

UACM

Universidad Autónoma
de la Ciudad de México

Nada humano me es ajeno

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS INDUSTRIALES

Construcción de un brazo robótico de uso didáctico

TRABAJO RECEPCIONAL

PARA OBTENER EL TÍTULO DE

LICENCIADO EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS INDUSTRIALES

PRESENTA

VLADIMIR REYES LUNA

DIRECTOR

Dr. Daniel Noriega Pineda

Ciudad de México, Febrero de 2019.

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS[©]

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

Integración del jurado:

Presidente: M.I. Amaranto de Jesús Dávila Jáuregui. San Lorenzo Tezonco

Secretario: Dr. Daniel Noriega Pineda. Casa Libertad

Vocal: Dr. Juan Carlos Martínez Rosas. San Lorenzo Tezonco

Plantel de adscripción:

Casa Libertad

DIRECTOR DE TESIS:

Dr. Daniel Noriega Pineda

Casa Libertad

Agradecimientos:

A mis padres Concepción Luna Cruz (QEPD) y Mario Reyes Bravo, por su cariño y por apoyarme en todo momento, por los valores que me inculcaron, por haberme dado la oportunidad de estudiar esta carrera. Y por ser mi principal ejemplo de vida, honradez y esfuerzo gracias.

A mis hermanos Rene Ulises y Guadalupe Marilyn, por su cariño y apoyo incondicional, durante todo este proceso, por estar conmigo en todo momento gracias.

A mis familiares por darme su incondicional apoyo, comprensión y enseñanzas.

A la Universidad Autónoma de la Ciudad de México por el apoyo otorgado para la impresión y empastado de la tesis.

A los profesores de la Universidad Autónoma de la Ciudad de México por su paciencia, dedicación, apoyo incondicional y por sus valiosos conocimientos que me brindaron a lo largo de mi carrera universitaria.

Agradezco a mi director de tesis Dr. Daniel Noriega, por haberme orientado en los momentos en que necesite de su asesoría, durante el desarrollo y culminación del presente trabajo.

A mis amigos por brindarme su apoyo moral, su apoyo incondicional y valiosa amistad.

A todos los que faltaron por mencionar que contribuyeron con un granito de arena para poder concluir con éxito la meta propuesta, siempre estaré en deuda con ustedes.

Índice general

Capítulo 1 Preliminares

| | | |
|-------|----------------------------------|---|
| 1.1 | Introducción | 1 |
| 1.2 | Planteamiento del problema | 6 |
| 1.2.1 | Objetivo | 7 |
| 1.2.2 | Justificación | 7 |
| 1.3 | Metodología | 7 |
| 1.4 | Alcances y limitaciones | 8 |

Capítulo 2 Antecedentes

| | | |
|---------------------------|--|----|
| 2.1 | Estructura y componentes de un robot | 9 |
| 2.1.1 | Eslabón | 9 |
| 2.1.2 | Actuador | 10 |
| 2.1.3 | Transmisiones | 10 |
| 2.1.4 | Articulaciones | 10 |
| 2.1.5 | Tipos de actuadores | 11 |
| 2.2 | Características de los robots | |
| 2.2.1 | Grados de libertad (GDL) | 12 |
| 2.2.2 | Espacio de trabajo | 12 |
| 2.2.3 | Precisión de los movimientos | 13 |
| Resolución espacial | 13 | |
| Exactitud | 14 | |
| Repetibilidad | 14 | |
| 2.2.4 | Capacidad de carga | 14 |
| 2.2.5 | Velocidad | 14 |
| 2.3 | Configuración de robots | 15 |
| 2.3.1 | Anatomía del robot | 15 |
| 2.3.2 | Configuración de robots | 15 |
| 2.3.2.1 | Configuración cartesiana | 16 |

| | | |
|-----------------------------------|---|----|
| 2.3.2.2 | Configuración cilíndrica | 16 |
| 2.3.2.3 | Configuración polar o esférica | 16 |
| 2.3.2.4 | Configuración angular | 16 |
| 2.3.2.5 | Configuración SCARA | 16 |
| Capítulo 3 Estructura mecánica | | |
| 3.1 | Descripción general | 18 |
| 3.2 | Estructura mecánica | 19 |
| 3.2.1 | Efector final | 19 |
| 3.2.2 | Muñeca | 21 |
| 3.2.3 | Codo | 24 |
| 3.2.4 | Hombro | 28 |
| 3.2.5 | Cintura | 32 |
| 3.3 | Base para el brazo robótico | 34 |
| Capítulo 4 Electrónica | | |
| 4.1 | Alimentación principal | 36 |
| 4.2 | Etapas de potencia | 37 |
| 4.3 | Tarjeta de adquisición de datos | 39 |
| 4.4 | Pruebas del sistema | 42 |
| 4.5 | Posición inicial con software Arduino | 42 |
| 4.6 | Descripción del software de programación LabView | 46 |
| 4.7 | Posición inicial por interfaz gráfica en LabView y Arduino | 48 |
| Capítulo 5 Programación y control | | |
| 5.1 | Objetivos de control para las articulaciones del brazo robótico | 50 |
| 5.1.1 | Control en lazo abierto de efector final - Pinza | 50 |
| 5.1.2 | Dirección de giro para motor a pasos | 51 |
| 5.1.3 | Funcionamiento del motor de corriente directa con reducción | 52 |
| 5.2 | Descripción del software del sistema | 55 |
| 5.3 | Funciones utilizadas | 55 |
| 5.4 | Interfaz de usuario en Labview para brazo robótico | 55 |

| | |
|---|----|
| Capítulo 6 Pruebas y resultados | |
| 6.1 Pruebas de brazo robótico con la interfaz gráfica | 58 |
| 6.2 Ajuste de la velocidad del movimiento de las articulaciones | 59 |
| 6.3 Resultados | 60 |
| Conclusiones | 77 |
| Trabajos futuros | 79 |
| Bibliografía | 81 |
| Anexo 1 Esquemático de conexiones eléctricas y electrónicas de brazo robótico | |
| 1.1 Diagrama esquemático de eléctricas y electrónicas. | |
| 1.2 Lista y significado de las etiquetas colocadas en el esquemático. | |
| Anexo 2 Costos | |
| 2.1 Análisis de costos. | |
| Anexo 3 Tarjetas de potencia | |
| 3.1 Circuitos impresos de etapas de potencia. | |
| Anexo 4 Programación implementada en el brazo robótico. | |
| 4.1 Programa de posición inicial para el brazo robótico | |
| Anexo 5 Guía de usuario. | |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1. Robots en aplicaciones de laboratorio | 2 |
| Figura 2. Aplicaciones de robots industriales | 5 |
| Figura 3. Eslabones en un brazo robótico | 9 |
| Figura 4. Actuador eléctrico | 10 |
| Figura 5. Eje de transmisión con banda dentada | 10 |
| Figura 6. Articulaciones de brazo robótico | 10 |
| Figura 7. Brazo robot con 6 grados de libertad | 12 |
| Figura 8. Espacio de trabajo de robots manipuladores | 12 |
| Figura 9. Robot manipulador antropomórfico | 15 |
| Figura 10. Configuraciones para brazos robóticos | 17 |
| Figura 11. Sistema mecánico (izq), vista superior (centro), vista lateral (der) | 20 |
| Figura 12. Conexión de servomotor en la regleta de conexiones | 22 |
| Figura 13. Motor a pasos | 23 |
| Figura 14. Montaje de motor a pasos | 23 |
| Figura 15. Interconexión de driver de potencia con motor a pasos | 24 |
| Figura 16. Conexionado de sensor óptico en regleta de conexiones | 25 |
| Figura 17. Montaje de estructura mecánica de articulación del codo | 27 |
| Figura 18. Motorreductor ASMO | 27 |
| Figura 19. Endoder ENA1D-481 | 27 |
| Figura 20. Conexión del sensor QTR-1A de la articulación del codo | 28 |
| Figura 21. Conexión de la etapa de potencia con el motor ASMO | 29 |
| Figura 22. Conexión del encoder ENA1D-481 de articulación del codo | 30 |
| Figura 23. Conexión del subsistema que compone la articulación del codo (izq), espacio de trabajo de la articulación del codo (der) | 30 |
| Figura 24. Motor Toshiba modelo DGM-204-2A | 31 |
| Figura 25. Encoder ENA1D-481 | 31 |
| Figura 26. Montaje mecánico de la articulación del hombro | 32 |

| | |
|--|----|
| Figura 27. Conexión de sensor QTR-1A de articulación del hombro | 33 |
| Figura 28. Conexión de etapa de potencia con motor Toshiba | 34 |
| Figura 29. Conexión de encoder ENA1D-481 para articulación del hombro | 34 |
| Figura 30. Conexión del subsistema que compone la articulación del hombro (izq), espacio de trabajo de la articulación del hombro (der) | 35 |
| Figura 31. Montaje mecánico de la cintura | 36 |
| Figura 32. Conexión de sensor QTR-1A para articulación de cintura | 36 |
| Figura 33. Etapa de potencia con motor Toshiba para articulación de cintura | 37 |
| Figura 34. Conexión del subsistema que compone la articulación de la cintura ... | 37 |
| Figura 35. Electrónica para brazo robótico | 38 |
| Figura 36. Instrumentación de brazo robótico | 38 |
| Figura 37. Fuente de alimentación para brazo robótico | 40 |
| Figura 38. Esquemático de driver de potencia ULN2803 | 42 |
| Figura 39. Esquemático de conexión de driver de potencia L298N | 42 |
| Figura 40. Conexiones de señales digitales para puente H | 43 |
| Figura 41. Placas de desarrollo Arduino, tarjetas de adquisición de datos USB 6009 | 46 |
| Figura 42. Esquemático de conexiones para control de giro para articulación de la muñeca | 48 |
| Figura 43. Esquemático de conexiones para control de giro para articulación del codo | 49 |
| Figura 44. Esquemático de conexiones para control de giro de articulación del hombro | 50 |
| Figura 45. Esquemático de conexiones para control de giro de articulación de la cintura | 51 |
| Figura 46. Ventana de inicio del programa LabView versión 2014 | 52 |
| Figura 47. Diagrama de bloques para control de apertura y cierre de efector final | 55 |
| Figura 48. Grafica de apertura y cierre del efector final | 56 |
| Figura 49. Diagrama para control de giro para articulación de la muñeca | 56 |
| Figura 50. Diagrama para control de articulación del codo | 57 |

| | |
|--|-----|
| Figura 51. Diagrama para control de articulación del hombro | 58 |
| Figura 52. Diagrama para control de articulación de la cintura | 58 |
| Figura 53. Diagrama en lazo cerrado para control de posición para motores de corriente directa de brazo robótico | 59 |
| Figura 54. Interfaz gráfica de usuario para control de giro de cada una de las articulaciones de brazo robótico | 62 |
| Figura 55. Activación de botón virtual Posición Inicial | 65 |
| Figura 56. Visualización de Posicionamiento inicial usando interfaz gráfica inicial | 65 |
| Figura 57. Visualización de Posicionamiento inicial usando interfaz gráfica final | 66 |
| Figura 58. Encendido de indicadores luminosos al mover cada articulación mediante la interfaz gráfica | 67 |
| Figura 59. Monitoreo de señales de sensores en interfaz gráfica de LabView ... | 69 |
| Figura 60. Monitoreo de señales de sensores con LabView y Arduino | 70 |
| Figura 61. Prueba de sujeción de objetos por medio de la interfaz gráfica | 70 |
| Figura 62. Circuitos impresos de etapas de potencia para motores de corriente directa | 83 |
| Figura 63. Configuración de tarjeta Arduino mega 2560 | 108 |
| Figura 64. Puertos de comunicación para tarjetas Arduino | 109 |
| Figura 65. Configuración de puerto serial para tarjeta Arduino mega | 109 |
| Figura 66. Configuración de tarjeta Arduino UNO | 110 |
| Figura 67. Configuración de puerto serial para tarjeta Arduino UNO | 110 |
| Figura 68. Reconocimiento de tarjetas DAQ 6009 en el programa NI MAX | 111 |
| Figura 69. Ejecución de la interfaz gráfica para mover cada una de las articulaciones de brazo robótico | 112 |
| Figura 70. Posición inicial para articulación de codo y cintura de brazo robótico | 114 |
| Figura 71. Posición inicial para hombro y cintura de brazo robótico | 114 |
| Figura 72. Botón de encendido de fuente de alimentación de brazo robótico | 115 |

| | |
|---|-----|
| Figura 73. Monitores seriales que muestran la ejecución para posicionar cada articulación a su posición inicial | 116 |
| Figura 74. Posición inicial de brazo robótico al oprimir botón Posición Inicial en interfaz de LabView | 117 |
| Figura 75. Espacio para programación extra para aplicaciones prácticas | 118 |

INDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Características del efector final | 22 |
| Tabla 2. Características mecánicas y eléctricas del motor a pasos | 23 |
| Tabla 3. Secuencias de giro para motor a pasos | 25 |
| Tabla 4. Características mecánicas y eléctricas del sensor de cuadratura y el motorreductor utilizado en la articulación de Codo | 28 |
| Tabla 5. Características del sensor de cuadratura y el motorreductor utilizado en la articulación del hombro | 32 |
| Tabla 6. Secuencias que se deben aplicar a motores de corriente directa para obtener una dirección de giro | 43 |
| Tabla 7. Características técnicas de la placa Arduino UNO | 44 |
| Tabla 8. Especificaciones técnicas de la tarjeta Arduino mega | 45 |
| Tabla 9. Especificaciones técnicas de la tarjeta de adquisición de datos USB 6009 | 46 |
| Tabla 10. Costo de materiales mecánicos | 79 |
| Tabla 11. Costo de materiales eléctricos | 80 |
| Tabla 12. Costo de materiales electrónicos | 80 |
| Tabla 13. Costo total para la construcción de brazo robótico | 81 |

Resumen

El presente proyecto consiste en el diseño, construcción y puesta en marcha de un brazo robótico de uso didáctico con cuatro grados de libertad, de arquitectura abierta, y de bajo costo.

