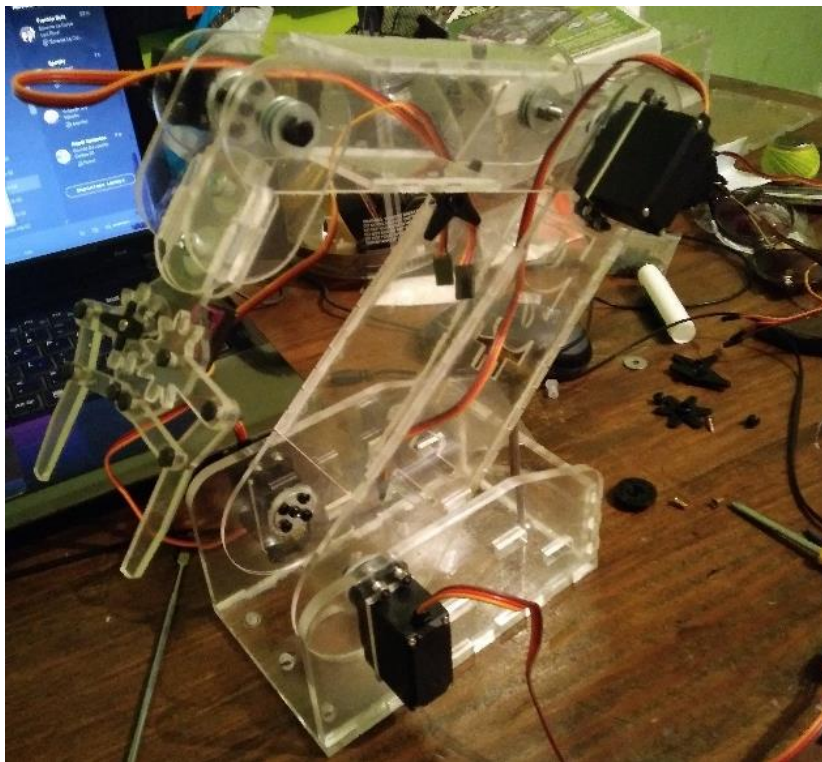


Figura 29. Ensamble físico del brazo robótico.

En la Figura 30 se muestra el acoplamiento del brazo y de la base giratoria con esto es creado el ensamble general del brazo robótico.

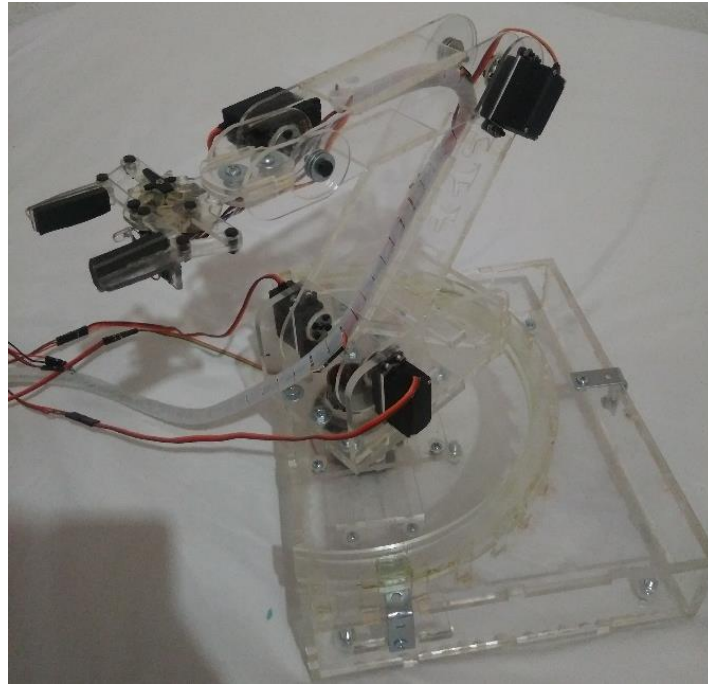


Figura 30. Ensamble general del brazo robótico y la base.

3.5 Control

El control de todo el sistema se realizará por medio del FPGA modelo Nexys 2 y el hardware de control que será ingresado se describirá en el software Active-HDL. El hardware tendrá un *reset* general que mandara a todo el sistema a sus condiciones de inicio (*home*) y se pondrá en modo espera hasta recibir la señal que le indique al sistema que debe de comenzar a trabajar, esta será la única señal que deberá ingresar el operador del robot en todo el proceso. Al recibir esta señal la banda transportadora girara acercando el objeto a clasificar al robot. El sistema contara con dos tipos de sensores de diferente modelo para determinar la ubicación del objeto en su trayectoria

por la banda transportadora, uno de estos sensores ubicado en el extremo final de la banda será el encargado de enviar la señal al FPGA cuando el objeto se encuentre al alcance del brazo robótico y activara los servomotores para posicionar el brazo a un centímetro aproximadamente del objeto y activar el sensor RGB, al identificar el color del objeto el gripper del robot sujetara el objeto y los servomotores se activarán para posicionarlo en su respectiva ubicación dependiendo de su color. Concluido esto, el sistema se pondrá en modo de espera nuevamente y el brazo tomará su posición inicial para realizar el proceso nuevamente.

3.5.1 Sensor HC-SR04

El sensor ultrasónico HC-SR04 mostrado en la Figura 31 será el encargado de identificar la posición del objeto a clasificar.



Figura 31. Sensor HC-SR04.

En la Figura 32 se describe las señales que son generadas en cada uno de sus pines.

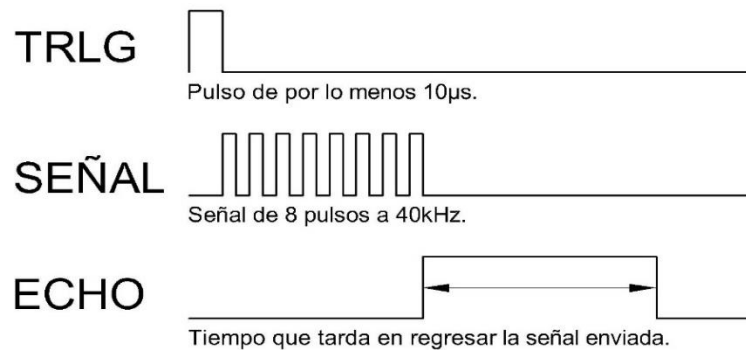


Figura 32. Señales generadas por el sensor HC-SR04.

Este sensor cuenta con cuatro pines llamadas *Trig*, *Echo*, *Vcc* y *Gnd* estos dos últimos son para su alimentación a 5 volts y tierra. Su funcionamiento es de la siguiente manera, se envía una señal al pin *Trig* en este caso por medio del FPGA de por lo menos 10 µs de duración, el sensor detectara esta señal y enviara una señal ultrasónica de 8 pulsos a una frecuencia de 40 KHz y en este momento el pin *Echo* cambiara su estado de '0' (bajo) a '1' (alto), si el pulso ultrasónico enviado encuentra un obstáculo en su camino el sonido rebotara y regresara al sensor este al detectar el eco cambiara el estado del pin *Echo* de '1' (alto) a '0' (bajo), este ciclo se repetirá cada 60 ms, la señal generada por el pin *Echo* deberá ser registrada en un contador de tiempo ya que la distancia a la que se encuentra el obstáculo es proporcional a la duración de este pulso y se calculara con las siguientes formulas y datos.

En la ecuación 20 se muestra la velocidad del sonido a una temperatura de 25 °C.

$$Velocidad\ del\ sonido = 340 \frac{m}{s} \quad (20)$$

En la ecuación 21 se muestra la fórmula general para obtener la velocidad.

$$v = \frac{d}{t} \quad \rightarrow \quad t = \frac{d}{v} \quad (21)$$



En la ecuación 22 se muestra la fórmula para obtener la frecuencia en base al tiempo

$$f = \frac{1}{t} \quad (22)$$

Donde:

C = frecuencia con respecto al reloj interno del FPGA

t = tiempo f = frecuencia d = distancia v = velocidad

Como se menciona anteriormente, este sensor ultrasónico será uno de los encargados de detectar la ubicación del objeto a clasificar. En específico este sensor determinará si el objeto sigue su correcta trayectoria en la banda transportadora, así como monitorear la distancia a la que se encontrara después de pasar por la guías que se instalaran en la banda transportadora para corregir su posición la ubicación ideal debería de ser entre 4.5 y 5.5 cm, por lo tanto cada que el pulso generado por el pin Echo este en '1' (alto) se realizara un conteo interno en el FPGA a la velocidad de 50 MHz que es la frecuencia con la que trabaja su reloj interno, con esta información realizamos las operaciones siguientes.

En la ecuación 23 se muestra el tiempo que tarda el sonido en recorrer 4.5 cm.

$$t = \frac{0.11 \text{ m}}{340 \frac{\text{m}}{\text{s}}} = 0.0001323529411 \text{ s} \times 2 = 264.70588 \text{ } \mu\text{s} \quad (23)$$

En la ecuación 24 se obtiene la frecuencia en base al tiempo obtenido en la ecuación 23.

$$f = \frac{1}{264.70588 \text{ } \mu\text{s}} = 3,777.77 \text{ Hz} \quad (24)$$

En la ecuación 25 se obtiene la relación de la frecuencia correspondiente al tiempo que tardara el sonido en recorrer 4.5 cm con respecto al reloj interno del FPGA.

$$C = \frac{50MHz}{3,777.77 Hz} \approx 13,235 \quad (25)$$

El tiempo deberá ser multiplicado por dos ya que el sonido recorre dos veces la distancia a la que se encuentra el obstáculo, una vez para llegar a él y otra para regresar al sensor como se observa en la Figura 33.

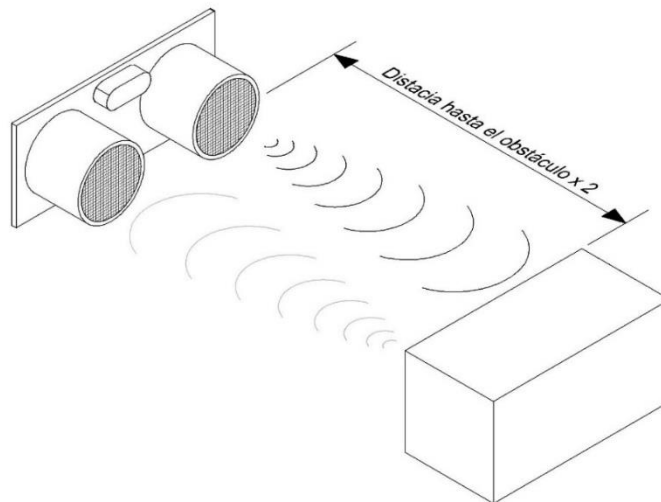


Figura 33. Sensor HC-RS04 mandando y recibiendo ultrasonido.

Por lo tanto, si el conteo interno que se realizó mientras el pin *Echo* está en '1' (alto) está entre 13,235 y 16,176 que son 4.5 cm y 5.5 cm respectivamente, el sistema enviara una señal para activar el siguiente proceso. La distancia a la que se encuentra el obstáculo se podrá visualizar en los led's que están incluidos en la PFGA, el conteo será en sistema binario, en la Figura 34 se muestra la conversión de decimal a binario

para identificar la distancia a la que se encuentre el objeto a través de la banda transportadora.

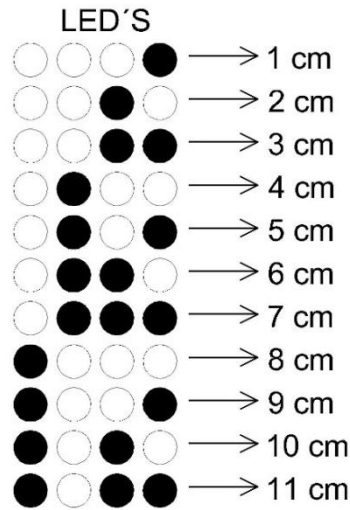


Figura 34. Conversión Decimal a Binario.

En la Figura 35 se muestra la máquina de estados finitos para el sensor HC-SR04 para ser escrito en VHDL.

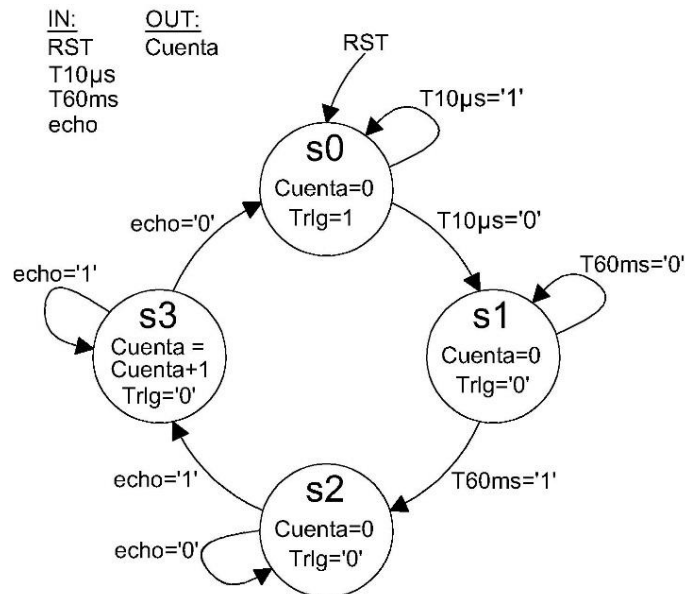


Figura 35. Máquina de estados finitos del sensor HC-SR04.