Se detalla paso a paso la construcción, montaje mecánico e instrumentación de cada una de las articulaciones del brazo.

El robot manipulador cuenta con una interfaz gráfica programada en Labview, donde el usuario puede interactuar con el brazo robótico manipulando cada una de sus articulaciones de forma independiente, y además pudiendo realizar el monitoreo de cada una de las señales provenientes de los sensores colocados en el robot.

Finalmente el brazo robótico es acompañado de una guía de usuario para la correcta operación del brazo robótico.

Capítulo 1 Preliminares

1.1 Introducción y aplicaciones

La definición para un robot manipulador de la Asociación de Industrias de Robótica (RIA, por sus siglas en inglés de *Robotic Industry Association*), establece que un robot industrial es un “manipulador multifuncional reprogramable, capaz de mover materiales, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar diversas tareas”. [1]. Esta definición deja en claro que los robots manipuladores pueden ser utilizados en aplicaciones funciones y características diferentes. Por tanto, son empleados en los siguientes campos:

Industria

En la actualidad los robots manipuladores tienen su principal fuente de aplicación en la industria como parte de la automatización de procesos. Generalmente operan en conjunto con otras máquinas-herramientas formando células de trabajo, siendo un elemento indispensable en la mayoría de los procesos de manufactura.

“En cuanto al tipo de robot a utilizar, se deben considerar por ejemplo velocidad de carga, capacidad de control, otras aplicaciones (soldadura pintura) etc”. [2]

Laboratorios.

“Últimamente los robots están encontrando un gran número de aplicaciones en los laboratorios. Éstos llevan a cabo de manera efectiva tareas repetitivas como la colocación de tubos de pruebas dentro de los instrumentos de medición. También se utilizan para realizar procedimientos manuales automatizados.” [2]



Figura 1. Robots en aplicaciones de laboratorios. [3]

“Los fabricantes de estos sistemas mencionan tres ventajas sobre la operación manual:

- Incrementan la productividad.
- Mejoran el control de calidad.
- Reducen la exposición del ser humano a sustancias químicas nocivas y ambientes hostiles”. [2]

“Otras tareas que se pueden realizar son la medición del pH, viscosidad, el porcentaje de sólidos en polímeros, preparación de plasma humano para muestras para ser examinadas, calor, flujo, peso y disolución de muestras para presentaciones espectromáticas.” [2]

Control de calidad

“La tendencia a conseguir una completa automatización de la producción abarca inclusive el control de la calidad. El robot industrial puede participar en esta tarea usando su capacidad de posicionamiento y manipulación”. [4] “Un robot puede usarse como un manipulador encargado de clasificar piezas según ciertos criterios de calidad (piezas correctas e incorrectas). En este caso, el control y decisión sobre qué familia pertenece la pieza se hace mediante un sistema específico, capaz de comunicarse con el robot (visión artificial). No existe, en este caso, un tipo concreto de robot más adecuado para estas tareas.

En el control dimensional suelen usarse robots cartesianos por la precisión de estos pero, en general, son igualmente válidos para robots articulados”. [4]

Educación

“El auge de la Robótica y la imperiosa necesidad de su implantación en numerosas instalaciones industriales, requiere de especialistas en la materia. La Robótica es una tecnología multidisciplinar. Hace uso de todos los recursos de vanguardia de otras ciencias afines”. [5] En la que destacan las siguientes:

- Mecánica
- Cinemática
- Dinámica
- Matemáticas
- Automática
- Electrónica
- Informática
- Energía y actuadores eléctricos, neumáticos e hidráulicos
- Visión artificial
- Sonido de máquinas
- Inteligencia artificial.

“Realmente la Robótica es una combinación de todas las disciplinas expuestas, más el conocimiento de la aplicación a la que se enfoca”. [5]

Existen robots los cuales tienen en cada una de sus articulaciones servomecanismos (motores que internamente tiene lazos de control retroalimentado), para aplicaciones donde el eje del mecanismo sea utilizado en control de posición, y velocidad. Son de bajo costo, en su mayoría carecen de la programación necesaria para realizar alguna aplicación, por ejemplo los robots didácticos lego, o prótesis de mano robótica.

Nuevos sectores de aplicación de la robótica

“Las aplicaciones de la robótica examinadas anteriormente responden a los sectores que van desde la manufactura, hasta la industria automotriz y que, han sido usuarios habituales desde hace 30 años de los robots industriales”. [4] Hoy en día, los robots manipuladores han extendido su campo de acción trabajando como *robots de servicio* y están siendo aplicados en sectores como”: [4]

- Agricultura y silvicultura
- Ayuda a discapacitados
- Construcción
- Domésticos
- Entornos peligrosos
- Espacio
- Medicina y salud
- Minería
- Entornos submarinos
- Vigilancia y seguridad
- Tele presencia

En la figura 2 se muestra algunas aplicaciones de los robots manipuladores.



Figura 2. Aplicaciones de robots industriales. [6] [7]

1.2 Planteamiento del Problema

La enseñanza de la robótica en las escuelas de nivel medio y superior debe contar con equipos que sean didácticos, con características de tipo industrial, sin embargo estos tienen costos muy elevados y su mantenimiento no lo puede hacer uno como usuario, sino que lo tiene que realizar la empresa donde se compró el robot. Sin embargo la ausencia de ellos limita el aprendizaje y la práctica de conocimientos impartidos en clase.

Este proyecto surge por la carencia de equipos robóticos didácticos en la UACM, en particular para la opción terminal de la carrera de Ingeniería en sistemas electrónicos industriales.

1.2.1 Justificación

El plantel Casa libertad no cuenta con el suficiente equipamiento en cuanto a robots manipuladores, al no tener los dispositivos o elementos que le permitan al estudiante interactuar y comprender de manera más sencilla el funcionamiento interno de un brazo robótico, esto acarrea como consecuencia que el estudiante no tenga una mejor comprensión de los temas que se abordan en las clases teóricas impartidas en el aula, la ausencia de ellos ocasiona que se limite el aprendizaje y la aplicación práctica de los conocimientos impartidos en clase.

Por tal motivo se ha optado por construir un brazo robótico de 4 grados de libertad cuya finalidad y propósito es para uso didáctico para la enseñanza en la carrera de Ingeniería en Sistemas Electrónicos Industriales, y que sea utilizado en las materias de las diferentes opciones terminales de la carrera como Robótica y Automatización; y Control e Instrumentación.

1.2.2 Objetivo

Diseñar un brazo robótico manipulador de 4 grados de libertad, de arquitectura abierta, con fines didácticos.

1.2.3 Objetivos particulares

- 1.- Diseño de un brazo robótico, que sea utilizado para realizar prácticas de laboratorio, que implique aplicar los conocimientos adquiridos en clase.
- 2.- Construir un brazo robótico de bajo costo, cuya estructura mecánica sea de madera y de bajo costo.
- 3.- Instrumentar con diferentes tipos de sensores para medición de desplazamiento articular.

1.3 Metodología

- Construcción física del robot.
- Implementar distintos tipos de motores y de sensores de uso común en la construcción de robots.
- Utilizar diferentes tipos de tarjetas de adquisición de datos.
- Desarrollo del ambiente grafico utilizando el software especializado para generar el movimiento del brazo robótico.
- Programación y funcionamiento del brazo robótico, con una interfaz gráfica amigable.
- Generación de una guía de usuario.

1.4 Alcances y limitaciones

Los estudiantes se enfrentaran al reto de aplicar sus conocimientos adquiridos en clase para controlar los diferentes actuadores que componen a cada una de las articulaciones del brazo robótico.

Como trabajos futuros se puede usar este robot como parte de un sistema que integre una serie de brazos robóticos para formar una línea de producción que posteriormente se pueda automatizar. Realización de practicarío que esté basado en temas que sean utilizados en el área de robótica, control e instrumentación, por ejemplo:

- a) Probar la cinemática directa e inversa.
- b) Programar al robot manipulador para que realice seguimiento de una trayectoria para llegar a un punto específico en el espacio de trabajo del mismo.
- c) Implementar controladores para controlar cada articulación del brazo robótico de forma independiente.
- d) Usarlo para temas relacionados con metrología e instrumentación.

El prototipo no pretende ser utilizado para uso industrial o exhaustivo, su alcance es permitir al usuario experimentar los distintos tipos de motores y sensores en aplicaciones que sí sean de tipo industrial y el uso de distintos tipos de tarjetas de adquisición de datos.

Capítulo 2 Antecedentes

En este capítulo se describirán algunos antecedentes de la robótica, se verán las partes fundamentales que componen a un robot, poniendo énfasis en los robots manipuladores, se estudiará brevemente la morfología y las configuraciones de cada uno de ellos.

2.1 Estructura y Componentes de un Robot

La estructura mecánica básica de un brazo robot está compuesta por **eslabones**, que cumplen una función similar a la de un hueso. También por **actuadores**, de funcionalidad parecida a la de un músculo. Además poseen **sistemas de transmisión**, con cierto parecido a los tendones. Y finalmente, los cables para el envío de señales, que en cierto modo se les considera los nervios del robot. Los puntos de unión entre los eslabones reciben el nombre de **nodos** y el movimiento que permite dicha unión y un movimiento relativo entre ellos, al igual que en el caso de un brazo humano, es la **articulación**.

A continuación se van a definir los elementos que componen la estructura mecánica del brazo robótico.

2.1.1 Eslabón

“Es la pieza rígida que sirve para la transferencia de movimiento a cada una de las articulaciones que componen a un robot manipulador, para que conduzcan a un elemento final, colocado en el extremo de la estructura; a las posiciones y orientaciones necesarias para la realización de una tarea dada”. [8] “Los eslabones se dividen en dos los eslabones móviles y los inmóviles (eslabón donde está montada la estructura mecánica del robot)”. [8]

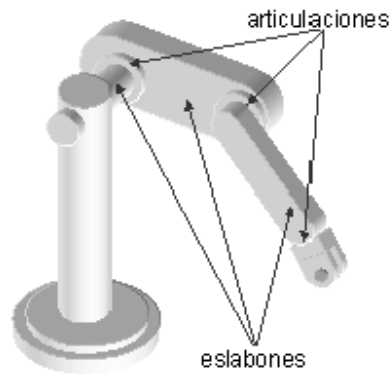


Figura 3. Eslabones en un brazo robótico. [9]

2.1.2 Actuador

Son los dispositivos que producen el par o fuerza mecánica que se requiere para mover los eslabones del manipulador. Pueden ser motores eléctricos, pistones y motores neumáticos o hidráulicos.



Figura 4. Actuador eléctrico. [10]

2.1.3 Transmisiones

Permiten la transferencia del torque generado por los actuadores hasta cada eslabón del manipulador. Pueden ser del tipo de bandas dentadas, engranes, cadenas etc.