La máquina de estados funciona de la siguiente manera:

En el estado s_0 se envía un pulso por 10 μs al pin $Trlg$, con esto el sensor empezara a enviar una señal ultrasónica de 40 KHz, en el estado s_1 el sensor esperara 60 ms en lo que se envía la señal ultrasónica, en el estado s_2 el pin $echo$ cambiara su estado de 0 (bajo) a 1 (alto) pasando al estado s_3 , donde se realizara una cuenta que se detendrá cuando el sensor detecte que la señal ultrasónica ha sido rebotada, esta cuenta será convertida internamente en el FPGA a distancia.

Cada parte de la programación será representada por una máquina de estados finitos o FSM, estas máquinas son de esencial ayuda al momento de escribir el código, facilitando su redacción y entendimiento, en la Figura 36 se muestran las entradas y salidas que se generaran en la programación para el control del sensor ultrasónico.

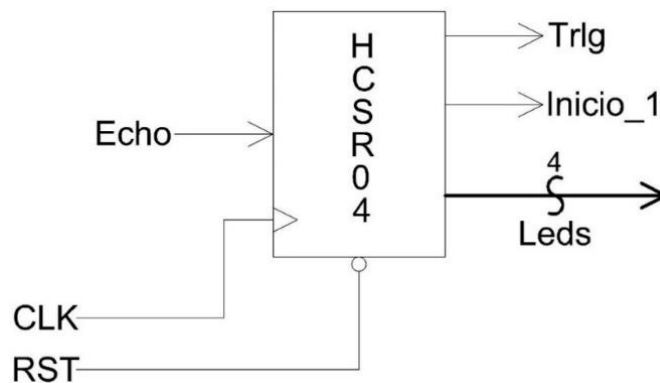


Figura 36. Diagrama de bloques para el sensor HC-RS04.

A continuación se describe el funcionamiento de la FSM; como se sabe para todo sistema se debe agregar un *Reset* con se asegura el correcto funcionamiento de la programación. Se generará internamente una base de tiempo de 60 ms, esto quiere decir que abra un pulso que se repetirá cada 60 ms, por medio de esta señal se generara un ancho de pulsos de 10 μs , de aquí partimos para posicionarnos en el estado s_0 . Para que haya un cambio de estado el ancho de pulsos de 10 μs debe de cambiar su estado de '1' a '0' con esto se controlara la salida *Trig* que enviara la

señal necesaria de $10 \mu s$ para que el sensor ultrasónico empiece a trabajar. En el estado $s1$ se esperará el cambio de pulso de la señal $T60ms$ para asegurar el retardo necesario. En el estado $s2$ el sensor hará un cambio en la señal $echo$ de '0' a '1' una vez registrado este cambio se pasara al estado $s3$ en este último estado se iniciara una conteo de tiempo expresado con la salida $Cuenta = Cuenta + 1$ hasta que la señal $echo$ regrese al estado inicial es decir de '1' a '0', este clico se repetirá indefinidamente, la cuenta que se registró en el proceso se convertirá a binario y se asignara a la salida Leds, en esta parte también se definirá la salida Inicio_1 cambiando su salida de '0' a '1' cuando el objeto se encuentre a una distancia de 11 cm, al concluir esto se reiniciara la cuenta y empezar todo el proceso nuevamente.

3.5.2 Sensor TCRT5000

El modulo sensor TCRT5000 mostrado en la

Figura 37 será el segundo sensor encargado de determinar la posición del objeto y estarán instalados en la banda trasportadora.

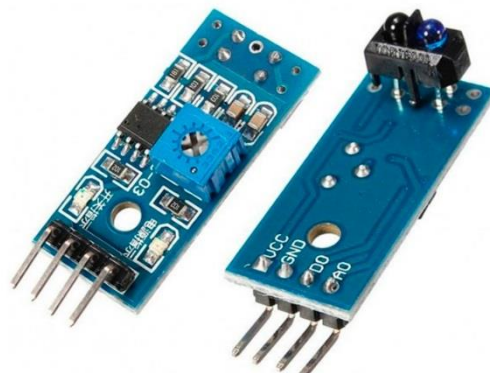


Figura 37. Sensor TCRT5000.



Se colocarán dos sensores de este modelo para distintas funciones, el primero se utilizara en el sistema implementado para acomodar la pieza al centro de la banda transportadora por medio de una guía, el objeto será ubicado aproximadamente en esta posición y el sensor enviara la señal al FPGA cuando esto se haya cumplido.

El segundo sensor se ubicara en el extremo final de la banda para detectar cuando el objeto se encuentre al alcáncense del brazo robótico y mandara la señal al FPGA para activar los servomotores, al mismo tiempo desactivara la banda transportadora para que el objeto se quede en su extremo final hasta que sea retirado por el brazo robótico.

La funcionalidad del sensor es sencilla y podrá ser implementada en cualquier sistema digital y analógico para detección de objetos. El modulo cuenta con cuatro pines para su conexión, los pines Vcc y GND son para alimentación a 5 volts, el pin D0_A es la salida digital y el pin A0 es la salida analógica. Para la aplicación que se le dará a los dos sensores en la banda transportadora no se ocupan las salidas analógicas ya que con la salida digital que ofrecen es suficiente para su correcto funcionamiento. La distancia de detección se encuentra entre 0.2 cm y 2.5 cm, cuando el sensor detecta un objeto dentro de su rango de detección enviará a la salida digital D0_A un 0 lógico (bajo) de lo contrario su salida estará en 1 lógico (alto). De igual manera el modulo cuenta con un led como ayuda visual que al contrario de su salida digital D0_A al detectar un objeto en su rango de detección el led estará en 1 lógico (alto) de lo contrario se mantiene en 0 lógico (bajo).

En la Figura 38 se muestra el funcionamiento del sensor así como el rango de operación máximo con el que trabaja, cabe mencionar que la aplicación principal para este tipo de sensores es en carros sigue líneas por la facilidad de detectar una línea negra en un fondo blanco, sin embargo se buscó probar otras aplicaciones y observar los resultados obtenidos.

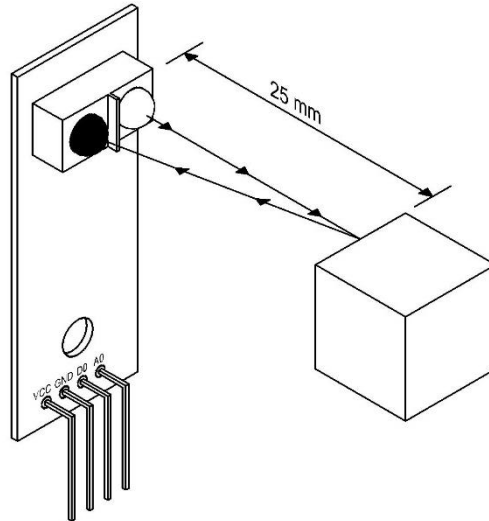


Figura 38. Sensor TCRT5000 en operación.

Para este sensor no habrá un diagrama de bloques (FSM) ni una máquina de estados finitos ya que por la simplicidad en su funcionamiento estas partes serán implementadas en otros procesos que se mencionaran más adelante.

3.5.3 Motor a pasos y Driver A4988

En esta subsección se explicara el funcionamiento de los motores a pasos bipolares y de los *driver* de control utilizados para su funcionamiento.

3.5.3.1 Motor a pasos Nema 17

El motor a pasos Nema 17 mostrado en la Figura 39 se encuentra dentro de la familia de los motores bipolares y se utiliza como el actuador principal para la banda transportadora.



Figura 39. Motor a pasos Nema 17

Este tipo de motor se caracteriza por tener una alta precisión en sus movimientos, cada paso completo corresponde a 1.8° , es decir, si se está trabajando a pasos completos y se desea que su eje de una vuelta completa, se enviara 200 pulsos para cumplir este propósito y se contara con la ventaja de saber exactamente el ángulo que va a girar el motor dependiendo del número de pulsos enviados, en la Figura 40 se muestra el diagrama para un motor a pasos bipolar.

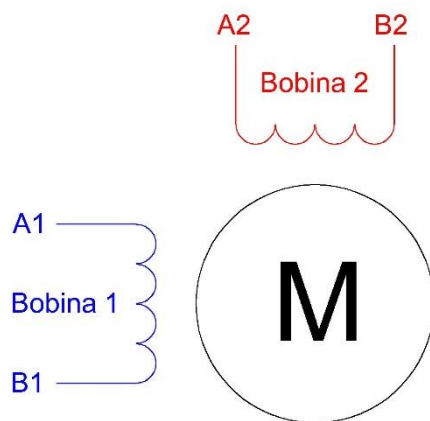


Figura 40. Diagrama motor a pasos bipolar.



Como se muestra en el diagrama anterior, los motores bipolares únicamente tienen dos bobinas para su conexión, por lo tanto, se consideran internamente más sencillos que los motores unipolares.

Los motores bipolares tienen mayor torque, mayor anclaje, son más pequeños y más baratos que los motores unipolares, la complejidad que se presenta en los motores bipolares es con el sistema de control, porque no solo tiene que permitir pasar la corriente por la bobina, si no que tienen que cambiar la polaridad de la corriente, es decir, en un primer momento la corriente viaja en una dirección, produciendo un campo magnético norte/sur, y en el siguiente momento la dirección de la corriente se invierte, produciendo un campo magnético sur/norte, de aquí su denominación, ya que las bobinas están pensadas para cambiar de polaridad durante el movimiento.

Existen distintos sistemas de control o *driver* utilizados para el control de motores a pasos de baja potencia entre los más conocidos se encuentran los basados en el *chip* Allegro A4988 de *Texas Instruments* DRV8825 o incluso se puede construir una etapa de potencia con opto-acopladores. Sin embargo, hoy en día los *driver* basados en los chips anteriores presenta gran ventaja comparados con una etapa de potencia construida con opto-acopladores, por lo tanto, esta última opción no será viable para el presente proyecto, en la subsección siguiente se describe a detalle el driver que se utilizara para el control del motor a pasos.

Entre los principales parámetros a controlar para un motor a pasos se encuentran la velocidad de giro, la dirección de giro y la resolución de pasos, todo esto controlado por medio del *driver*.

El diagrama a bloques y la máquina de estados finitos (FSM) para el control del motor a pasos se describirá en la subsección siguiente.

3.5.3.2 Driver A4988 y DRV8825

Para el control del motor a pasos Nema 17 se tenían dos opciones de *driver*, el chip A4988 de Allegro y el DRV8825 de Texas Instruments, ambos *drivers* son de bajo costo y de fácil implementación con algunas diferencias entre ambos, sin embargo, su funcionamiento es bastante parecido.

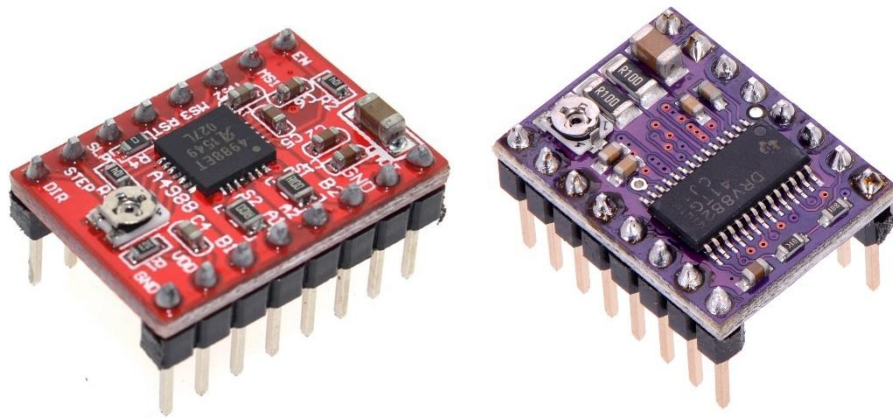


Figura 41. Driver A4988 y DRV8825.

En su página web Luis Llamas (2016) define a los *drivers* A4988 y DRV8825 como controladores que simplifican el manejo de motores a pasos desde un automata o procesador como un FPGA, estos controladores permiten manejar los altos voltajes e intensidades que requieren estos motores, limitar la corriente que circula por el motor, y proporcionar las protecciones para evitar que la electronica pueda resultar dañada. Para su control únicamente requieren dos salidas digitales, una para indicar el sentido de giro y otra para indicar que el motor avance un paso. Además, permiten realizar microstepping, una técnica para conseguir precisiones superiores al paso nominal del motor, el A4988 ha alcanzado una gran popularidad en sus últimos tiempos debido a su uso en proyectos como, por ejemplo, en impresoras 3D caseras.

Por su parte, el DRV8825 es una versión mejorada del A4988 y, por tanto, tiene unas características ligeramente superiores. En particular, el DRV8825 permite trabajar con tensiones superiores al A4988 (45 volts frente a 35 volts), e intensidades superiores (2.5 amperes frente a 2 amperes). Además, añade un nuevo modo de microstepping (1/32) que no está presente en el A4988, por lo demás, aparte de unas pequeñas diferencias, ambos dispositivos son similares en su montaje y uso. Incluso pueden ser compatibles entre sí, es decir, bajo ciertas condiciones podemos usar indistintamente uno u otro, e incluso una combinación de ambos de forma simultánea.

El motor a pasos Nema 17 utilizado como actuador en la banda transportadora trabaja con 12 volts y consume una corriente de 1.7 amperes por bobina, en base a esta información se decidió utilizar el *driver* A4988. Este sistema de control está dentro de los requerimientos de operación del motor, además que se trabajara en pasos completos, por lo tanto no será necesario utilizar microstepping, otra ventaja que tiene este *driver* es que es de más fácil adquisición que su adversario y a pesar que ambos *driver* son de bajo costo el A4988 tiene un precio aún más bajo que el DRV8825. El *driver* A4988 se muestran en la Figura 42.