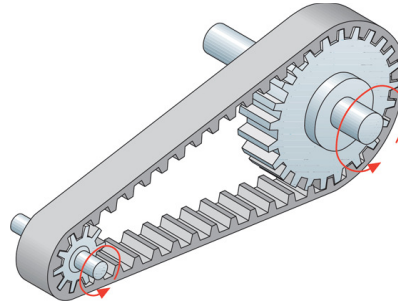


Figura 5. Eje de transmisión con banda dentada. [11]

2.1.4 Articulación

Las articulaciones de un robot pueden ser de **revolución o prismáticas**, la primera se refiere a un movimiento rotacional generado por el movimiento de la flecha de un motor o una caja reductora de engranes, y la articulación prismática permite realizar solo traslaciones lineales, de avance o retroceso.

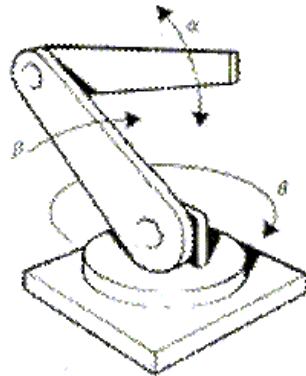


Figura 6. Articulaciones de brazo robótico. [12]

2.1.5 Tipos de actuadores

Los elementos motrices que generan el movimiento de las articulaciones pueden ser, según la energía que consuman, de tipo hidráulico, neumático o eléctrico. [13] Los actuadores de tipo hidráulico se aplican a tareas que requieren una gran potencia y grandes capacidades de carga. Dado el tipo de energía que emplean, se construyen con mecánica de precisión y su coste es elevado.

Los robots hidráulicos se diseñan formando un conjunto compacto, que consiste en una central hidráulica, la cabina electrónica de control y el brazo del manipulador. [13]

Finalmente los motores eléctricos, que cubren la gama de media y baja potencia, acaparan el campo de la Robótica, por su gran precisión en el control de su movimiento y las ventajas inherentes a la energía eléctrica que consumen. [13]

2.2 Características de los Robots

La robótica tiene diversas características para que se puedan desarrollar y aplicar las tareas deseadas en un proceso de producción; las características de los robots son las siguientes:

- Grados de libertad (GDL)
- Espacio de trabajo
- Precisión de los movimientos
- Capacidad de carga
- Velocidad
- Tipo de actuadores

2.2.1 Grados de libertad (GDL)

Un grado de libertad es la capacidad de moverse a lo largo de un eje (movimiento lineal) o rotar a lo largo de un eje (movimiento rotacional). [14]

El número de grados de libertad del robot viene dado por la suma de los GDL de las articulaciones que lo componen. [12] Un mayor número de grados de libertad conlleva un aumento en la flexibilidad en el posicionamiento del elemento terminal. [12]

Los grados de libertad de un brazo manipulador de tipo industrial están directamente relacionados con su anatomía o configuración. [12]



Figura 7. Brazo robot con 6 grados de libertad. [15]

2.2.2 Espacio de trabajo

Es la zona que está definida por las dimensiones de los elementos del manipulador, junto a los grados de libertad definen la zona de trabajo del robot, característica fundamental en las fases de selección e implementación del modelo adecuado. [12]

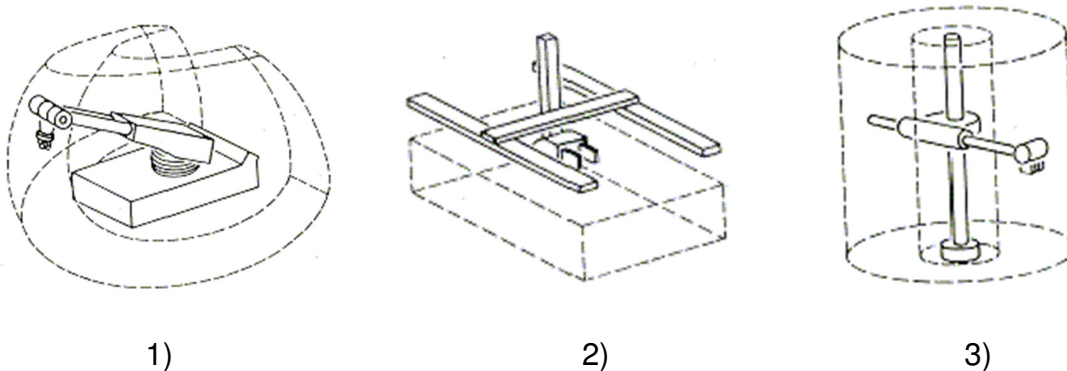


Figura 8. Espacio de trabajo de robots manipuladores: 1) Angular, 2) Cartesiano, 3) Cilíndrico. [12]

2.2.3 Precisión de los movimientos

La precisión del movimiento de un robot tipo industrial depende de tres factores:

- a) Resolución espacial
- b) Exactitud
- c) Repetibilidad

a) Resolución espacial

La resolución espacial se define como el incremento más pequeño de movimiento en que el robot puede dividir su espacio de trabajo. [12] La resolución espacial depende de dos factores. [12] Los sistemas que controlan la resolución y las inexactitudes mecánicas. [12]

Los sistemas de control son precisamente, el medio para realizar todos los incrementos individuales de una articulación. [12] Los controladores dividen el intervalo total de movimiento para una articulación particular en incrementos individuales (resolución de control o de mando). [12] Para una articulación con un encoder digital, el número de incrementos separados e identificables es 2^n . [12] Por ejemplo, en un robot con $n=8$ la resolución de mando puede dividir el intervalo del movimiento en 256 posiciones discretas. [12] Así, la resolución de mando es el intervalo de movimiento/256. Los incrementos casi siempre son uniformes. [12]

b) Exactitud

Se refiere a la capacidad de un robot para situar el extremo de su muñeca en un punto señalado dentro del espacio de trabajo. [12] Mide la distancia entre la posición especificada y la posición real del actuador terminal del robot. [12] Mantiene una relación directa con la resolución espacial, es decir, con la capacidad del control del robot de dividir en incrementos muy pequeños el espacio de trabajo. [12]

c) Repetibilidad

Se refiere a la capacidad del robot de regresar al punto programado las veces que sean necesarias. [12] Esta magnitud establece el grado de exactitud en la repetición de los movimientos de un manipulador al realizar una tarea programada. [12]

2.2.4 Capacidad de carga

El peso, que puede transportar la pinza del manipulador recibe el nombre de capacidad de carga. [12] Este dato lo proporcionan los fabricantes, incluyendo el peso del efector final. [12] En modelos de robots industriales, la capacidad de carga de la pinza, puede oscilar entre 0.9 kg y 205 kg. [12]

La capacidad de carga es una de las características que más se tienen en cuenta en la selección de un robot, según la tarea a la que se destine. [12] En soldadura y mecanizado es común precisar capacidades de carga superiores a los 50kg. [12]

2.2.5 Velocidad

“Se refiere a la velocidad máxima alcanzable por las articulaciones. En muchas ocasiones, una velocidad de trabajo elevada, aumenta extraordinariamente el rendimiento del robot, por lo que esta magnitud se valora considerablemente en la elección del mismo”. [12] “En tareas de soldadura y manipulación de piezas es muy aconsejable que la velocidad de trabajo sea alta. En pintura, mecanizado y ensamblaje, la velocidad debe ser menor“. [12]

2.3 Configuración de los robots

De manera general, y basándose en su morfología, los robots se suelen dividir en diferentes tipos:

Robot industrial o manipulador, robots de servicio, robots de investigación, robots móviles, robots militares, robots andróides o humanoides, robots médicos, nano robots, robots en la educación.

2.3.1 Anatomía del robot (Robot antropomorfo)

La anatomía del robot se refiere a la construcción física del cuerpo. La mayoría de los robots utilizados en la industria se encuentran sobre una base fija al suelo. En el caso de los brazos robóticos, el cuerpo está compuesto de diversos componentes que le permiten posicionar y orientar su elemento final. Los movimientos del brazo obedecen a las articulaciones que generan giros o deslizamientos.

La muñeca del robot se compara como un elemento importante contenido en un brazo humano la cual tiene unida una mano; que en términos técnicos para un brazo robótico de tipo industrial se conoce como efector final, que dependiendo del tipo, realiza la tarea designada.

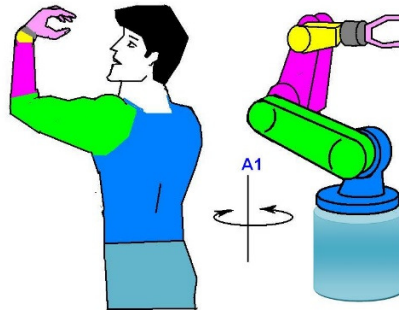


Figura 9. Robot manipulador antropomórfico. [16]

2.3.2 Configuraciones de robots

Las articulaciones del brazo son las más potentes del robot, ya que soportan el peso del resto del mismo. Además, si se requiere un volumen de trabajo grande, la potencia será mayor. El brazo se puede construir según diferentes configuraciones, siendo de destacar como las más utilizadas las siguientes:

2.3.2.1 Configuración cartesiana

“Los robots que utilizan esta configuración poseen tres movimientos, es decir, tiene tres grados de libertad, los cuales corresponden a los movimientos localizados a lo largo de los ejes cartesianos X, Y, Z.

Los movimientos que realiza este tipo de robot que va dentro de un punto al otro son con base de interpolaciones lineales. A la trayectoria realizada en línea recta se le conoce como interpolación lineal”. [17]

Usado para trabajos de tomar y colocar, operaciones de ensamblado, manipulación de máquinas de herramientas y soldadura por arco.

2.3.2.2 Configuración cilíndrica

“Esta configuración tiene dos articulaciones prismáticas y una de rotación. La primera articulación es normalmente de rotación, la posición se fija en coordenadas cilíndricas”. [18] Se usa para soldadura por punto, fundición a presión, para operaciones de ensamblaje.

2.3.2.3 Configuración polar o esférica.

Esta configuración tiene 3 articulaciones, las primeras dos articulaciones son de tipo rotacional, en tanto la tercera es de tipo primatica. “Este tipo de robot utiliza la interpolación por articulación para moverse en sus dos primeras articulaciones y la interpolación lineal para la extensión y retracción”. [19] Es utilizado para soldadura por gas y por arco.

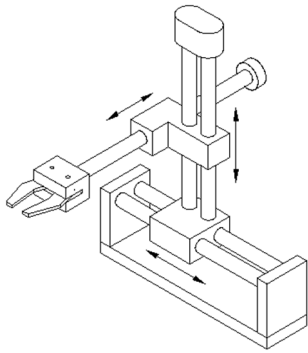
2.3.2.4 Configuración angular

“Esta configuración es una estructura con tres articulaciones de rotación”. [18] La posición del extremo final se especifica de forma natural en coordenadas angulares. Es muy empleada en robots manipuladores industriales, especialmente en tareas de manipulación que tengan cierta complejidad. “La configuración angular es la más utilizada en educación y en actividades de investigación y desarrollo”. [18]

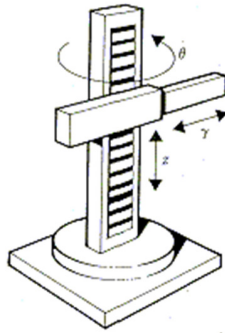
2.3.2.5 Configuración SCARA

“Está constituida por dos articulaciones de rotación con respecto a dos ejes paralelos, y una de desplazamiento en sentido perpendicular al plano”. [20]

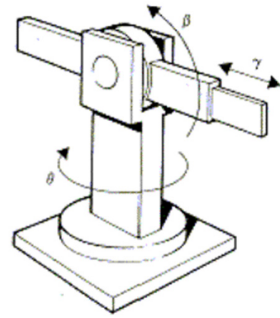
Las aplicaciones más comunes son en las áreas de ensamblaje mecánico (discos duros, componentes electrónicos, etc.), alimentación a otras máquinas y en muchas operaciones donde se requieren movimientos repetitivos, rápidos y precisos.