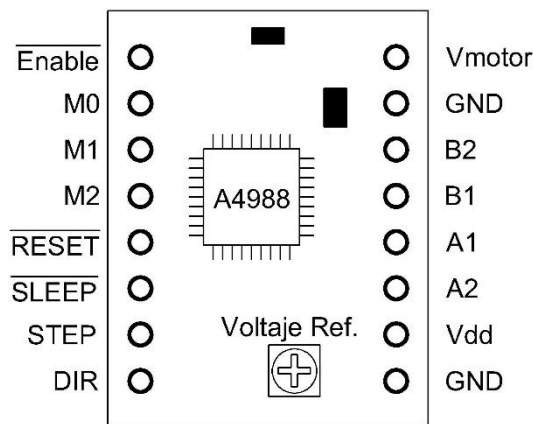


Figura 42. Identificación de pines driver A4988.

La funcionalidad de los pines del *driver* A4988 se describe en la Tabla 4.



Tabla 4. Descripción de los pines del driver A4988

Pines de conexión para el driver A4988			
Enable	Habilitar o deshabilitar la salida a las fases del motor.	Vmotor	Voltaje para el motor.
M0	Selección de pasos completos o micropasos.	GND	Tierra.
M1		B2	Se conectan a fases del motor a pasos.
M2		B1	
RESET	Reset del <i>driver</i> .	A1	
SLEEP	Suspender salida del <i>driver</i> .	A2	
STEP	Numero de pasos que avanzara el motor.	Vdd	Voltaje para el <i>Driver</i> .
DIR	Dirección de giro del motor.	GND	Tierra.

El pin *Enable* como se menciona en la tabla tiene la función de habilitar o deshabilitar la salida a las fases del motor, esto quiere decir que si queremos que el motor se detenga o avance lo haremos por medio de este pin, el sensor TCRT5000 ubicado en el extremo final de la banda transportadora será el encargado de realizar esta función, la salida del sensor (D0_A) mandara la señal correspondiente por medio del FPGA a la entrada del driver (*Enable*).

Los pines M0, M1 y M2 son los encargados de realizar pasos completos o micropasos dependiendo de la combinación en la conexión de sus pines, estos pines están conectados a tierra con una resistencia *Pull-Up*, por lo tanto si no se realiza ninguna conexión en ellos estarán directamente conectados a '0' lógico (bajo), en la Tabla 5 se muestra la conexión de estos pines para las distintas resoluciones de pasos.

Tabla 5. Conexión para resolución de pasos.

Resolución	M0	M1	M2
Paso Completo	Bajo	Bajo	Bajo
1/2 Paso	Alto	Bajo	Bajo
1/4 Paso	Bajo	Alto	Bajo
1/8 Paso	Alto	Alto	Bajo
1/16 Paso	Alto	Alto	Alto



La banda transportadora no variara la velocidad del motor por medio del FPGA, la resolución de los micro-pasos será constante a 1/8 de paso y se realizara directamente por medio de la conexión de los pines correspondientes en el *driver*.

Los pines *Reset* y *Sleep* no se utilizaran para nuestro proyecto por lo tanto se conectaran entre sí para habilitar el driver.

El pin *Step* estará conectado al FPGA, por medio de este pin se enviará los pulsos al driver que determinaran los pasos que avanzara el motor a pasos, como se mencionó anteriormente el motor a pasos Nema 17 tiene una resolución por paso completo de 1.8° lo que corresponde a 200 pasos para completar una vuelta completa como se muestra en la ecuación 26.

$$\text{Giro completo} \rightarrow 1.8^\circ \times 200 = 360^\circ \quad (26)$$

Con respecto al reloj interno del FPGA se realizará los siguientes cálculos mostrados en la ecuación 27 para terminar las características del pulso que controlara los pasos del motor.

$$t = \frac{1}{f} \rightarrow \frac{1}{200\text{Hz}} = 0.005 \text{ seg.} \quad (27)$$

Esto se obtiene sabiendo que un hercio representa un ciclo por cada segundo, s^{-1} , entendiendo ciclo como la repetición de un evento. Por ejemplo, el hercio se aplica en física a la medición de la cantidad de veces por un segundo que se repite una onda (ya sea sonora o electromagnética), magnitud denominada frecuencia y que es, en



este sentido, la inversa del período. Un hercio es la frecuencia de una partícula en un período de un segundo.

$$Hz = s^{-1} \quad (28)$$

$$v[Hz] = \frac{1}{T[s]} \quad (29)$$

Con esto obtenemos que para que el motor a pasos de una vuelta completa en un segundo debemos mandar un pulso cada 0.005 segundos, relacionando este tiempo con el reloj interno del FPGA (50MHz) se obtiene el siguiente resultado mostrado en la ecuación 30.

$$c = \frac{50MHz}{200Hz} = 250,000 \quad (30)$$

Se creará un base de tiempo que envíe un pulso cada 250,000 ciclos del reloj interno del FPGA para garantizar una vuelta completa en el motor a pasos cada segundo.

El pin Dir controlara la dirección de giro del motor, su funcionalidad es bastante sencilla ya que la banda transportadora estar girando en todo momento en sentido contrario a las manecillas del reloj por lo que bastara con poner este pin a '1' lógico directamente en el driver.

Los pines Vmotor y GND serán la alimentación del motor desde una fuente externa de 12 volts, el fabricante del driver recomienda poner un capacitor electrolítico de 100µF entre ambos pines con la finalidad de no dañar el driver.



Los pines A1, A2, B1 y B2 son conectados directamente a las fases del motor en el orden correcto para evitar que el motor gire en sentido contrario o no realice el giro correctamente.

Y por último los pines Vdd y GND son la alimentación del driver a 5 volts Esta alimentación se toma directamente del FPGA, como se observa existen dos pines llamados GND uno para el motor y otro para el driver, se debe realizar una conexión entre ambos para que el sistema funcione ya que de lo contrario no se obtendrán los resultados esperados en el funcionamiento del motor a pasos.

En la Figura 43 se muestra la máquina de estados finitos para el control del Driver A4988.

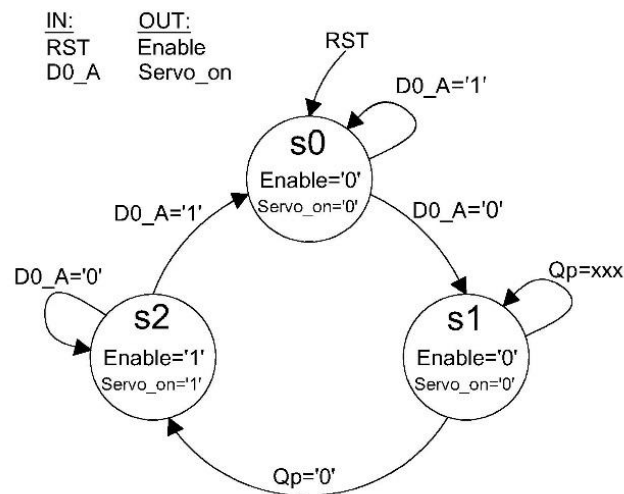


Figura 43. Máquina de estados finitos para el control del driver A4988.

La máquina de estados funciona de la siguiente manera, la señal de entrada RST es la encargada de mandar todo el sistema al inicio, mientras tanto la señal D0_A es la señal enviada por el sensor TCRT5000 y procesada por el FPGA. En el primer estado las salidas Enable y Servo_on estarán en '0' lógico (bajo) como se mencionó anteriormente D0_A estará en '1' lógico (alto) a menos que el sensor detecte un

obstáculo en su rango de operación, si esto sucede su salida pasara a '0' lógico (bajo) y el sistema pasara al siguiente proceso donde se inicia una cuenta en la señal interna llamada Qp, esto con la finalidad de dar el tiempo necesario al objeto para que llegue al extremo final de la banda, las señales de salida Enable y Servo_on seguirán en '0' lógico (bajo), una vez que la cuenta en Qp llegue a 0 se pasara al último estado, en este estado habrá un cambio en las salidas Enable y Servo_on a '1' logico (alto), por este motivo la banda transportadora se detendrá por la señal Enable y la señal Servo_on activara los servomotores del brazo robótico para realizar su secuencia y retirar el objeto del rango de operación del sensor TCRT5000, una vez que la señal D0 este en '1' lógico (alto) de nuevo el sistema se volverá a repetir.

La señal Step se realizará en otro proceso pero como solamente será una base de tiempo ya explicada anteriormente se agregara sin más a la máquina de estados finitos.

En la Figura 44 se muestra el diagrama de bloques para el control del *driver* A4988.

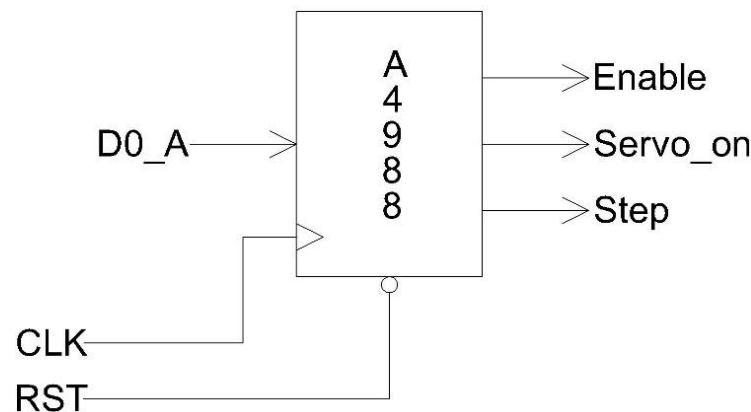


Figura 44. Diagrama de bloques para el control del driver A4988.

La señal *Enable* será envía directamente al Driver A4988 para habilitar o deshabilita el movimiento de la banda, la señal Servo_on será envía a la FMS de los Servomotores para formar parte de su control, de igual forma esta señal funcionara como habilitación

para este proceso y la señal *Step* será enviada también al Driver A4988 en forma de pulsos continuos hasta que la señal *Enable* interfiera en esta.

3.5.4 Servomotores MG995

Los actuadores utilizados en el brazo robótico serán servomotores de la marca Tower Pro. En los eslabones se utilizara el modelo MG995 y para el actuador final o *gripper* el modelo MG90S ambos tienen un ángulo de movimiento de 180°, para los servomotores que darán movimiento a cada eslabón se controlara su posición y velocidad, empezando con el segundo parámetro, en la Figura 45 se muestra el actuador utilizado en los eslabones.



Figura 45. Servomotor de la marca Tower Pro Modelo MG995.



En la Tabla 6 se muestran las características de los servomotores MG995.

Tabla 6. Características Servomotor MG995.

Especificaciones	MG995
Peso	55 gr.
Dimensiones	40.7 x 19.7 x 42.9 mm
Torque	8.5 kgf x cm (4.8 volts) - 10 kgf x cm (6 volts)
Velocidad de operación	0.2 s/60° (4.8 volts) – 0.16 s/60° (6 volts)
Angulo de rotación	180°
Voltaje de operación	4.8 volts a 7.2 volts
Temperatura de operación	0 °C a 55 °C

Internamente un servo frecuentemente consta de un mecanismo reductor. Por tanto, proporcionan un alto par y un alto grado de precisión (incluso décimas de grado). Los servos se admiten una tensión de alimentación entre 4.8 volts a 7.2 volts, siendo el valor más adecuado 6 volts. Con tensiones inferiores el motor tiene menos fuerza y velocidad. Con tensiones superiores a 6.5 volts los servos empiezan a oscilar demasiado, lo cual los hace poco útiles.

Internamente un servo está constituido por un motor de corriente continua, acoplado a un reductor para reducir la velocidad de giro, junto con la electrónica necesaria para controlar su posición en la Figura 46 se muestra los componentes internos de los servomotores utilizados como actuadores.



Figura 46. Componentes internos de servomotor.

Se dispone de un potenciómetro unido al eje del servo, que permite al servo conocer la posición del eje. Esta información es procesada por un controlador integrado que se encarga de ajustar el movimiento del motor para alcanzar la posición deseada.

3.5.4.1 Velocidad angular

El servomotor MG995 cuenta con tres cables para su conexión estos son identificables por un respectivo color según su uso, para su alimentación a 6 volts se usa el cable rojo (Vcc) y café (GND), el cable amarillo es para la señal PWM o ancho de pulso que controlara su posición, por este cable se envía una señal con un ancho variable, este ancho deberá de ser como mínimo de 0.5 ms hasta un máximo de 2.5 ms y deberá ser enviado cada 20 ms, esto quiere decir que para posicionar el ángulo de giro del servomotor a 0° se deberá enviar un pulso de 0.5 ms de duración en cambio si se quiere una posición de 180° el pulsos deberá de ser de 2.5 ms, los ángulos intermedios se conseguirán variando el tiempo de duración del ancho de pulso mínimo

y el máximo descrito anteriormente, en la Figura 47 se muestra tres señales enviadas con la respectiva ubicación del servomotor.

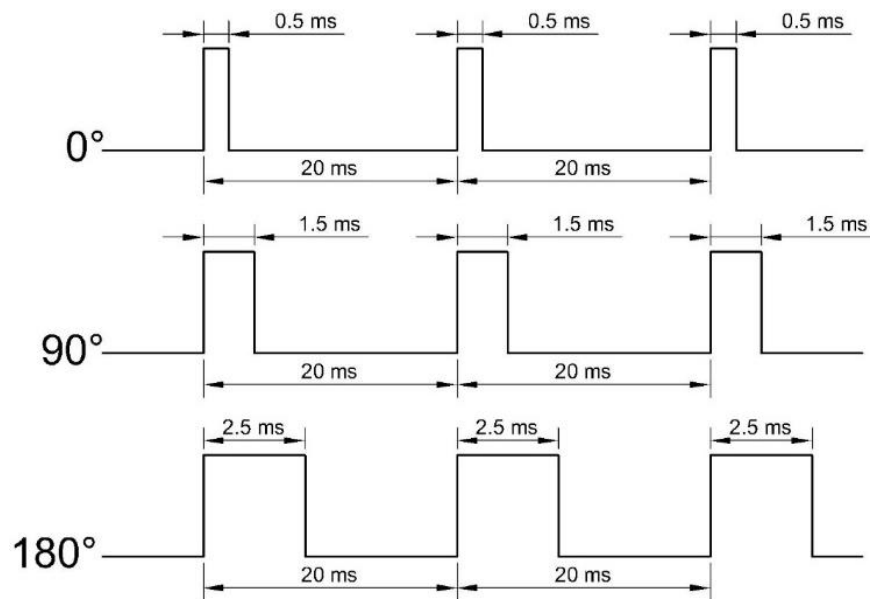


Figura 47. PWM y sus respectivas posiciones angulares para el servomotor.