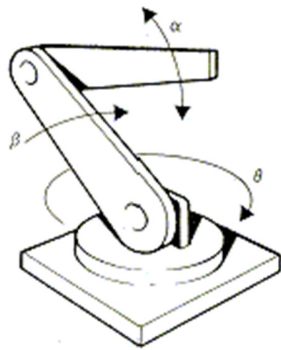
Configuración Cartesiana



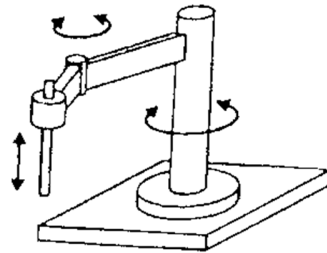
Configuración Cilíndrica



Configuración Esférica



Configuración Angular



Configuración Sacara

Figura 10. Configuraciones para brazos robóticos. [12] [21]

Capítulo 3 Estructura mecánica

En este capítulo se describe el proceso de construcción del brazo robótico, así como los elementos que se tomaron en cuenta para el ensamblaje de la estructura mecánica, y los dispositivos electrónicos que se implementaron en cada una de las articulaciones del robot, como: son actuadores de corriente directa, sensores ópticos y sensores de cuadratura.

3.1 Descripción general

El brazo robótico es de 4 grados de libertad, tres de sus articulaciones tienen la función de brindar la posición, y la última articulación da la orientación que permite al robot realizar los diferentes movimientos de traslado y colocación de un objeto.

El brazo robótico realiza la manipulación y traslado de objetos de un punto a otro dentro del espacio de trabajo, en este caso será de 0 a 180 grados (cintura). El brazo robótico no dará un giro completo debido a que en la zona que va de 180 grados a los 360 grados está asignada para la colocación del cable de alimentación eléctrica, y los cables USB tipo B, para la comunicación con la PC.

Para el diseño de la estructura mecánica se comenzó por el efector final encargado de sujetar los objetos a manipular de un peso y tamaño específico, como se explicará más adelante. La siguiente etapa de construcción del brazo robótico consiste en la colocación de un motor a pasos, el cual va a cumplir la función de la muñeca, el eje del motor estará unido con el efector final con elementos mecánicos.

La tercera etapa a implementar es el codo, se construyó una estructura mecánica de madera la cual va a acoplar el codo, la muñeca y el efector final. A esta articulación se le colocó un motorreductor de corriente directa de la marca ASMO y diferentes tipos de sensores como parte de la instrumentación.

La cuarta etapa fue construir una base de madera que forma parte de la articulación del hombro, para montar ahí la estructura mecánica que compone a las articulaciones del codo, muñeca y efector final.

Al hombro se le colocó también un motorreductor de corriente directa de la marca Toshiba modelo DGM-204-2A, y diferentes tipos de sensores de posición.

Por último se construyó la cintura del brazo robótico que es el soporte de toda la estructura mecánica del brazo, se tuvo que colocar elementos mecánicos que se describirán posteriormente con el fin de permitir dar soporte y movilidad para generar un giro suave, también se implementaron sensores de posición.

3.2 Estructura mecánica

El brazo robótico se diseñó de tal manera que pueda sujetar objetos con un diámetro no mayor a 5 cm y que no pesen más de 100 gramos, eso se estableció en función de que las limitaciones físicas de la pinza delimitando también selección de cada uno de los motores a colocar en cada una de las articulaciones siguientes: la muñeca, el codo, el hombro y por último la cintura, donde cada uno de estos motores debe tener ciertas características que se deben tomar en cuenta como: el peso de cada motor, el par o fuerza mecánica que se requiera para mover los eslabones del robot que van a sujetar la masa del eslabón, más la masa de la pinza y el objeto.

Los motores para el codo, hombro y cintura se deben tomar en cuenta las características mencionadas anteriormente para que pueda moverse el robot manipulador.

El brazo robótico se construyó pensando en cuatro grados de libertad, construidos en base a madera debido a la limitación de recursos. Para la parte del desarrollo mecánico de la construcción de cada una de las articulaciones del brazo robótico, se hizo con piezas que son de fácil acceso como: espárragos, tornillos, tuercas, rondanas, coples de aluminio, etc. Estos materiales se pensaron que sean de fácil acceso y mantenimiento.

3.2.1 Efecto final

El efecto final es el elemento que se coloca en el extremo del último eslabón y se encarga de ejecutar la acción para la cual fue diseñado el manipulador, las acciones más comunes son el transporte de objetos, la soldadura, la pintura, etc.

Uno de los efectores que más se utilizan es la pinza, un manipulador dotado con una pinza es utilizado para levantar o trasladar diversos tipos de objetos. Mientras que otros manipuladores industriales como el robot soldador que consiste en un sistema de soldadura que contiene una fuente de energía para soldar, el alimentador de alambre, la estación de limpieza y todos los componentes auxiliares del sistema de soldadura que lo componen que permiten al efector final sea de uso específico para aplicaciones particulares.

Debido a la complejidad en cuanto a la construcción del efector final, diseño y fabricación de engranes, ejes, etc. Fue más viable adquirir de forma comercial la pinza mostrada en la figura 11, es un sistema actuado con un servomotor para su apertura y cierre.

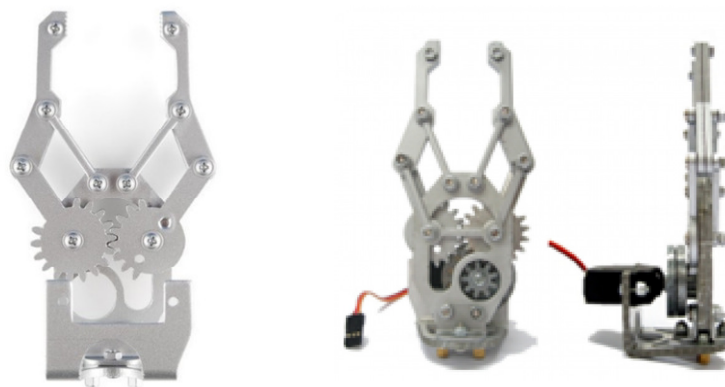


Figura 11. Sistema mecánico (izq), vista superior (centro), vista lateral (der). [22] [23]

La pinza está hecha de metal y cuenta con buje o manga de bronce, pieza mecánica cilíndrica que sirve como punto de unión entre la parte mecánica de la pinza y el servo en donde estará montado el eje del servomotor que le brindara el soporte necesario para tener un giro suave, el material con el que está construida la pinza hace sea más rígida y menos propensa a aflojarse, puede recoger objetos de un tamaño específico que no sobrepase los 5 cm en sus dimensiones y un peso menor a 100gr.

Para poder abrir y cerrar la pinza, el efector final cuenta con un motor de corriente directa, un control de posición interno, llamado servomotor, el cual esta interconectado a la pinza por un sistema de engranaje que tiene internamente un resorte cuya función es no forzar y hacer más suave el giro del eje del servomotor cuando el motor gira en el sentido horario provoca que los dos dedos de la pinza, se abran, si el motor gira en sentido anti horario, provoca el cierre de los dedos. En la siguiente tabla se muestran las características del efector final:

| | |
|---------------------------------|------------------|
| Pinza robótica | ROB – 11524 MKII |
| Relación de engranaje | 2:1 |
| Mecanismo de agarre de la pinza | Paralelo |
| Distancia de apertura | 5 cm |
| Peso de la pinza | 120 gr |
| Capacidad de carga | 300 gr |
| Servomotor | S05NF |
| Alimentación | 4.8 a 6 Volts |
| Torque | 3.19kg/cm |
| Velocidad | 0.20 seg/60º |
| Rotación | 180º |
| Peso de servomotor | 20 gr |

Tabla 1 Características del efector final.

El servomotor S05NF tiene 3 cables de conexión que son de color rojo, café y naranja, los cuales se identifican de la siguiente forma: el cable de color negro corresponde a la terminal de referencia la cual se etiqueta como **GND Servo**, el cable de color rojo es la terminal de alimentación que tiene la etiqueta **5 volts Servo**, el cable de color naranja corresponde a la terminal de control la cual se identificó como **Servo IN1**. En la figura 12 se muestra la conexión del motor en la regleta de conexiones.

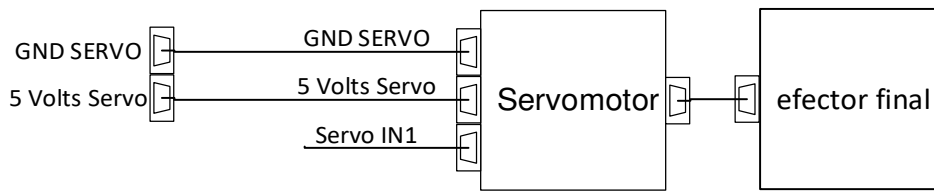


Figura 12. Conexión de servomotor en la regleta de conexiones.

El efector final se puede resumir como un subsistema cuya salida es una apertura de 0 a 5 cm y su entrada es una señal PWM que va de 0 a 100%, que se etiqueta como **Servo IN1**.

3.2.2 Muñeca

La muñeca es un elemento mecánico en donde su tamaño debe ser pequeño, para optimizar el volumen de trabajo del robot. Pero su robustez debe ser elevada es decir que debe tener buenas características de fuerza, solidez y firmeza, de cara a mantener una buena repetitividad en su movimiento de giro, y debe tener un peso reducido para no disminuir la carga efectiva del robot. Para la aplicación en este trabajo, la muñeca va a realizar un solo movimiento de giro, debido a la configuración aplicada a esta articulación. [24]

A las diferentes configuraciones de construcción de brazos robóticos, se les pueden acoplar diferentes configuraciones de muñeca que permiten obtener la orientación de la herramienta final. Las más comunes son:

- d) Muñeca en línea (in-line wrist)
- e) Muñeca PR (pitch-roll)
- f) Muñeca de giro (roll)
- g) Muñeca desplazada (offset wrist)
- h) Muñeca de tres giros o de triple giro (triple roll wrist)

La configuración que se utilizó para la articulación de la muñeca, es la llamada muñeca de **giro (roll)**. Por el movimiento de giro que va a realizar, permitirá la orientación del efector final a una cierta posición para sujetar y liberar diferentes tipos de objetos según convenga. [24]

El motor empleado para actuar la muñeca es un motor a pasos de tipo *unipolar* modelo **stepper 28BYJ48**. Se eligió por las características mostradas en la tabla 2, entre las cuales destaca su tamaño, peso y costo.

| | |
|----------------------------|----------------------|
| Motor a pasos | 28BYJ48 |
| Alimentación | 12 Volts |
| Bobinas | 4 |
| Resistencia | 50 Ω |
| Torque | 0.34 Kg * cm |
| Corriente | 55 mA |
| Numero de pasos por vuelta | 4076 |
| Relación de engranaje | 1/63.88 |
| Peso del motor a pasos | 37gr |
| Diámetro del eje | 5mm, aplanado a 3 mm |

Tabla 2. Características mecánicas y eléctricas del motor a pasos.

La figura 13 muestra el motor a pasos que se utilizó para la articulación de la muñeca. La muñeca se montó en una base de madera rectangular de 8 x 6 cm como se ve en la figura 14.



Figura 13. Motor a pasos [25]

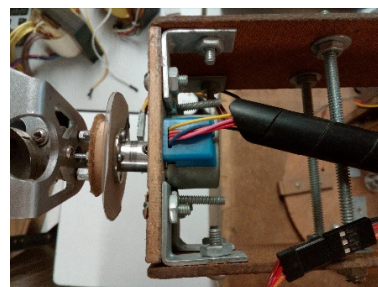


Figura 14. Montaje de motor a pasos

El motor a pasos cuenta con 5 cables de colores, los cuales se etiquetaron como sigue, el cable azul como **MPasCAzul**, el rosa **MPasCRosa**, el amarillo **MPasCAma**, el naranja **MPasCNar** y el rojo **MPasCROjo**.

Para el manejo del motor se utilizó el driver **UL2803N** como interface, para acondicionar pulsos o señales digitales de baja intensidad de tal manera que puedan mover elementos que requieran corrientes de hasta 0.5 A con voltajes de hasta 18 volts. Es compatible para el uso con tarjetas Arduino y, útil para la conmutación de cargas inductivas por que internamente está formado por siete pares de transistores NPN Darlington de uso general, incluyendo diodos de fijación de cátodo común para cada par Darlington NPN, dichos diodos ayudan a proteger a los componentes electrónicos de picos de voltajes debidos a la electricidad estática, lo que hace que este circuito integrado sea muy útil para la conmutación de bobinas.

Por esto fue que se escogió como etapa de potencia para controlar la dirección de giro del motor a pasos.

En la figura 15 se muestra como se interconecto el driver de potencia con el motor a pasos como un subsistema que compone a la articulación de la muñeca.

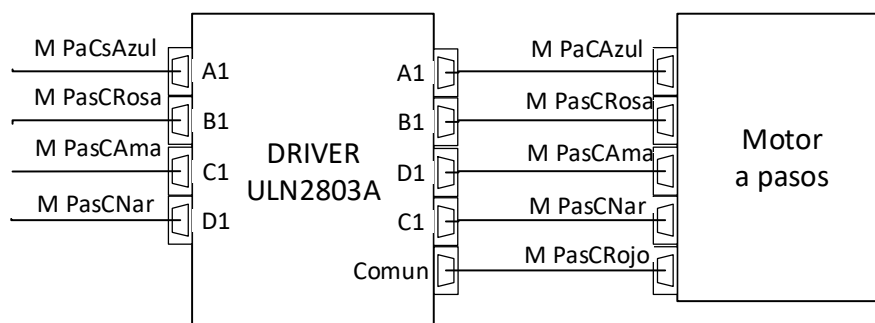


Figura 15. Interconexión de driver de potencia con motor a pasos.

En la tabla 3 se muestra la secuencia para energizar cada una de las bobinas internas del motor para girar en el sentido horario (giro a la derecha), y anti horario (giro a la izquierda).

| Giro Izquierda | | | | | Giro Derecha | | | | |
|----------------|---|---|---|---|--------------|---|---|---|---|
| Paso | A | B | C | D | Paso | A | B | C | D |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 1 | 0 | 2 | 0 | 0 | 1 | 1 |
| 3 | 0 | 0 | 1 | 1 | 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 | 4 | 1 | 1 | 0 | 0 |

Tabla 3. Secuencias de giro para motor a pasos

Estas secuencias se implementan para que el motor genere un paso cada vez (paso completo) y se obtiene un alto par de paso debido a que siempre hay dos bobinas activadas.

Entre la articulación de la muñeca y el efector final se colocó un sensor optoreflexivo, el cual actúa como un sensor de posición inicial, que permite ubicar el motor a pasos en su posición de referencia cero, para ello se montó un disco codificado colocado entre la articulación de la muñeca y la base del efector final, de esta manera el disco y el sensor óptico funcionan como un encoder simple que genera una señal digital de entrada para indicar cuando la muñeca haga el cambio de giro para alcanzar su posición cero.

El sensor optoreflexivo tiene tres cables de conexión que son de color rojo, negro y amarillo, los cuales se identifican de la siguiente forma: el cable negro es para la terminal de referencia se etiqueta como **GND OPTO1**, el cable de color rojo es la terminal de alimentación que tiene la etiqueta **5 Volts OPTO1** y el cable de color amarillo corresponde a la señal de salida del sensor y se etiqueta como **Opto1 IN1**. En la figura 16 se ve como está alambrado el sensor óptico en la regleta de conexiones.

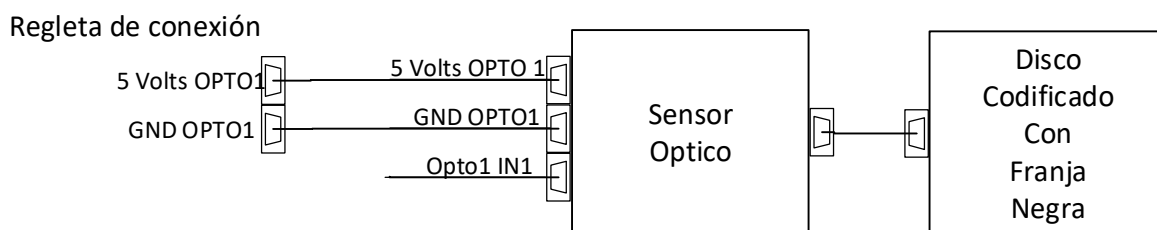


Figura 16. Conexión de sensor óptico en regleta de conexiones.

3.2.3 Codo

El codo es una articulación muy importante en el brazo robótico, da mayor libertad de movimientos gracias a la geometría de la articulación.

Para su construcción se usó madera aglomerada de 3 mm de grosor y se cortaron 2 tiras de madera de las siguientes medidas; 31 cm de largo, por 6.5 cm de ancho. Las dos piezas de madera que van a conformar la articulación del codo van a estar sujetas a la base de madera por varillas roscadas de 3/8 con sus respectivas tuercas y rondanas para sostener los elementos que componen al codo. Como eslabón entre el codo y la muñeca se colocó un elemento mecánico (cople de aluminio), como unión de estas dos articulaciones para el robot manipulador.

El eslabón del codo es actuado mediante un motorreductor de corriente directa ASMO, además, se acoplaron dos sensores. El primer sensor es un encoder de cuadratura ENA1D-481 que es un dispositivo electromecánico que convierte la posición angular de un eje, directamente a señales digitales en sus dos salidas, que internamente tiene un disco codificado con cierto número de ranuras, cuya finalidad es medir la posición y sentido de giro del motor ASMO; el segundo es optoreflexivo QTR-1A y tiene como función colocar la articulación del codo a su posición cero, el sensor óptico se encuentra ubicado de forma lateral al motor ASMO y el encoder ENA1D-481 se encuentra ubicado enfrente del motor ASMO.

El motor y el sensor de cuadratura están fijados por dos coples de aluminio. A continuación en la Figura 17, se muestra el montaje de la estructura del codo.

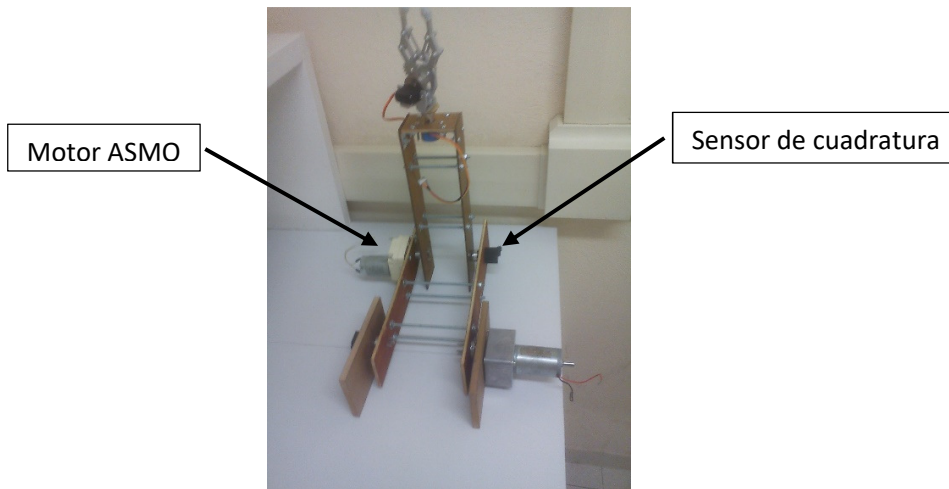


Figura 17. Montaje de estructura mecánica de articulación del codo.

El peso total de la articulación del codo es de 683 gramos, este peso incluye al efector final, y la muñeca. El motorreductor ASMO que se utilizó para la articulación del codo se muestra en la figura 18, y sus características pueden consultarse en la Tabla 5; de igual forma el sensor ENA1D-481 utilizado se muestra en la figura 19 y también se encuentra descrito en la Tabla 5.

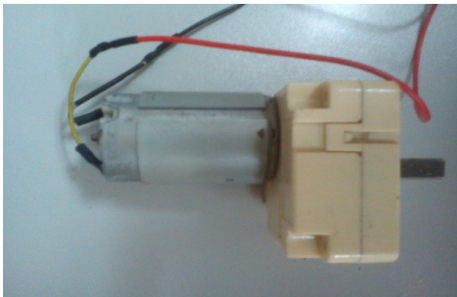


Figura 18. Motorreductor ASMO.



Figura 19. Endoder ENA1D-481. [26]

| | |
|------------------------------------|---------------------------|
| Encoder | ENA1D-481-L00100 |
| Alimentación | 5 volts |
| Resolución | 100 ciclos por revolución |
| Corriente | 26 mA |
| Diámetro del eje del sensor | 6.42 mm |
| Peso del sensor | 28.76 gr |
| Motorreductor | ASMO |
| Alimentación | 0 – 24 Volts |
| Diámetro del eje del Motorreductor | 6.42 mm |
| Peso del motorreductor | 28.76 gr |
| Torque de salida del eje del motor | 25 kg/cm |

Tabla 4. Características mecánicas y eléctricas del sensor de cuadratura y el motorreductor utilizado en la articulación de Codo.

En la articulación del codo se colocó un sensor optoreflexivo QTR-1A, cuya función es poner el motorreductor ASMO a una referencia inicial, esto con ayuda de un disco codificado ubicado enfrente de la base donde se encuentra montado el motorreductor ASMO, el sensor óptico QTR-1A, se encuentra alambrado como sigue: la terminal de referencia tiene la etiqueta **GND OPTO2**, para el cable de alimentación se le coloco la etiqueta **5 Volts OPTO2** y el cable de la señal de salida del sensor se etiqueto como **Opto2 IN2**. En la Figura 20 se ilustra la conexión del QTR-1A en la regleta de conexiones.

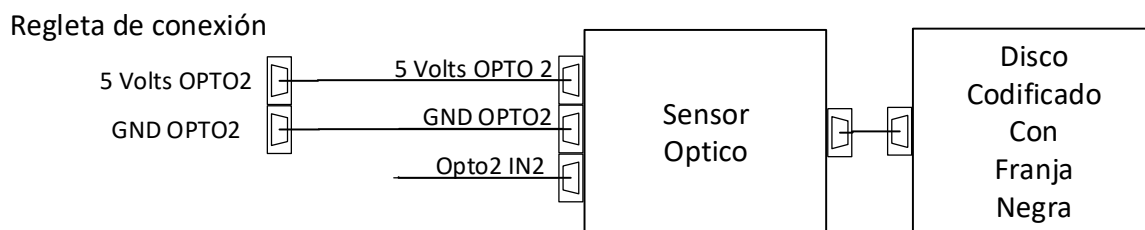


Figura 20. Conexión del sensor QTR-1A de la articulación del codo.

El motorreductor ASMO cuenta con dos cables que se etiquetaron como sigue: el cable de color negro se etiqueto como **M1-** y el cable de color rojo como **M1+**.

El motorreductor ASMO es manejado por medio del driver L298N, el cual es un circuito integrado que posee dos Puente H, formado por 4 transistores que de acuerdo a su configuración interna nos permite invertir el sentido de giro para motores de corriente directa, más importante aún mediante señales digitales PWM se puede controlar la velocidad de giro del motor.

En la figura 21 se muestra como se conecta el driver de potencia con el motorreductor ASMO.

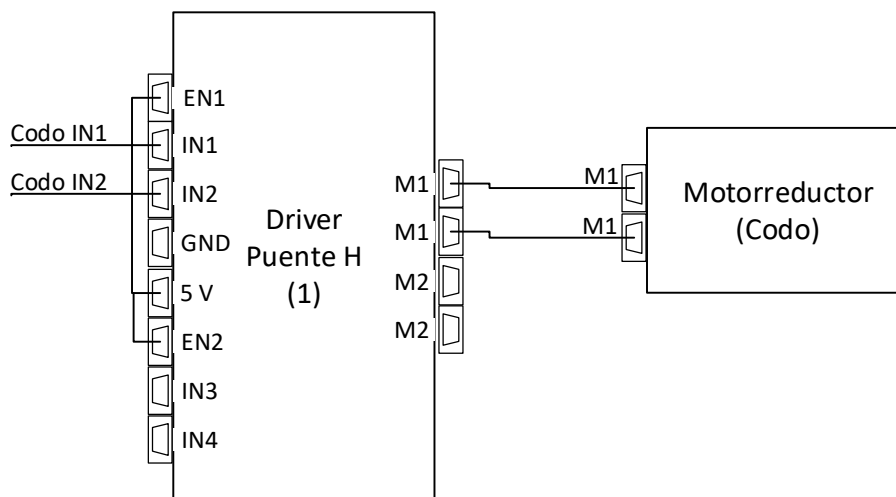


Figura 21. Conexión de la etapa de potencia con el motor ASMO.