Si se cambia directamente un ancho de pulsos de 0.5 ms a 2.5 ms para un servo alimentado a 6V, este girara a una velocidad de $0.16 \text{ s}/60^\circ$. Esto se considera como una alta velocidad de operación y provocaría fallos y errores en el mecanismo final. Por lo tanto, la velocidad a la que se deberá mover el servomotor para aplicar un movimiento suave y preciso será directamente proporcional a la velocidad con la que se incremente el ancho de pulso.

Tomando en cuenta los datos anteriores se sabe que el pulso mínimo sería 0.5 ms y el máximo de 2.5 ms, la diferencia entre ambos es de 2 ms este es el rango de tiempo total con el que trabajan esta clase de actuadores, se aplica la fórmula mostrada en la ecuación 31 para obtener la frecuencia de trabajo.



$$f = \frac{1}{t} = \frac{1}{0.002 s} = 500Hz \quad (31)$$

Con referencia en este resultado se sabe que para que el servomotor gire a una velocidad de 1 s/180° se deberán incrementar o decrementar el valor del ancho de pulsos cada 500 Hz.

En base a lo anterior se muestra en la Tabla 7 la relación entre la velocidad angular (ω), frecuencia (f), y su valor en sistema binario (*Bin*).

Tabla 7. Relación para la velocidad, frecuencia y su valor en sistema binario.

Núm.	ω	f	<i>Bin</i>
1	1 s/180°	500 Hz	"00111110100"
2	1.5 s/180°	750 Hz	"01011101110"
3	2 s/180°	1,000 Hz	"01111101000"
4	2.5 s/180°	1,250 Hz	"10011100010"
5	3 s/180°	1,500 Hz	"10111011100"
6	3.5 s/180°	1,750 Hz	"11011010110"

Se incluirán en la programación las seis velocidades angulares mostradas en la tabla anterior con el fin de determinar la velocidad que mejor se ajuste al mecanismo, para ser posible ingresar todas las velocidades y que el operador del robot seleccione manualmente por medio de interruptores la velocidad a la que trabajara el robot se implementara en la programación un multiplexor, estos circuitos son de mucha importancia dentro de la electrónica digital su principio básico es el paso selectivo de una o varias señales hacia uno o varios destinos, en la Figura 48 se muestra el multiplexor diseñado.

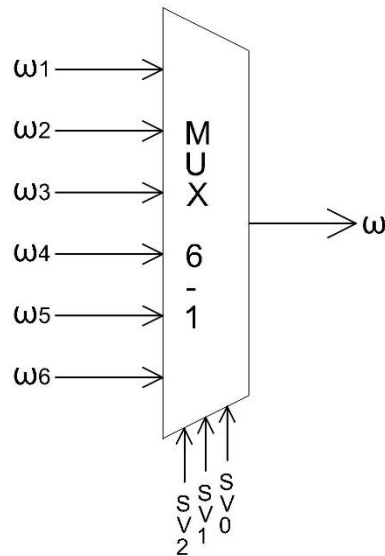


Figura 48. Multiplexor diseñado para controlar la velocidad del servomotor.

Al describir el multiplexor Mux 6-1 en VHDL se deberán sustituir las velocidades angulares por sus respectivos valores numéricos en binario como se muestra en la Tabla 8.

Tabla 8. Asignación numérica para cada velocidad.

SV_2	SV_1	SV_0	<i>Bin</i>
0	0	1	"00111110100"
0	1	0	"01011101110"
0	1	1	"01111101000"
1	0	0	"10011100010"
1	0	1	"10111011100"
1	1	0	"11011010110"

En la Figura 49 se muestra el diagrama de bloques final para el control de velocidad de los servomotores, la señal SV constara de 3 bits, esta se controlara directamente

de los interruptores que vienen incomparados en la tarjeta FPGA con esto el usuario podrá variar la velocidad del robot incluso cuando este se encuentre realizando una trayectoria, la salida Vel será de un bit y será una señal variable que dependerá directamente de la entrada de tres bits SV. En conclusión, Vel será la que determine a qué velocidad se incrementará el ancho de pulso, esta señal será enviada a la FMS de los servomotores para formar parte de su control y complementar el PWM de cada actuador.

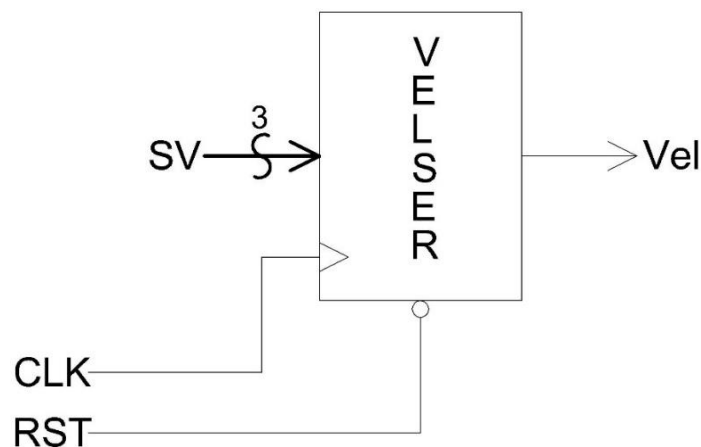


Figura 49. Diagrama de bloques para la velocidad de los servomotores.

3.5.4.2 Control de Servomotor MG995

El control de los servomotores se realizará por medio del FPGA, en la programación se generaran seis PWM para cada uno de los servomotores y la alimentación de voltaje se realizara por medio de una fuente externa de 5 volts y 28A ya que el FPGA no tiene la capacidad de aportar la corriente necesaria para su función. Otro punto de importancia es que una fuente externa ayuda a proteger la tarjeta FPGA de corto

circuitos, estas tarjetas tienen un sistema interno de protección para cuando sus salidas sobrepasan la corriente máxima que pueden suministrar, este sistema consiste en básicamente desprogramarse logrando así la deshabilitación de sus salidas y cortando el flujo de corriente.

El control de los servomotores se realizara por dos programas el primero llamado Ctrl_A será el encargado de controlar la salida PWM, y la velocidad, este parámetro fue mencionado en la subsección anterior, igualmente en este programa se generaran las señales con las que se determinara la posición en la que se encuentra cada servomotor estas señales serán de gran utilidad y serán explicadas a detalle más delante de la presente subsección, el segundo programa llamado Ctrl_B controlara la habilitación de los servomotores, la dirección de giro y mandar los anchos de pulso al programa Ctrl_A para complementar el control del PWM, como se verá más adelante cada programa se complementa uno a otro al enviar y recibir señales simultáneamente, ejemplificado en la Figura 50.

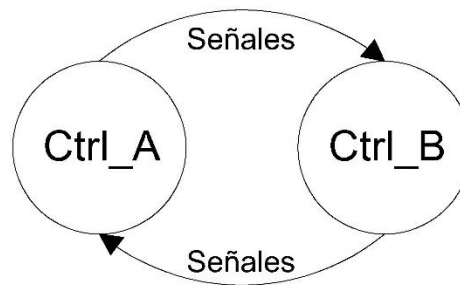


Figura 50. Diagrama para el control de Servomotores.

En esta subsección se analizará a detalle la lógica que se utilizó para el control de los servomotores.



En la Tabla 9 se describen las señales que son generadas y enviadas por el programa Ctrl_A.

Tabla 9. Definición de señales para el programa Ctrl_A.

Ctrl_A	
Señales	Definición
PWM_A, PWM_B, PWM_C, PWM_D, PWM_E, y PWM_F	Estas señales son generadas por el programa Ctrl_A y serán establecidas como salidas del FPGA para ser conectadas a los pines de los servomotores, en estas señales se enviará el ancho de pulso que definirá el ángulo en el que se posicionaran cada uno de los servomotores, cada señal PWM corresponde a un servomotor.
SA, SC, SD, SE y SF	Estas señales son generadas por el programa Ctrl_A a partir de las señales “M” mencionadas en la tabla siguiente y enviadas al programa Ctrl_B para monitorear la posición en la que se encuentran los servomotores en tiempo real.
ZA y ZE	Esta señal es generada por el programa Ctrl_A a partir de las señales “M” mencionadas en la tabla siguiente y enviada al programa de control de la banda transportadora con el fin de establecer comunicación entre ambos mecanismos, su funcionamiento será explicado en el programa de control para las guías de objeto a clasificar.



En la Tabla 10 se describen las señales que son generadas y enviadas por el programa Ctrl_B.

Tabla 10. Definición de señales para el programa Ctrl_B.

Ctrl_B	
Señales	Definición
TA, TB, TC, TD, TE y TF	Estas señales son generadas por el programa Ctrl_B y enviadas al programa Ctrl_A para definir el sentido de giro de los servomotores, solo tendrá dos estados “10” y “01”, cuando la señal este en “10” los servomotores giraran en sentido de las manecillas del reloj y cuando este en “01” girara en sentido contrario, en los estados faltantes el motor deberá estar en reposo.
M1, M2, M3, M4, M5 y M6	Estas señales son generadas por el programa Ctrl_B y enviadas al programa Ctrl_A y contendrá el ancho de pulso para cada PWM, la señal M1 corresponderá al PWM_A, la señal M2 corresponderá al PWM_B y así consecutivamente hasta la señal M6 que corresponderá al PWM_F, de esta señal dependerá las señales S que serán generadas por el programa Ctrl_A.

Como se mencionó en la tabla y en las subsecciones anteriores algunas señales serán enviadas a otros programas de control con la finalidad de establecer comunicación entre ellos y así realizar un mejor control de todo el sistema.

En la Figura 51 se muestra el diagrama de bloques para el control de los servomotores.

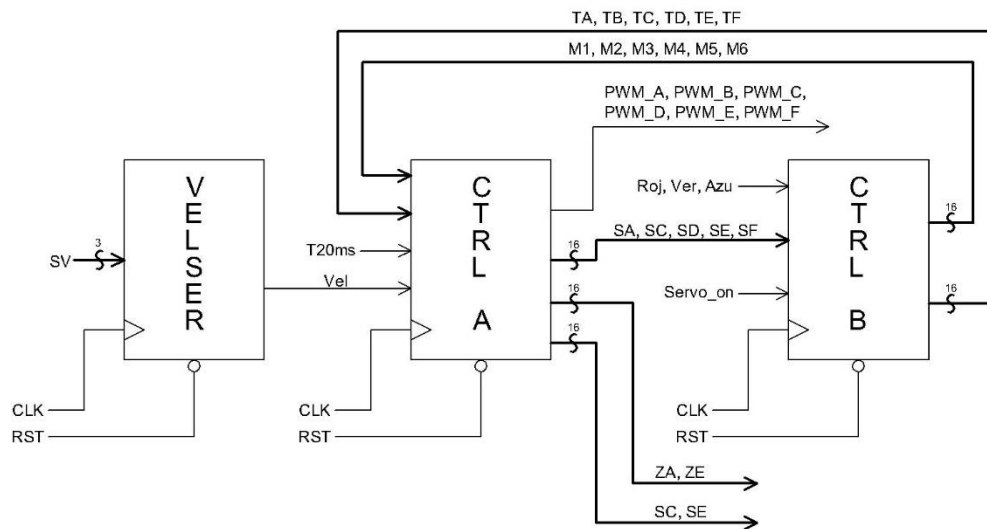


Figura 51. Diagrama de bloques para el control de servomotores.

En el diagrama de bloques para el control de servomotores se complementó con el diagrama de bloques para la velocidad de los mismos, el cual fue mencionado en el subcapítulo anterior y tiene gran importancia en el control general de los servomotores.

El funcionamiento para este diagrama es el siguiente, como se mencionó anteriormente los servomotores tienen un rango de movimiento de 180° controlados por un ancho de pulso mínimo de 0.5 ms para 0° y máximo de 2.5 ms para 180° , obteniendo los ángulos intermedios variando estos tiempos. Pasando el tiempo mínimo y máximo a frecuencia y relacionándolo con el reloj interno del FPGA nos queda la ecuación 30.

Para 0.5 ms

$$f = \frac{1}{t} = \frac{1}{0.0005 \text{ s}} = 2000 \text{ Hz} \quad \therefore \quad c = \frac{50 \text{ MHz}}{2000 \text{ Hz}} = 25000 \quad (32)$$



Para 2.5 ms nos queda la ecuación 31.

$$f = \frac{1}{t} = \frac{1}{0.0025 \text{ s}} = 400 \text{ Hz} \quad \therefore \quad \frac{50 \text{ MHz}}{400 \text{ Hz}} = 125000 \text{ Hz} \quad (33)$$

Por lo tanto, la frecuencia mínima sería de 25000 Hz para 0° y la máxima de 125000 Hz para 180°, en la Tabla 11 se muestra la relación entre la frecuencia y el ángulo de movimiento realizando incrementos de 5000 Hz.

Tabla 11. Relación entre la frecuencia y el ángulo de movimiento.