En la articulación del codo se colocó un sensor de cuadratura ENA1D-481, cuenta con 4 pines dos de ellos son de alimentación y los otros dos pines son de señales de salida, se cablearon y etiquetaron de la siguiente manera; **GND ENC1, 5 Volts ENC1** para la alimentación del sensor, y para las señales de salida las etiquetas son estas **ENC1 ENA** y **ENC1 ENB**. En la figura 22, se muestra el sensor de cuadratura en la regleta de conexiones.

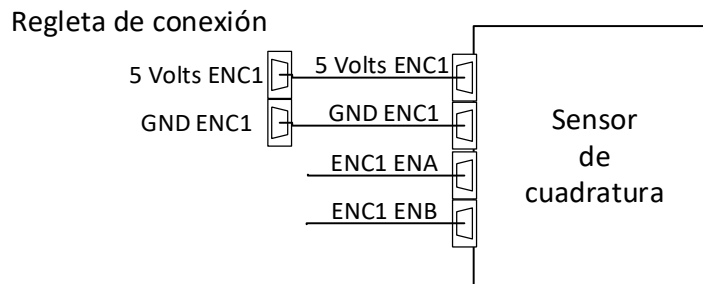


Figura 22. Conexión del encoder ENA1D-481 de articulación del codo.

El tercer grado de libertad consiste en todo un subsistema, en donde se inyectan señales digitales y lo que se obtiene físicamente es una posición angular, el codo tiene un rango de 113° , el sistema tiene dos sensores; el primer sensor es de posición límite o posición cero. El segundo sensor es un encoder de cuadratura que internamente tiene un disco codificado con cierto número de ranuras, a su salida el encoder nos entrega dos pulsos digitales, que mediante ciertas conversiones podemos conocer la posición en la que se encuentra el motorreductor. En la figura 23 se muestra las conexiones del subsistema que compone la articulación del codo, y el espacio de trabajo donde se puede mover la articulación del codo.

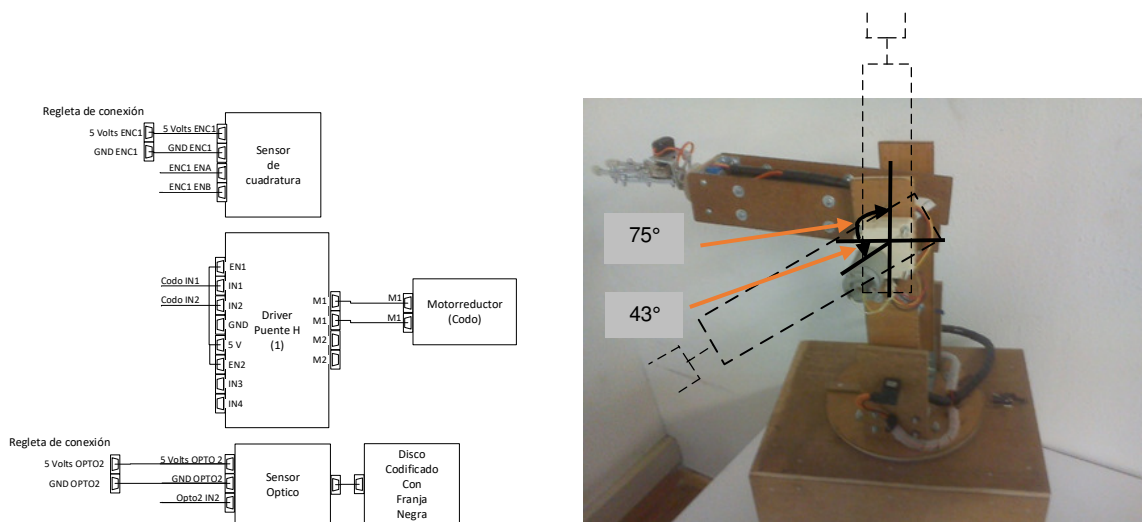


Figura 23. Conexión del subsistema que compone la articulación del codo (izq), espacio de trabajo de la articulación del codo (der).

3.2.4 Hombro

El hombro se construyó usando dos tiras de madera de triplay cuyo espesor es de 6 mm, de largo tiene 32 cm y de ancho 7 cm, que conforman la articulación, para el eslabón del hombro se cortaron otras dos piezas de madera MDF rectangulares con las dimensiones de 15.5 cm de largo y 9 cm de ancho y 6 mm de espesor.

El eslabón del hombro es actuado por el motorreductor de la marca Toshiba, modelo DGM-204-2A como se muestra en la figura 24, y sus características eléctricas y mecánicas se pueden consultar en la tabla 5, de igual forma los siguientes sensores utilizados. El primero es un encoder con modelo: ENA1D-481 como se muestra en la figura 25 y sus características se pueden consultar en la tabla 5. El segundo sensor es optoreflexivo modelo, QTR-1A.



Figura 24. Motor Toshiba modelo DGM-204-2A Figura 25. Encoder ENA1D-481. [26]

Las características eléctricas y mecánicas del sensor de cuadratura y el motorreductor usado en la articulación de Hombro; se muestran en la siguiente tabla.

| | |
|---------------------------------------|---------------------------|
| Encoder | ENA1D-481-L00100 |
| Alimentación | 5 volts |
| Resolución | 100 ciclos por revolución |
| Corriente | 26 mA |
| Diámetro del eje del sensor | 6.42 mm |
| Peso del sensor | 28.76 gr |
| Motorreductor | DGM-204-2A |
| Alimentación | 0 – 24 Volts |
| Corriente | 0.85 A |
| Velocidad de salida del eje del motor | 22 RPM |
| Diámetro del eje del motor | 6 mm |
| Torque | 25 kg * cm |
| Caja de engranes | 200:1 |
| Peso | 624.06 gr |

Tabla 5. Características del sensor de cuadratura y el motorreductor utilizado en la articulación del hombro.

El montaje del motorreductor Toshiba y los sensores de posición de esta parte de la estructura se puede ver en la figura 26.

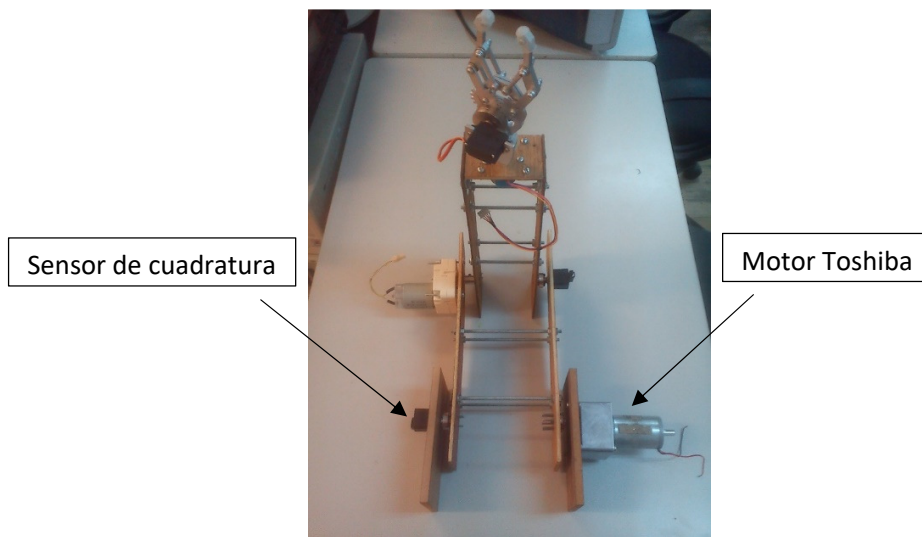


Figura 26. Montaje mecánico de la articulación del hombro.

El motor Toshiba y el sensor de cuadratura ENA1D-481 se montaron de manera que hagan, contrapeso a la estructura mecánica del robot, y que el movimiento de las articulaciones del codo y el hombro no se fueran hacia un solo lado y no hubiera un desbalance cuando el efector final haga la manipulación de objetos.

En la articulación del Hombro se colocó un sensor optoreflexivo QTR-1A, para poner el motorreductor Toshiba a una posición cero o posición de inicio, el sensor óptico se encuentra alambrado de la siguiente forma; la terminal de referencia tiene la etiqueta, **GND OPTO 3**, la alimentación se le puso la etiqueta **5 Volts OPTO 3** y el cable de la señal de salida del sensor se le coloco la etiqueta **Opto 3 IN3**. En la figura 27 se indica la regleta de conexiones el sensor óptico.

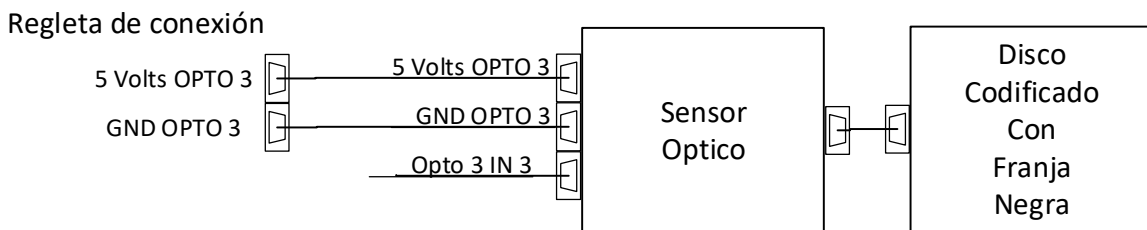


Figura 27. Conexionado de sensor QTR-1A de articulación del hombro.

El motorreductor Toshiba cuenta con dos cables que se etiquetaron así, el cable de color negro como **GND Motor 2** y el de color rojo como **12 Volts Motor 2**; que posteriormente se conectaran a un driver de potencia.

El actuador para la articulación del hombro, se va a controlar con una etapa de potencia que incluye un driver L298N, cuya finalidad es el control del sentido de giro, y regular la velocidad de giro mediante señales de entrada digitales de tipo PWM. En la figura 28 se muestra como se interconecto el driver de potencia con el motorreductor Toshiba como un subsistema que compone al hombro del robot.

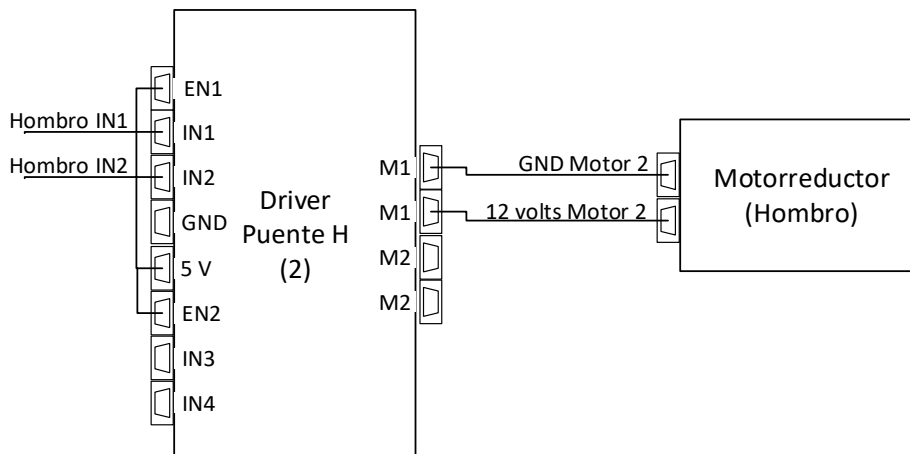


Figura 28. Conexión de etapa de potencia con motor Toshiba.

En la articulación del Hombro se colocó un sensor de cuadratura ENA1D-481 cuenta con 4 pines, dos de ellos son señales de entrada para la alimentación y los últimos dos son de señales de salida.

Se cablearon y etiquetaron de la siguiente manera: **GND ENC2**, para la terminal de referencia, **5 Volts ENC2** para la alimentación del sensor y para las señales de salida, las etiquetas que se colocaron son **ENC2 ENA** y **ENC2 ENB**. En la figura 29 se muestra sensor de cuadratura conectado en la regleta de conexiones.

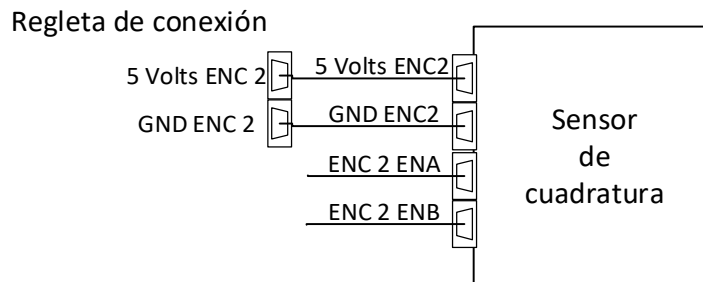


Figura 29. Conexión del encoder ENA1D-481 para articulación del hombro.