INCREMENTOS	FRECUENCIA HZ	ÁNGULO
1	25000	0°
2	30000	9°
3	35000	18°
4	40000	27°
5	45000	36°
6	50000	45°
7	55000	54°
8	60000	63°
9	65000	72°
10	70000	81°
11	75000	90°
12	80000	99°
13	85000	108°
14	90000	117°
15	95000	126°
16	100000	135°
17	105000	144°
18	110000	153°
19	115000	162°
20	120000	171°
21	125000	180°

En la programación en VHDL se trabajará en todo momento con la frecuencia, es por eso que se realizó la tabla anterior para tener como referencia el ángulo al que se

desplazara el eje del servomotor con respecto de los hertz que se incrementen o disminuyan según sea la secuencia de movimiento programado.

Para complementar la explicación en la Figura 52 se ilustrara la identificación de los servomotores colocados en el brazo robótico.

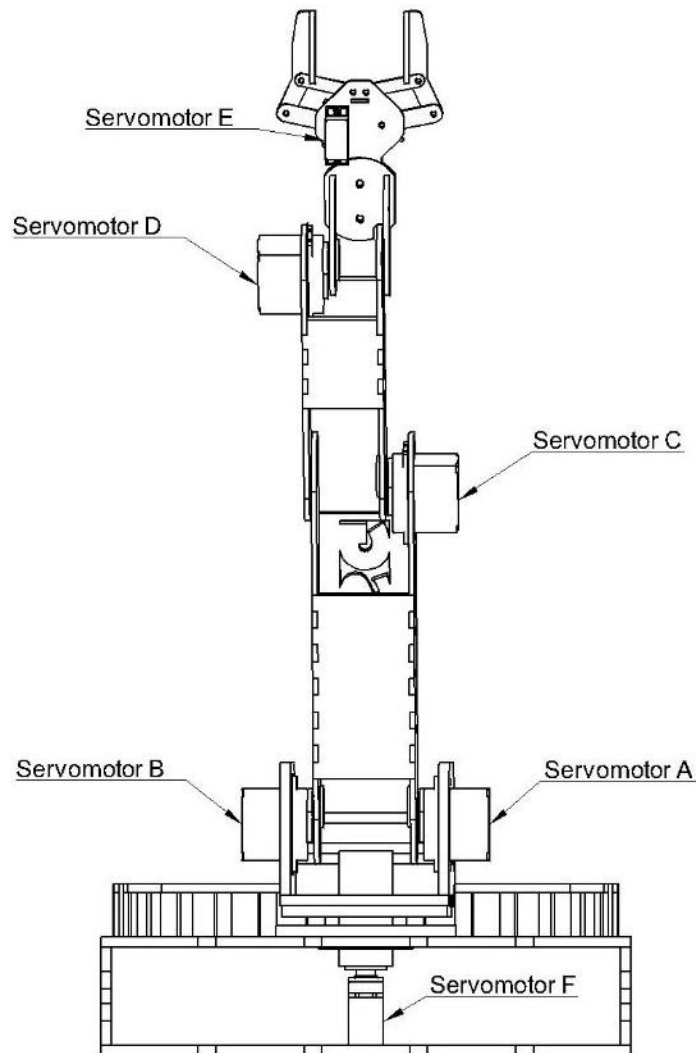


Figura 52. Identificación de Servomotores.

Una vez cargado el programa en el FPGA y antes de habilitar la alimentación externa de los servomotores se realizará un reset general del sistema con la finalidad



de enviar todos los programas a su origen con esto también obtendremos que el brazo robótico se situé en su origen o HOME, el *reset* se realizara en el programa Ctrl_A.

La señal encargada de iniciar con la secuencia de movimiento será servo_on esta señal está en modo de entrada para el programa Ctrl_B y proviene de la señal D0_A emitida por el sensor TCRT5000 anteriormente se explicó la función que tendría en la habilitación de los servomotores. Una vez que se recibió esta señal el programa Ctrl_B verifica que el robot este en HOME, esto lo realiza por medio de las señales SA, SB, SC, SD, SE y SF asignadas respectivamente a las señales PWM_A, PWM_B, PWM_C, PWM_D, PWM_E y PWM_F, las señales iniciadas con el prefijo S generadas en el programa Ctrl_A envía la posición en la que se encuentran los servomotores en tiempo real, de igual forma dos de estas señales SC y SE serán enviadas a la programación del sensor de color RGB para identificar cuando el robot se encuentre en la posición en la que el sensor se deberá activar. Una vez que se han cumplido las condiciones anteriores el programa Ctrl_B le enviara simultáneamente las señales TA, TB, TC, TD, TE, TF, M1, M2, M3, M4, M5 y M6, al programa Ctrl_A, las señales iniciadas con el prefijo T controlaran la dirección de giro de cada servomotor y la señales M enviaran el nuevo ancho de pulso para cada uno de ellos, como se menciona anteriormente las señales T serán de dos bits, cuando la señal este en “10” el servo girara en sentido horario y cuando este en “01” girara en sentido contrario, en los demás estados el servomotor permanecerá en la misma posición. En la Tabla 12 se muestra el ancho de pulso correspondiente al HOME del brazo robótico y la nueva posición en la que se ubicaran los servomotores con respecto de las señales M, las señales T determinan la dirección de giro, explicado de otra forma, la señales T determinan el incremento o decremento de la frecuencia y las señales M serán el nuevo ancho de pulso en el que se posicionaran los servomotores.

Tabla 12. Función de la señal T y señal M.

PWM	HOME	TA	TB	TC	TD	TE	TF	SEÑAL M
PWM_A	25000	"10"						65000
PWM_B	100000		"01"					60000
PWM_C	45000			"10"				95000
PWM_D	115000				"01"			70000
PWM_E	35000					"10"		75000
PWM_F	80000						"10"	125000

En la tabla anterior solo se describe dos posiciones, esto con el fin de dar a entender el funcionamiento de la secuencia de los servomotores. Sin embargo, en la programación en VHDL se cuentan con 15 posiciones aproximadamente dependiendo del color del objeto a clasificar para realizar la secuencia completa del brazo robótico.

En la Figura 53 se muestra la ubicación que tendrá el brazo robótico correspondiente a la posición de *home*. Por otro lado, en la Figura 54 se muestra la nueva posición en la que se ubicara el brazo con respecto de la señal M descrita en la Tabla 12.

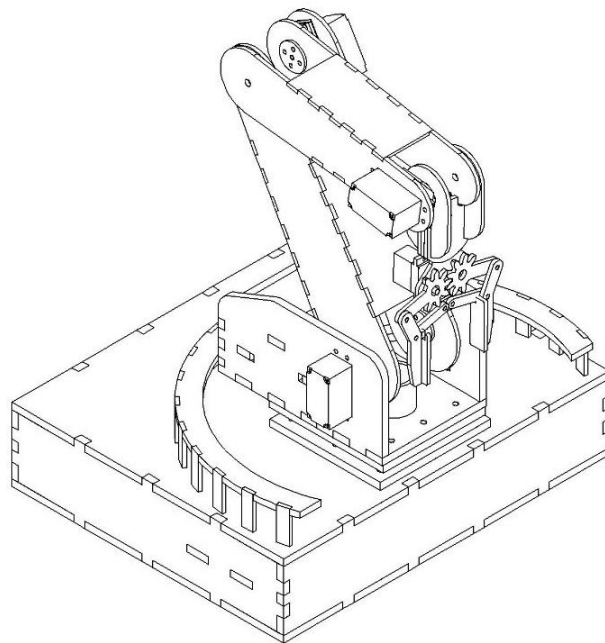


Figura 53. Brazo robótico posicionado en Home.

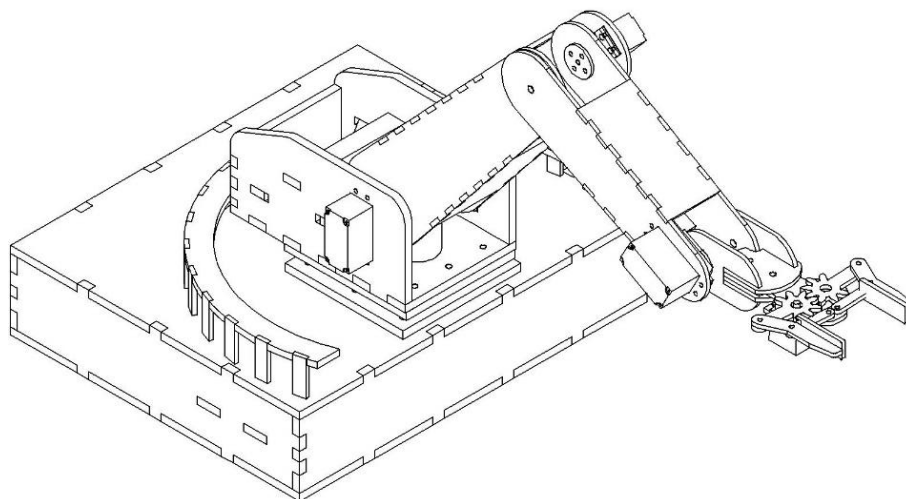


Figura 54. Nueva secuencia de movimiento con respecto de la señal M

Por medio de las señales T, S y M el sistema realizará todas las secuencias de movimientos del brazo robótico, sin embargo habrá señales enviadas por otros procesos que influirán directamente en esta secuencia, por ejemplo los sensores TCRT5000 y el sensor TCS3200 que tendrá un papel de gran importancia en el cumplimiento del objetivo general del proyecto y su funcionamiento se describirá en la subsección siguiente.

3.5.5 Sensor TCS3200

En la Figura 55 se muestra el sensor TC3200 el cual será el encargado de identificar los colores de los objetos a clasificar. Este sensor es del tipo óptico y permite detectar el color de un objeto ubicado en frente de él. Internamente, el TCS3200 está formado por una matriz de fotodiodos de silicón junto con un conversor de frecuencia, en un único integrado CMOS. La matriz dispone de 8 x 8 fotodiodos de 110 μm , de los cuales 16 tienen filtros azules, 16 verdes, 6 rojos, y 16 no tienen filtro. Los fotodiodos están distribuidos de forma que minimizan el efecto de la incidencia no uniforme de la luz.

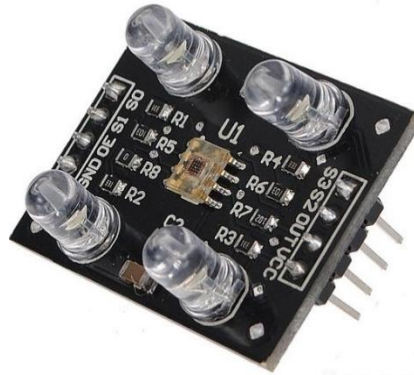


Figura 55. Sensor TCS3200.

La salida del TCS3200 es una onda cuadrada del 50% *duty*, esto quiere decir que permanecerá el 50% de un ciclo en estado alto y el otro 50% en estado bajo, esta frecuencia será proporcional al tipo de color RGB que se esté analizando y a la intensidad luminosa que se refleje en el sensor, en Figura 56 la se muestra el diagrama general.

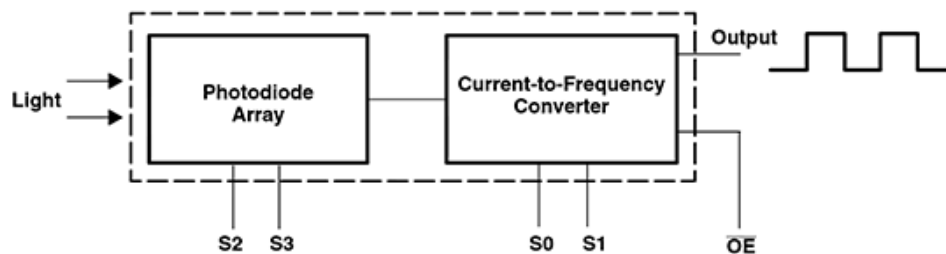


Figura 56. Diagrama general sensor TCS3200.

El sensor TCS3200 cuenta con 8 pines para su conexión, de los cuales Vcc y GND son para su alimentación a 5 volts, los pines S0 y S1 conforman una señal de dos bits para seleccionar la frecuencia en su salida. Los pines S2 y S3 de igual forma conforman una señal de dos bits y se utilizaran para seleccionar el filtro con el que se van a analizar los objetos a clasificar, el pin OE se utilizara para habilitar o deshabilitar el sensor y el pin OUT es la salida del sensor que se conectara al FPGA donde se



procesara esta señal y, posteriormente se determinará el color del objeto que se haya analizado en base a la frecuencia recibida.

En las Tabla 13 y Tabla 14 se muestran las configuraciones de los pines S0 y S1 para la escala de la frecuencia de salida y los pines S2 y S3 para la selección del filtro con el que se va a analizar los objetos respectivamente.

Tabla 13. Configuración para la escala de la frecuencia.

S0	S1	ESCALA PARA FRECUENCIA DE SALIDA
0	0	Sensor apagado
0	1	2%
1	0	20%
1	1	100%

Tabla 14. Configuración para la selección de filtro.

S2	S3	SELECCIÓN DE FILTRO
0	0	Rojo
0	1	Azul
1	0	Sin filtro
1	1	Verde

El análisis de color para cada objeto se realizará en un segundo para cada filtro y su frecuencia de salida será escalada al 2%, esto quiere decir que un objeto será analizado por los tres filtros con una duración de un segundo para cada filtro independientemente cual sea su color. Posteriormente se guardarán los tres valores obtenidos por cada filtro para ser comparados entre si al final del análisis y la frecuencia más alta obtenida será la que coincida entre el color del filtro y el color del objeto. Por ejemplo, si se analiza un objeto de color rojo, el filtro rojo dará como resultado la frecuencia más alta entre los tres filtro, lo mismo pasara con los filtros de color azul y verde para objetos de color azul y verde respectivamente, en la Figura 57

se muestra el diagrama de bloques para la programación del sensor TCS3200 en VHDL.

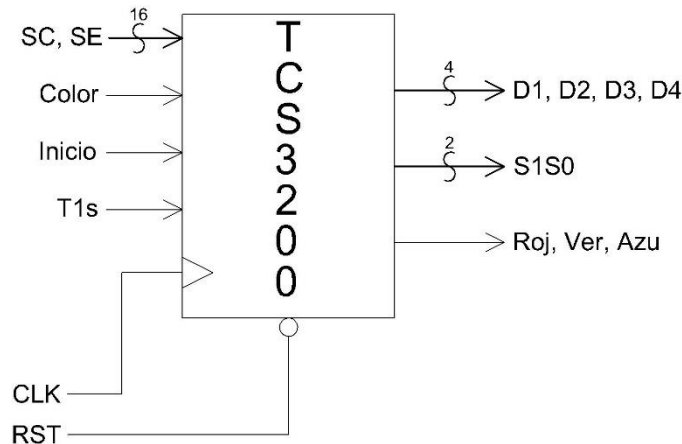


Figura 57. Diagrama de bloques para el control del sensor RGB.

A continuación, se describen el funcionamiento del diagrama de bloques para el control del sensor RGB.

Las señales SC y SE serán enviadas por la programación de los servomotores, estas señales como se explicó anteriormente envían la posición en la que se encuentran los servomotores en tiempo real, por lo tanto, por medio de estas señales se activará el sensor RGB cuando el robot este posicionado frente al objeto a analizar. La señal llamada Color proviene del sensor RGB, y será la frecuencia con la que se determinará el color de los objetos a analizar, la señal llamada T1s proviene de un contador generado en la programación para las bases de tiempo que se utilizarán en todos los programas, esta señal se activará cada segundo y será la encargada de activar el contador para los pulsos recibidos por la señal Color en este caso el conteo se realizará por medio de un segundo para cada filtro del sensor, la señal Inicio será utilizada para reiniciar la cuenta mostrada en el display de 7 segmentos disponible en el FPGA, se utilizará esta interfaz para monitorear el correcto funcionamiento del sensor RGB, cada que se realice una lectura con alguno de los tres filtros se mostrará

el resultado en el display, cuando se inicie la siguiente cuenta se restablecerá la cuenta para mostrar el siguiente resultado así hasta completar el proceso, la señal S1S0 será la encargada de cambiar el filtro con el que se analizara el objeto a clasificar. En la Tabla 14 se describen las combinaciones para cada filtro, esta señal estará cambiando de valor con respecto con la señal T1s, las señales Roj, Ver y Azu enviaran la información respecto al color del objeto a clasificar. Estas señales serán enviadas al programa de control de los servomotores para que el robot realice la secuencia de movimiento correspondiente al color del objeto que se analizó. Por último, las señales D1, D2, D3, y D4 enviarán la información del conteo registrado por cada filtro del sensor, estas señales serán enviadas al programa de control para el display de 7 segmentos descrito en la subsección siguiente.

3.5.6 Display 7 Segmentos

El dispositivo FPGA Nexys 2 que se utilizara en este proyecto dispone de un *display* de 7 segmentos de ánodo común con 4 dígitos como el mostro en la Figura 58.

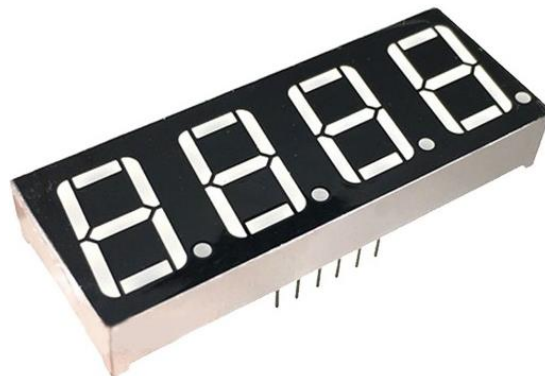


Figura 58. Display 7 Segmentos de cuatro dígitos.

Este *display* es del tipo multiplexado, es decir, que sus segmentos están conectados internamente de tal manera que al exterior sólo aparecen las conexiones correspondientes a los siete segmentos y en este caso cuatro conexiones para cada uno de sus ánodos, en la Figura 59 se muestra la distribución de sus conexiones internas.

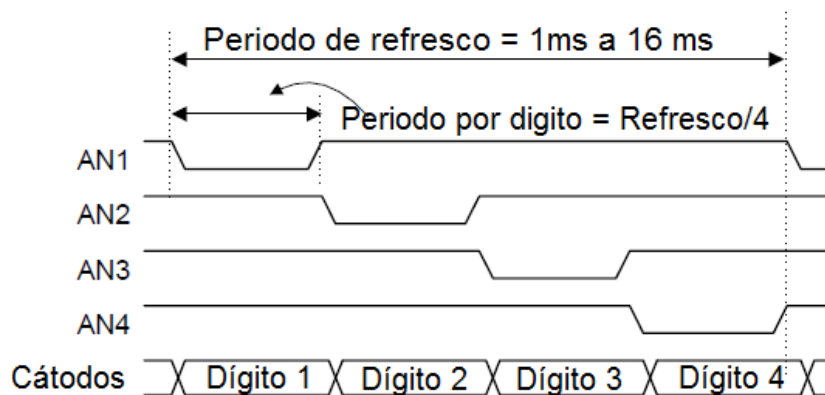


Figura 59. Display de 7 segmentos Nexys 2.

La característica principal de los display de 7 segmentos multiplexados es que los 7 pines para su conexión a los cátodos están compartidas entre todos los dígitos, para este caso son cuatro, por lo tanto el control se realiza por medio de los ánodos en la Figura 60 se muestra el diagrama su conexión interna.

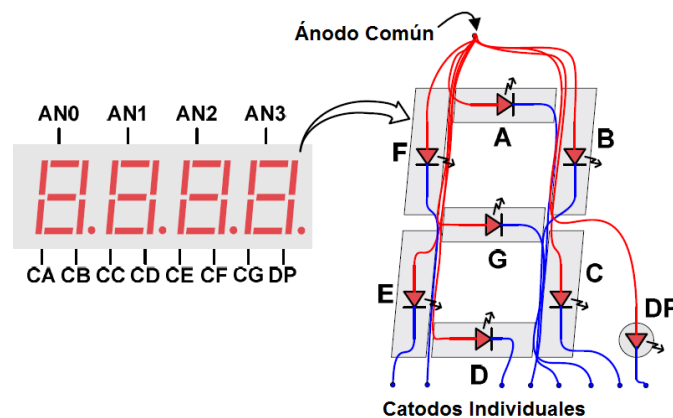


Figura 60. Diagrama de sincronización para display de 7 segmentos.



En el programa de control para el display de 7 segmentos implementado en FPGA se enviara una señal cada 4 ms para cada uno de los ánodos para completar un periodo de refresco de 16 ms entre los cuatro dígitos, cada digito se activara cuando en su ánodo se reciba un 0 lógico en la Tabla 15 se muestra la configuración para la activación de los cuatro dígitos.

Tabla 15. Configuración para la activación de los ánodos.

A	B	C	D	E	F	G	NÚMERO
0	0	0	0	0	0	1	0
1	0	0	1	1	1	1	1
0	0	1	0	0	1	0	2
0	0	0	0	1	1	0	3
1	0	0	1	1	0	0	4
0	1	0	0	1	0	0	5
0	1	0	0	0	0	0	6
0	0	0	1	1	1	1	7
0	0	0	0	0	0	0	8
0	0	0	1	1	0	0	9

Por lo tanto, si se quiere mostrar un carácter en el digito número uno se deberá enviar la con configuración deseada a los cátodos y poner la entrada de los ánodos en 1110 por 4 ms y así para cada uno de los dígitos restantes.

En la Tabla 16 se muestra la configuración de los cátodos para mostrar una numeración del 0 al 9.

Tabla 16. Configuración de cátodos para numeración decimal.

ÁNODOS	ACTIVACIÓN
Ánodo 1	1110
Ánodo 2	1101
Ánodo 3	1011
Ánodo 4	0111

En la Figura 61 se muestra es diagrama de bloques para la programación del display de 7 segmentos.

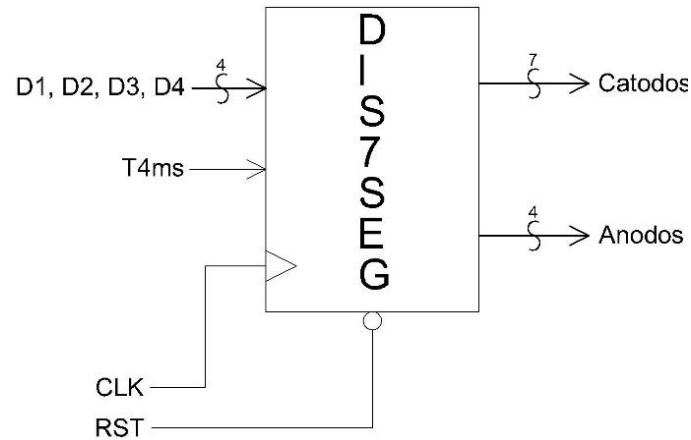


Figura 61. Diagrama de bloques para la programación del display de 7 segmentos.

Las señales D1, D2, D3, y D4 serán enviada por el programa de control del sensor RGB explicado anteriormente, cada una de estas señales corresponde a cada uno de los dígitos y al ánodo. Por medio de estas señales se enviará la cuenta de los pulsos recibidos por el sensor RGB y se expresara en sistema decimal, la señal T4ms se generara en el programa para las bases de tiempo que se utilizaran en distintas partes de la programación en el subcapítulo siguiente se explicara cada una de las bases de tiempo generadas y cuál será su utilización en los distintos programas.

Las señales de salida cátodos y ánodos mostrados en el diagrama de bloques de la Figura 61 serán de siete y cuatro bits respectivamente y se asignaran directamente a las entradas del FPGA para el display de 7 segmentos.



3.5.7 Divisores de frecuencia

Los divisores de frecuencia implementados en VHDL son componentes simples, cuyo objetivo se basa en reducir su frecuencia de entrada, estos divisores se implementan con ayuda del factor de escalamiento y un contador. Primeramente, el factor de escalamiento es la relación entre la frecuencia de entrada y la frecuencia de salida deseada, el FPGA utilizado en este proyecto cuenta con un reloj interno de 50 MHz, por lo cual es necesario realizar varios divisores de frecuencia según las características de cada programa de control, como se mencionó anteriormente se generaron tres divisores de frecuencia para distintos programas de control los cuales son:

T_20ms: Este divisor de frecuencia es utilizado para el control de los servomotores, específicamente en la señal PWM, esta señal debe de ser enviada al servomotor cada 20 ms con esto se genera un refresco en la señal con la que el servomotor podrá distinguir un aumento de decremento en el ancho de pulso.

T_4ms: Este divisor de frecuencia es utilizado en el control del display de 7 segmentos mencionado en el subcapítulo anterior, este divisor tiene la tarea de realizar un refresco en los cuatro dígitos del display, teniendo un total de 16 ms para todo el proceso.

T_1s: Este divisor de frecuencia es utilizado en el control del dispositivo RGB, por medio de este divisor de frecuencia el sensor realizara una medición por un segundo para cada uno de los filtros, con esto se obtienen resultados precisos reflejándose directamente en la correcta identificación y clasificación de los colores primarios.

En la Figura 62 se muestra el diagrama de bloques para los divisores de frecuencia.

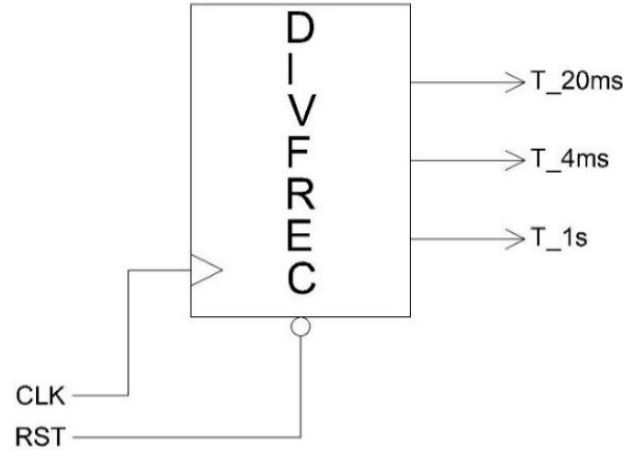


Figura 62. Diagrama de bloques para control de divisores de frecuencia.

Como se observa en el diagrama, para esta programación no existirán señales de entrada, si no que se utilizara el reloj interno del FPGA y simplemente se hará un escalamiento de este, así obteniendo los tres divisores de frecuencia mostrados en la imagen anterior.

Capítulo 4

Pruebas y Resultados

4.1 Pruebas

Todos los sensores y actuadores fueron probados al momento de realizar sus respectivos códigos de programación empezándonos con los actuadores del brazo robóticos, para determinar que los servomotores están funcionando de manera correcta es necesario realizar la siguiente prueba, en la Figura 63 se muestra el robot ubicado manualmente en una posición aleatoria.