El segundo grado de libertad consiste en todo un subsistema en donde se inyectan señales digitales y lo que se obtiene físicamente es una posición angular, el hombro tiene un rango de 61° , se tiene dos sensores de posición, el primer sensor su función es comportarse como un sensor de posición cero.

El segundo sensor es un encoder, el cual nos va a indicar el número de pulsos digitales que da por una vuelta o un ciclo, conociendo la frecuencia de una de las dos señales de salida podemos determinar la velocidad de giro del motor, también se puede conocer la dirección de giro y la posición en la que se encuentra el motorreductor. En la figura 30 se muestra el subsistema en la regleta de conexiones que compone la articulación del hombro.

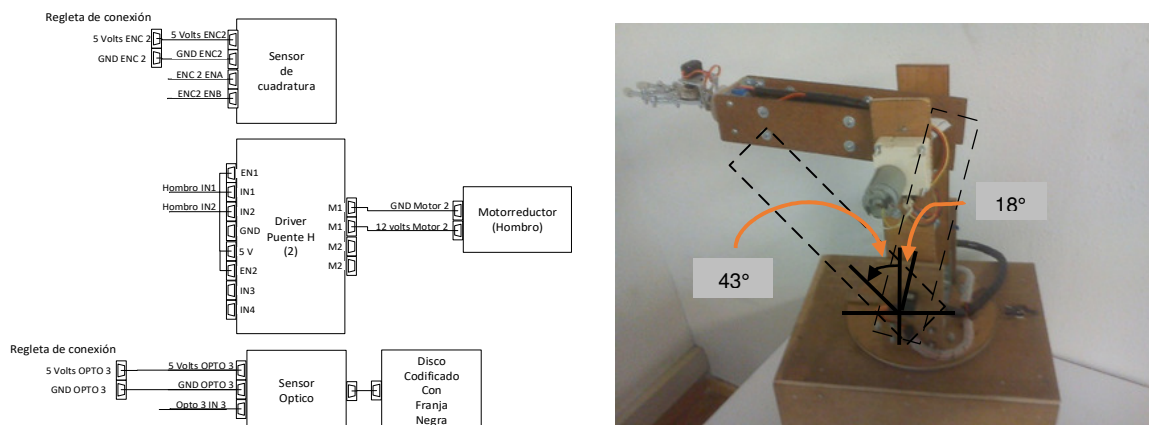


Figura 30. Conexión del subsistema que compone la articulación del hombro (izq), espacio de trabajo de la articulación del hombro (der).

3.2.5 Cintura

La cintura está compuesta por una base circular de 12 pulgadas de diámetro, montada sobre una estructura de madera de MDF de dimensiones de 45 x 45 cm, que forma parte de la base, donde en su interior estará toda la parte electrónica del brazo robótico y que en adelante se denominara **módulo de control y potencia**.

La cintura, es el primer grado de libertad del robot y consiste en un sistema en donde se inyectan señales digitales para obtener físicamente a la salida una señal de posición que va de **0 a 180 grados**, se tienen dos sensores, la función del primero es indicar el límite final de trabajo al que debe de moverse la cintura. El segundo sensor es óptico, cuya función es comportarse como un sensor de posición.

El montaje de las articulaciones sobre la base circular de madera se muestra en la figura 31.



Figura 31. Montaje mecánico de la cintura.

En esta articulación, se colocó un sensor óptico en uno de los extremos del semicírculo que conforma el disco de madera donde se montó el hombro, esto para que ese punto se utilice como una referencia para la cintura del robot e indicar mediante el conteo de pulsos la posición en la que se encuentra la cintura, respetando un espacio de trabajo para esta articulación del brazo robótico, el sensor cuenta con tres terminales de salida, etiquetados de la siguiente manera, para la alimentación del sensor se etiqueto como **GND OPTO4, 5 Volts OPTO4** y para la señal de salida se le coloco la etiqueta **Opto IN4**. En la figura 32 se ve como está el sensor óptico alambrado en la regleta de conexiones.

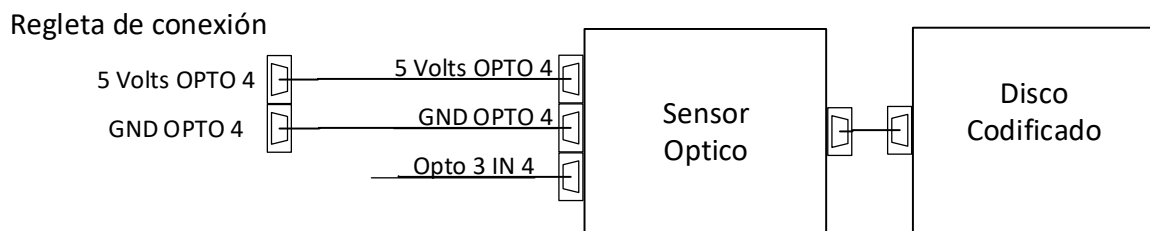


Figura 32. Conexión del sensor QTR-1A para articulación de cintura.

El motorreductor cuenta con dos cables que se etiquetaron así, el cable de color negro se etiqueto como **M3** y el de color rojo como **M3**. Que posteriormente se conectaran a un driver de potencia.

Se utilizó la tarjeta de potencia, donde están conectados los pines IN1 y IN2 de la articulación del codo, esta etapa solo va a controlar dos motorreductores de corriente directa. En la figura 33 se muestra como se interconectó el driver de potencia con el motorreductor como un subsistema que compone a la articulación del hombro.

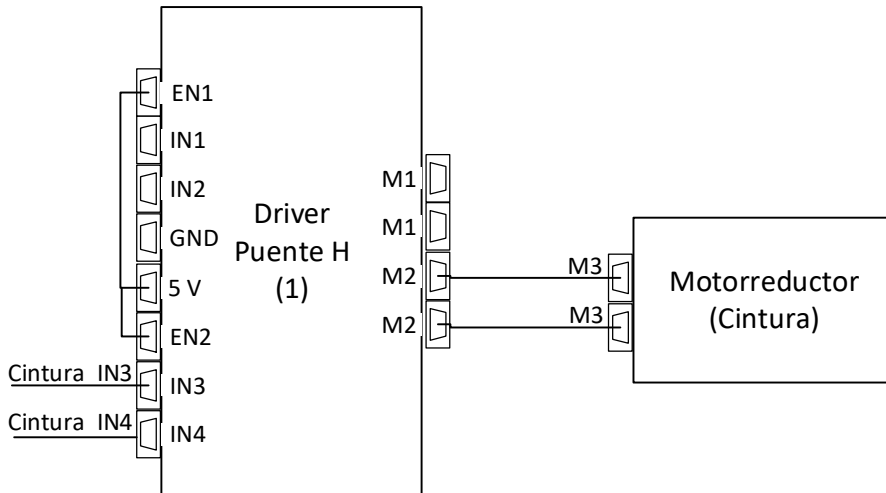


Figura 33. Etapa de potencia con motor Toshiba para articulación de cintura.

En la figura 34 se muestra el subsistema en la regleta de conexiones que compone la articulación de la cintura.

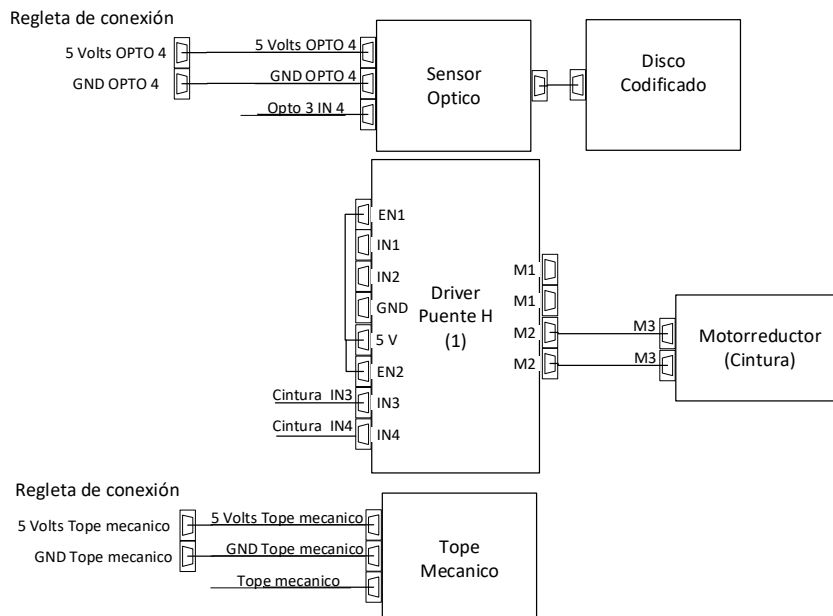


Figura 34. Conexión del subsistema que compone la articulación de la cintura.

3.3 Base para el brazo robótico

La base, en forma de caja se construyó de madera MDF con las siguientes dimensiones 30 cm de ancho, 39 cm de largo, y 32 cm de alto, como se muestra en la figura 35.

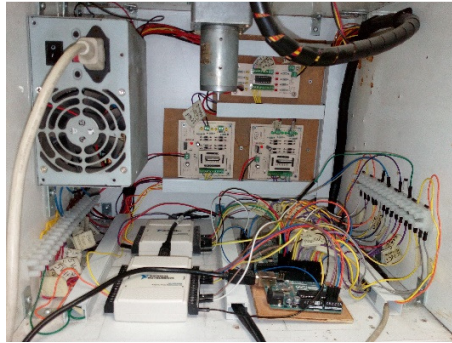


Figura 35. Electrónica para brazo robótico.

La figura 36 muestra el montaje completo del brazo robótico y la instrumentación del mismo.



Figura 36. Instrumentación de brazo robótico.

En la base se colocaron tarjetas electrónicas Arduino y USB 6009 de National Instruments, la función de las tarjetas Arduino es posicionar cada una de las articulaciones del brazo mecánico, además se encargan de la adquisición de señales provenientes de los encoders, y las tarjetas 6009 también se encargan de la adquisición de señales provenientes de los encoders y del envío de señales de referencia para controlar el sentido de giro de cada una de las articulaciones del brazo robótico.

En la base se colocó una fuente de alimentación que suministra energía eléctrica a los diferentes dispositivos electrónicos montados en las articulaciones del brazo robótico como: encoders, actuadores y etapas de potencia implementadas en el mismo.

Capítulo 4 Electrónica

Este capítulo trata de la parte electrónica contenida en la base del robot (**módulo de control y potencia**) encargada de recibir y enviar señales eléctricas a las diferentes etapas de potencia que moverán las articulaciones del brazo, mediante de National Instruments las tarjetas de adquisición de datos: Arduino UNO y USB 6009.

Además se indican las pruebas hechas para verificar el correcto funcionamiento de los sensores implementados en cada articulación del robot. En la parte final del capítulo, se muestra la interfaz gráfica hecha en Labview encargada de la comunicación con la parte de control entre la PC y el usuario.

4.1 Alimentación principal

Los voltajes que son requeridos para alimentar a sensores, actuadores y a la parte electrónica que se implementó al brazo robótico, son 5 volts y 12 volts, se utilizó una fuente de PC porque este tipo de fuentes entregan a su salida voltajes de ± 5 volts, ± 12 volts y de 3.3 volts. Y también por los valores de corrientes que entrega la fuente que es adecuada para alimentar cada uno de los dispositivos electrónicos contenidos en cada una de las articulaciones del brazo robótico, y las tarjetas de potencia. La fuente que se utilizo es de la marca LANIX como se muestra en la figura 37



Figura 37. Fuente de alimentación para brazo robótico.

Las características eléctricas de la fuente son:

- a) Entrada de alimentación de AC es de 120 – 240 Volts @ 60 Hertz
- b) Máxima salida de potencia: 180 W
- c) Para voltajes de + 3.3 V, + 5 V la potencia máxima = 125 W
- d) Para voltajes de + 3.3 V & + 5 V & + 12 V la potencia máxima es = 125 Watts
- e) Salida de voltaje: + 3.3V --- 17A, +5V --- 18A , +12V --- 10A
- 5V --- 0.2A, - 12V --- 0.3A

El valor real medidos con carga de los dos voltajes entregados por la fuente de PC, son los siguientes: 5.6 volts y 11.1 volts.