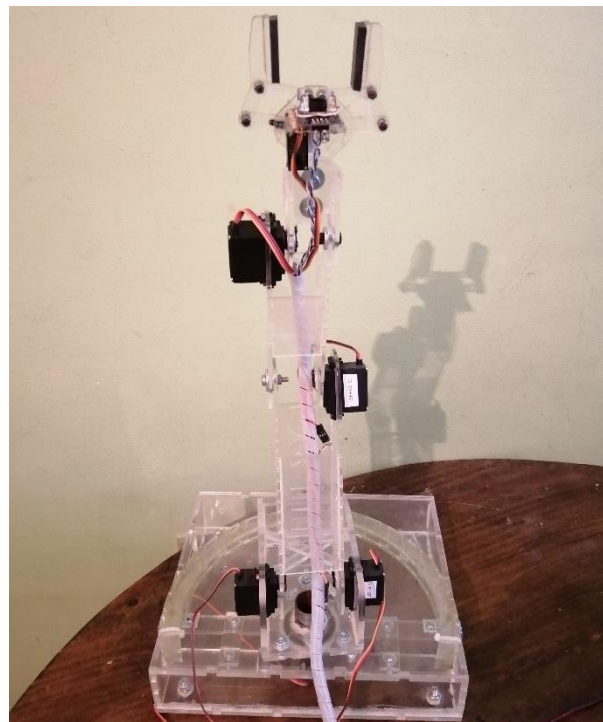


Figura 63. Brazo robótico ubicado en posición aleatoria.

Posteriormente se deberá energizar todos los componentes de este sistema y realizar un *reset* general, para determinar que los actuadores funcionan correctamente el brazo robótico deberá ubicarse en su posición de *home* como se muestra en la Figura 64.

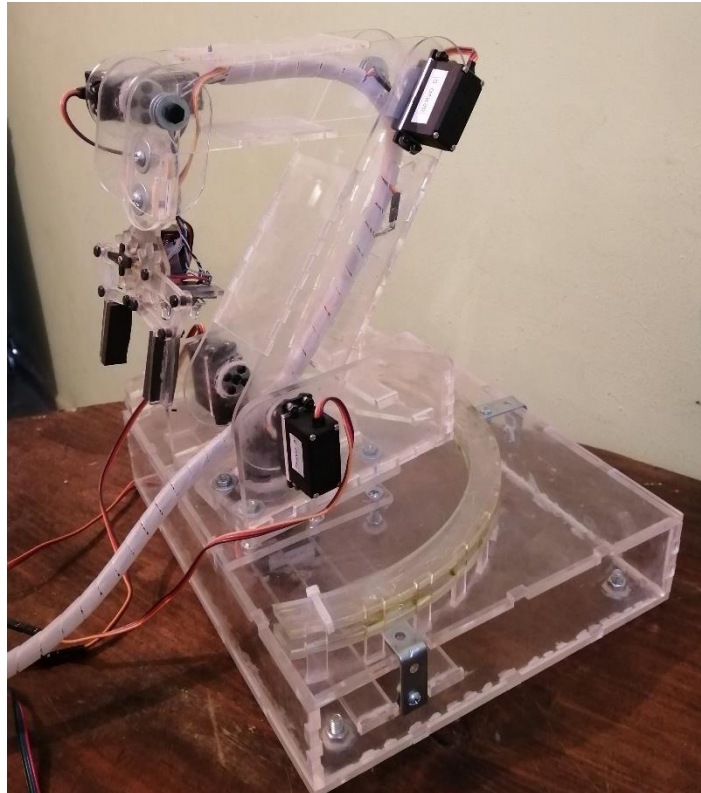


Figura 64. Brazo robótico en su posición de home.

Por medio de este procedimiento se podrá determinar el correcto funcionamiento de los servomotores ya que si alguno no regresa a su posición de home querrá decir que hay una anomalía en su funcionamiento.

Para determinar el correcto funcionamiento del sensor RGB se deberá realizar un escaneo, este módulo se incorporaron 4 leds que tendrás dos funciones en el sistema, la primera es para avisar cuando el modulo inicie el escaneo y la segunda es para iluminar el objeto a escanear con esto se obtendrá un resultado con más precisión al

determinar el color del objeto a clasificar, en la Figura 65 se muestra la prueba para determinar el funcionamiento del sensor RGB.

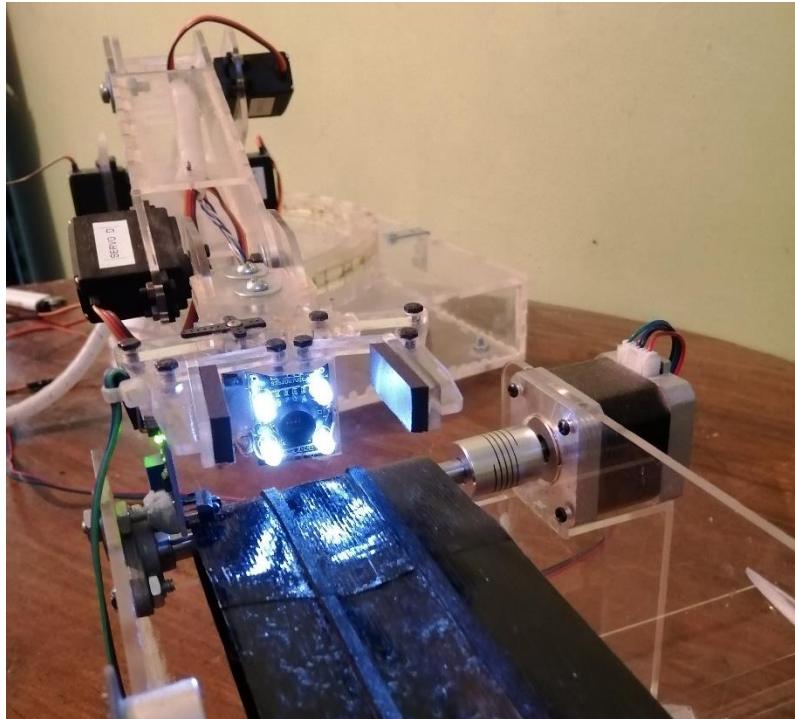


Figura 65. Prueba de funcionamiento del sensor RGB.

El sensor TCRT5000 tendrá tres funciones para el control del sistema, la primera será para controlar la banda transportadora, por medio de este sensor se activara o desactivara el motor a pasos que funcionara como actuador en la banda trasportadora, la segunda función será para iniciar el ciclo de trabajo del brazo robótico y la última función es para activar el contador que determinara el color del objeto, esta cuenta se podrá observa en el *display* de siete segmentos incorporado en el FPGA, en la Figura 66 se muestra el sensor desactivado en espera de un objeto a clasificar para su activación.



Figura 66. Sensor TCRT5000 desactivado.

En la Figura 67 se muestra el sensor activado, este estado se puede confirmar por el led color verde que tiene incorporado el sensor para determinar que se ha detectado un objeto.

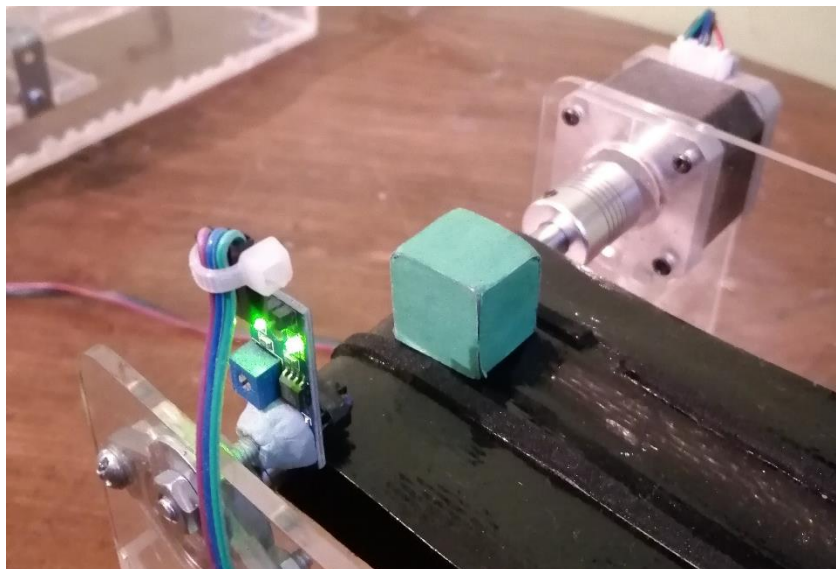


Figura 67. Sensor TCRT5000 activado.

4.2 Resultados

Al realizar las pruebas de todos los componentes los resultados fueron satisfactorios, el sistema del brazo robótico era capaz de detectar, identificar y clasificar los objetos dependiendo de su color

En las siguientes figuras se mostrarán los resultados obtenidos para la clasificación de tres objetos de color rojo, verde y azul.

En la Figura 68 muestra el brazo robótico realizando el escaneo de un objeto de color rojo, este escaneo se realiza por un segundo.

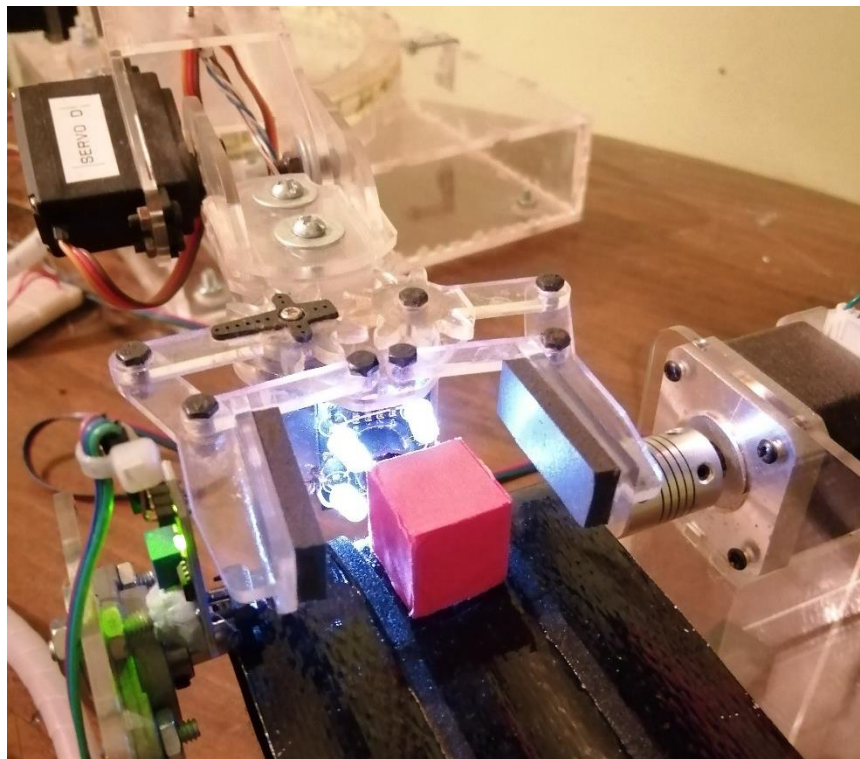


Figura 68. Brazo robótico realizando escaneo de objeto color rojo.

En la Figura 69 se muestra la frecuencia registrada por el escaneo del objeto color rojo, como se mencionó anteriormente, la frecuencia recibida es directamente proporcional al color del objeto a clasificar, por ejemplo para que el sistema determine que el objeto que se escaneo es de color rojo la frecuencia registrada deberá estar entre los 1500 Hertz y 1700 Hertz.

Se debe tener en cuenta las condiciones de iluminación del lugar en donde se esté trabajando ya que influyen considerablemente en los resultados de las frecuencias leídas.

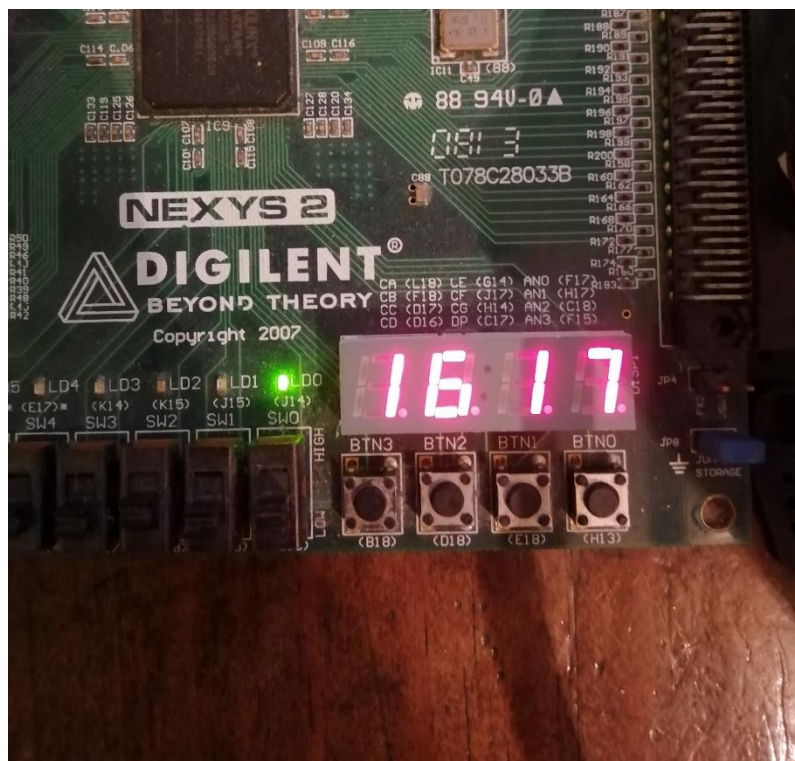


Figura 69. Frecuencia registrada para objeto de color rojo.

En la Figura 70 se muestra al brazo robótico realizando el escaneo del objeto color verde.

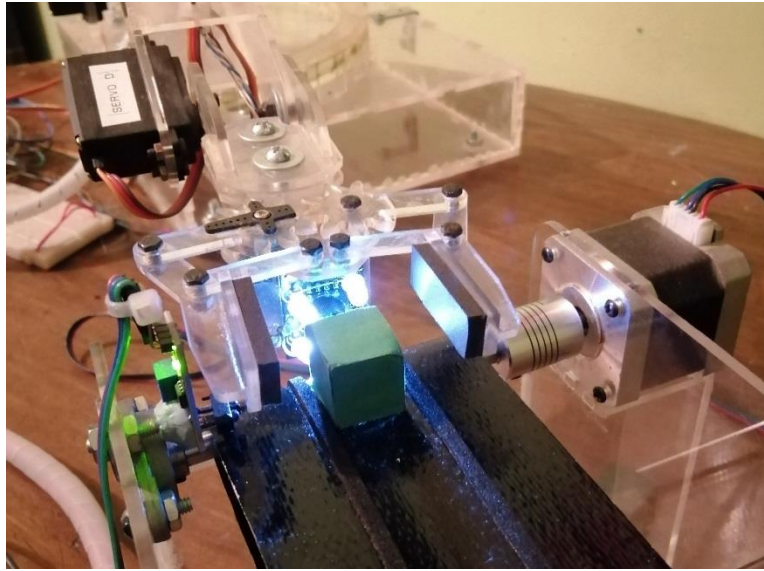


Figura 70. Brazo robótico realizando escaneo de objeto color Verde.

En la Figura 71 se muestra la frecuencia registrada por el objeto de color verde, para que el sistema determine que el objeto que se escaneo es de color verde la frecuencia registrada deberá estar entre los 900 Hertz y 1100 Hertz.

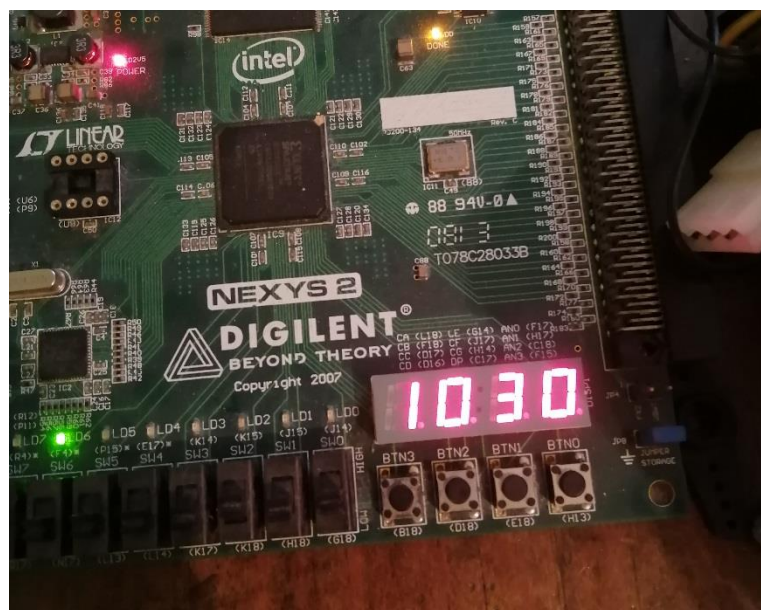


Figura 71. Frecuencia registrada para objeto de color verde.

En la Figura 72 se muestra al robot realizando el escaneo del objeto color azul.

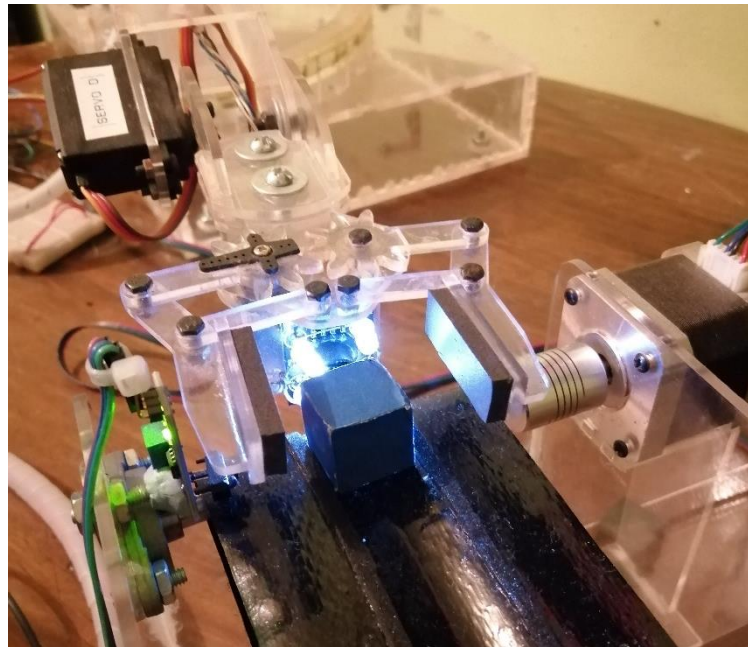


Figura 72. Brazo robótico realizando escaneo de objeto color Azul.

En la Figura 73 se muestra la frecuencia registrada por el objeto de color azul,

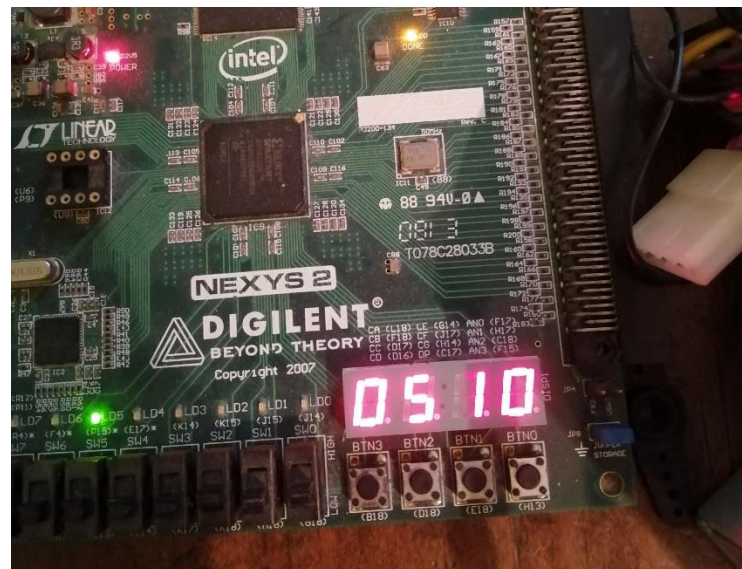


Figura 73. Frecuencia registrada para objeto de color azul.



Por último para que el sistema determine que el objeto que se escaneo es de color azul la frecuencia registrada deberá estar entre los 400 Hertz y 600 Hertz.

Como se mencionó anteriormente un factor que influye de manera directa en los resultados del sensor RGB son las condiciones de iluminación, lo que significa que al trabajar con el robot en un área más o menos iluminada se deberán realizar pruebas y en base en los resultados obtenidos cambiar en la programación los rangos que determinan los colores para cada objeto.



Conclusiones y Recomendaciones

Conclusiones

El dispositivo FPGA utilizado fue capaz de producir, recibir y enviar una gran cantidad de señales simultáneamente de forma rápida y eficiente, todos los actuadores y sensores implementados funcionaron y cumplieron con las expectativas esperadas.

En la Figura 74 se muestra el resumen de los recursos utilizados para el control total del sistema, como se muestra los recursos utilizados no sobrepasan el 20% de los disponibles para una tarjeta FPGA Nexys 2.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	576	9,312	6%	
Number of 4 input LUTs	1,681	9,312	18%	
Number of occupied Slices	908	4,656	19%	
Number of Slices containing only related logic	908	908	100%	
Number of Slices containing unrelated logic	0	908	0%	
Total Number of 4 input LUTs	1,743	9,312	18%	
Number used as logic	1,681			
Number used as a route-thru	62			
Number of bonded IOBs	36	232	15%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	3.17			

Figura 74. Resumen de recursos utilizados de tarjeta FPGA Nexys 2.

En base a la Figura 74 se determina que se podrían agregar más procesos al sistema de control sin ver afectada la capacidad de la tarjeta FPGA utilizada.



Se demostró que los sistemas de control basados en FPGA son una viable y económica alternativa para sustituir sistemas de control basados en dispositivos de más alto costo, obteniendo los resultados esperados en cada uno de los procesos en los que se deseen aplicar.

Uno de los principales motivos de este proyecto fue entender y explicar cómo funcionan los actuadores y sensores que se utilizaron en todo el sistema, y el FPGA presenta una ventaja en este punto ante otras tarjetas de control, las cuales tienen librerías y ejemplo ya diseñados de fácil acceso en internet, en cambio quien trabaje con FPGA desde un principio entenderá el funcionamiento de las señales enviadas y recibidas por la tarjeta hacia cualquier actuador o sensor, en base a esto, considero que este punto se cumplió completamente ya que todos los códigos de programación fueron escritos y entendidos a la par en el desarrollo del proyecto.

Actualmente en la industria no es muy común encontrar sistemas de control basados en FPGA, sin embargo se considera que son una opción viable para introducir a los estudiantes en los sistemas de control digital ya que en toda la industria se trabaja en base a estos sistemas y en todos ellos una parte fundamental es la lógica programable la cual considero como un lenguaje universal que si es dominado y aplicado bajo los correctos estándares industriales, facilitaría el trabajar y entender el funcionamiento del cualquier autómatas programa o bien cualquier otro dispositivo programable.

Recomendaciones

Si bien todos los sensores en el sistema funcionan correctamente se tiene que reconocer que son elementos de bajo costo y por ende su desempeño no será igual a sensores de uso industrial, por tal motivo si se quiere hacer un sistema de control aplicable a algún proyecto industrial se tiene que implementar elementos que den los resultados esperados en las condiciones extremas que la industria demanda.



Una parte fundamental en la exactitud de los movimientos del robot es la calidad de los actuadores, en este proyecto se utilizan servomotores de modelismo sin embargo en el mercado existen actuadores de las mismas dimensiones, pero con componentes internos de mayor calidad, por lo tanto se podría contemplar como mejora cambiar los actuadores por otros de mayor calidad y torque, una opción viable sería el servomotor MG959 de la marca Tower Pro.

La estructura utilizada fue construida en acrílico, este material es bastante frágil para ser utilizado en un brazo robótico de uso industrial por lo tanto una mejora que se podría aplicar al proyecto para ser empleado en un proceso industrial sería construir la estructura en aluminio, este material es idóneo para un brazo robótico de las dimensiones propuestas en este proyecto, además que el aluminio es ligero y resistente.

Los actuadores utilizados en este proyecto requieren de una fuente de alimentación externa que sea capaz de mantener un voltaje constante de 6 volts y un amperaje suficiente para todos los actuadores, es por ellos que es de suma importancia alimentar los servomotores con una fuente con estas características para evitar vibraciones y fallas en la eficiencia de los actuadores.

Para minimizar las variaciones en las lecturas de los colores RGB por las condiciones de iluminación se podría utilizar el sensor RGB TCS34725, este sensor incorpora un filtro de infrarrojos, con esto mejora su precisión ante la luz del entorno. La comunicación del sensor se realiza por medio del protocolo I2C.

Para evitar que el robot sufra posible colisiones se podría modificar el diseño del robot y agregar una articulación extra en el efector final y en ella acoplar un limit switch para detectar si el robot colisiono con algún obstáculo y aplicar un paro en los servomotores para evitar que el robot se dañe.



Referencias

- [1] Andueza L., & Aguirre, I. (2009). Diseño de un manipulador robótico con tres grados de libertad. Ciencia e Ingeniería , 12.
- [2] Barrientos, A., Peñin, L. F., Balaguer, C., & Aracil, R. (1997). Fundamentos de Robótica. Madrid: McGraw-Hill.
- [3] Cipres Torres, E. J. (2016). Uso de un controlador con tecnología FPGA y tiempo real para el scrobot-er 4u. Ciudad de México: Universidad Nacional Autónoma de México.
- [4] Craig, J. J. (2006). Robótica. Edo. de México: Pearson.
- [5] Fu, K. S., Gonzalez, R. C., & Lee, C. S. (1988). Robótica Control, detección, visión e inteligencia. McGraw-Hill.
- [6] Jiménez Hernández, A. (Noviembre de 2015). Desarrollo de un sistema en FPGA para ensamble robotizado guiado por visión. México, D.F.: Universidad Nacional Autónoma de México.
- [7] Llamas, L. (23 de Agosto de 2016). Ingeniería, informática y diseño. Obtenido de www.luisllamas.es
- [8] Llamas, L. (7 de Noviembre de 2017). Ingeniería, informática y diseño. Obtenido de www.luisllamas.es/que-es-una-fpga/
- [9] Luis, E. S., & Morales, E. F. (24 de 08 de 2017). [ccc.inaoep.mx](https://ccc.inaoep.mx/~emorales/Papers/2009/eduardo.pdf). Obtenido de <https://ccc.inaoep.mx/~emorales/Papers/2009/eduardo.pdf>
- [10] M, R., & Idiarte, E. (2002). Robots Industriales Manipulables (1 ed.). Barcelona: UPC.
- [11] Nhivekar , G., & Mudholkar, R. (2012). FPGA based Fuzzy Optoelectronic Color Sensor System. India: Shivaji University.
- [12] Ollero Baturone, A. (2001). Robótica manipuladores y robots móviles. Barcelona: Marcombo.
- [13] Pérez, S. A., Soto, E., & Fernández, S. (2006). Diseño de Sistemas Digitales con VHDL (1 ed.). Madrid: Thomson.



- [14] Reyes Cortés, F. (2011). Robótica control de robots manipulables (1a. ed.). México D.F.: Alfaomega.
- [15] Romero Troncoso, R. d. (2007). Electrónica digital y lógica programable. Guanajuato: Universidad de Guanajuato.
- [16] Somolinos Sánchez, J. A. (2002). Avances en robótica y visión por computador (1 ed.). Cuenca: Universidad de Castilla-La Mancha.



Anexos

Se agrega un CD con los planos del robot, la programación en VHDL y un video demostrativo del sistema completo del robot.