4.2 Etapas de potencia

Para las etapas de potencia se utilizaron los circuitos **ULN2803** y **L298N**, El primer circuito se escogió porque permite amplificar corrientes pequeñas, esto debido a que tiene siete pares de transistores NPN en configuración Darlington. Se compone también de diodos de fijación cátodo común para cada par Darlington NPN lo que lo hace útil para la conmutación de cargas de tipo inductivas. El **ULN2803** puede manejar una corriente máxima de 500mA por cada una de las salidas de circuito integrado. Para el motor a pasos colocado en la muñeca del brazo la corriente que requiere para poder energizar cada una de las bobinas del motor es de 92 mA y este circuito integrado es el adecuado para controlar el giro del motor a pasos. En la figura 38 se muestra el esquemático del circuito, el cual se implementó de acuerdo a la hoja de especificaciones del fabricante.

Para operar los motorreductores, se empleó como etapa de potencia el circuito, L298N conocido como "puente H", utilizando el esquemático recomendado por la hoja de datos del fabricante se construyeron dos tarjetas, se escogió ese driver porque las señales de comando (entradas) son digitales (cero y uno lógico) y permite una corriente de salida de 2 amperes por canal. En la figura 39, se muestra la conexión utilizada para este circuito.

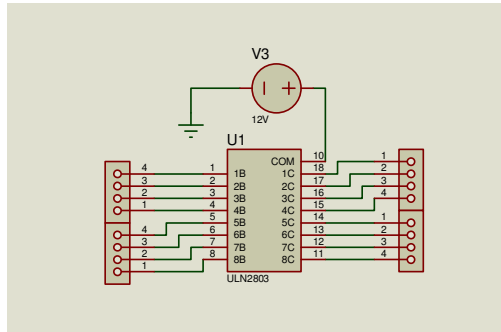


Figura 38. Esquemático de driver de potencia ULN2803.

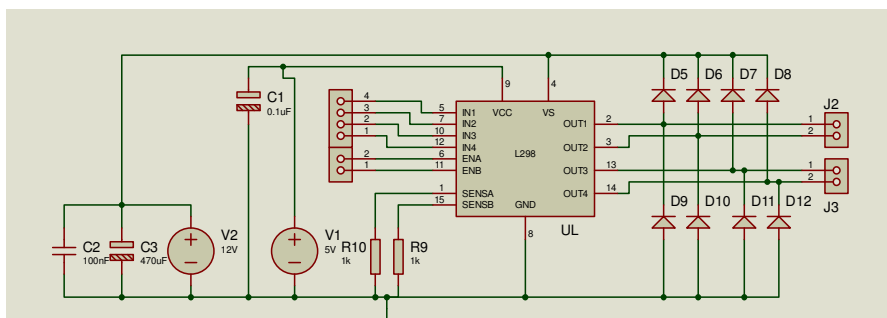


Figura 39. Esquemático de conexión de driver de potencia L298N.

Para alimentar al driver **L298N** se aplica un voltaje de 12 volts proveniente de una de las salidas de la fuente de PC, los pines **ENA** y **ENB** se conectaron a un voltaje de 5 volts, esto para habilitar el canal 1 y el canal 2 del puente H, a las entradas **IN1**, **IN2**, **IN3**, e **IN4** se les van aplicar señales de entrada de tipo PWM, estos giros dependen de las señales digitales aplicadas a cada una de las entradas, en una respectiva secuencia que se van aplicar a los motores como se muestra en la tabla 7. Para obtener una dirección de giro para cada uno de los motorreductores del codo, hombro, cintura, se muestra en la figura 40 el esquemático de cómo se conectaron los pines EN1 y EN2, y las señales que se inyectan a los pines de entrada y en donde se conectaron los motores.

| Motorreductor | Entrada | Entrada | Puente H | Dirección de giro |
|---------------|---------|---------|----------|-------------------|
| Codo | IN1 | IN2 | 1 | |
| | LOW | HIGH | | Arriba |
| | HIGH | LOW | | Abajo |
| Cintura | IN3 | IN4 | 1 | |
| | LOW | HIGH | | Izquierda |
| | HIGH | LOW | | Derecha |
| Hombro | IN1 | IN2 | 2 | |
| | LOW | HIGH | | Abajo |
| | HIGH | LOW | | Arriba |

Tabla 6. Secuencias que se deben aplicar a motores de corriente directa para obtener una dirección de giro.

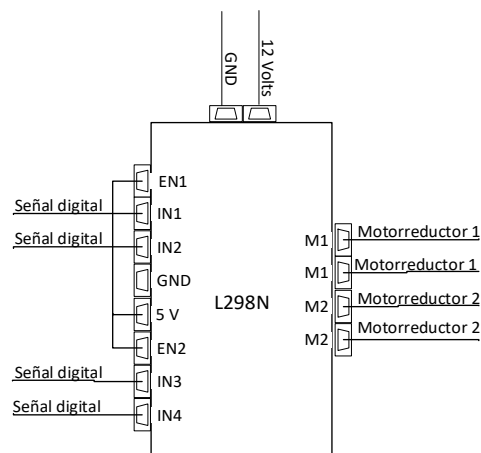


Figura 40. Conexiones de señales digitales para puente H.

4.3 Tarjetas de adquisición de datos

Ahora hablaremos de las tarjetas de adquisición de datos y las placas de desarrollo que se utilizaron para operar el brazo robótico, primero se hará una descripción de las características de la tarjeta Arduino UNO:

En la tabla 8 se muestran las características de la placa Arduino UNO utilizada para la parte de adquisición y el acondicionamiento de señales de entrada, como para el envío de señales de salida para la parte de control del brazo robótico.

| Microcontrolador | Atmega328 | Unidad |
|--|--|--------|
| Voltaje de operación | 5 | Volts |
| Voltaje de entrada (Recomendado) | 7 – 12 | Volts |
| Voltaje de entrada (Límite) | 6 – 20 | Volts |
| Pines para entrada- salida digital. | 14 (6 pueden usarse como salida de PWM) | PWM |
| Pines de entrada analógica. | 6 | |
| Corriente continua por pin IO | 40 | mA |
| Corriente continua en el pin 3.3V | 50 | mA |
| Memoria Flash | 32 KB (0,5 KB ocupados por el bootloader) | KB |
| SRAM | 2 | KB |
| EEPROM | 1 | KB |
| Frecuencia de reloj | 16 | MHz |

Tabla 7. Características técnicas de la placa Arduino UNO.

La tarjeta Arduino UNO que se utilizó es la que cuenta con un microcontrolador tipo DIP, también se utilizó una tarjeta Arduino Mega 2560 y sus características se describen en tabla 9.

| Microcontrolador | Atmega2560 | Unidad |
|--|---|--------|
| Voltaje de operación | 5 | Volts |
| Voltaje de entrada (Recomendado) | 7 – 12 | Volts |
| Voltaje de entrada (Límite) | 6 – 20 | Volts |
| Pines para entrada- salida digital. | 54 (15 pueden usarse como salida de PWM) | PWM |
| Pines de entrada analógica. | 16 | |
| Corriente continua por pin IO | 40 | mA |
| Corriente continua en el pin 3.3V | 50 | mA |
| Memoria Flash | 256 KB (8 KB ocupados por el bootloader) | KB |
| SRAM | 8 | KB |
| EEPROM | 4 | KB |
| Frecuencia de reloj | 16 | MHz |

Tabla 8. Especificaciones técnicas de la tarjeta Arduino mega 2560.

La tarjeta NI USB 6009 es una tarjeta de adquisición de datos multifuncional para Windows 2000/XP/Vista, MAC OS X, LINUX; posee alto rendimiento y alta velocidad de muestreo y de bajo costo. Las tarjetas USB 6009 que se emplearon para esta tesis son dos, y en la tabla 10 se muestran sus características con la que cuenta dicha tarjeta:

| Tarjeta | USB 6009 | Unidad |
|-------------------------------|---|------------|
| Entradas Analógica | 8 | |
| Entradas Digital | 8 | |
| Salidas Analógicas | 2 | |
| Salidas digitales | 8 | |
| Convertidor A/D | 14 | bits |
| Rango de muestreo | 5 | MHz |
| Rangos de entradas analógicas | Unipolares ± 10 Diferenciales $\pm 20, \pm 10, \pm 5, \pm 4, \pm 2.5, \pm 2, \pm 1.25, \pm 1.$ | Volts |
| Resolución de entrada | 14 bits en modo unipolar 13 bits en modo diferencial | bits |
| Tasa de muestreo | 48 | KS/s |
| Impedancia de entrada | 144 | K Ω |
| Protección de sobretensión | ± 35 | Volts |

Tabla 9. Especificaciones técnicas de la tarjeta de adquisición de datos USB 6009.

En la figura 41 se muestran las tarjetas de adquisición de datos utilizadas para el control del robot.

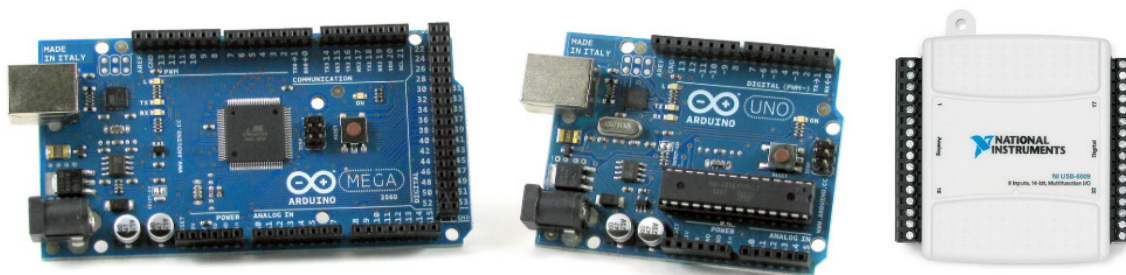


Figura 41. Placas de desarrollo Arduino, tarjetas de adquisición de datos USB 6009.

[27] [28]

4.4 Pruebas del sistema

Las pruebas que se le hicieron al brazo robótico antes de realizar la programación fueron las siguientes. Primero se checo la continuidad de las señales de alimentación eléctrica provenientes de la fuente de PC, encargadas de energizar a los diferentes tipos de sensores implementados en el brazo robótico. Después se revisó la continuidad de señales entrada y salida provenientes de los sensores que van a estar conectados a las tarjetas Arduino, USB 6009, y a las etapas de potencia, siguiendo el diagrama de Gantt que se realizó en Microsoft office llamado Visio mostrado en la parte de los anexos de la tesis.

Se hicieron programas de prueba en el IDE de Arduino, para verificar el correcto funcionamiento de los sensores ópticos y de cuadratura. La adquisición de las señales de entrada provenientes de los sensores ópticos son señales de tipo analógica y mediante programación se configuraron los sensores ópticos para comportarse de forma digital, esto por la aplicación que se le va a dar en el brazo robótico. Para los sensores de cuadratura de igual forma se tuvo que hacer un acondicionamiento de señales convirtiendo los pulsos digitales que entregan a grados; esto para saber en qué posición se encuentra el encoder.

4.5 Posición inicial de articulaciones para brazo robótico

Para el posicionamiento inicial de cada una de las articulaciones que componen al brazo robótico se hizo la programación individual, comenzando con el cuarto grado de libertad que es la muñeca.

El programa consiste primero en hacer que el motor a pasos gire a la izquierda, esto de acuerdo a la secuencia descrita de la tabla 3 que se mencionó en la sección 3.2.2 del capítulo 3. Cuando el sensor óptico detecte un pulso digital en “Alto” es decir un “1” lógico, el motor a pasos de la muñeca debe detenerse y posteriormente cambiar el sentido de giro usando la secuencia de la tabla 4, y mediante la programación tiene que hacer un conteo de pasos que previamente se le asignó al programa para que cuando llegue a ese número de pasos le indique al motor de la muñeca que ahí debe detenerse indicándonos que ésa articulación llego a su referencia inicial.

En la figura 42 se muestra el esquemático de conexiones, de los pines de la tarjeta de Arduino Mega utilizados para la programación del posicionamiento inicial para la articulación de la muñeca.

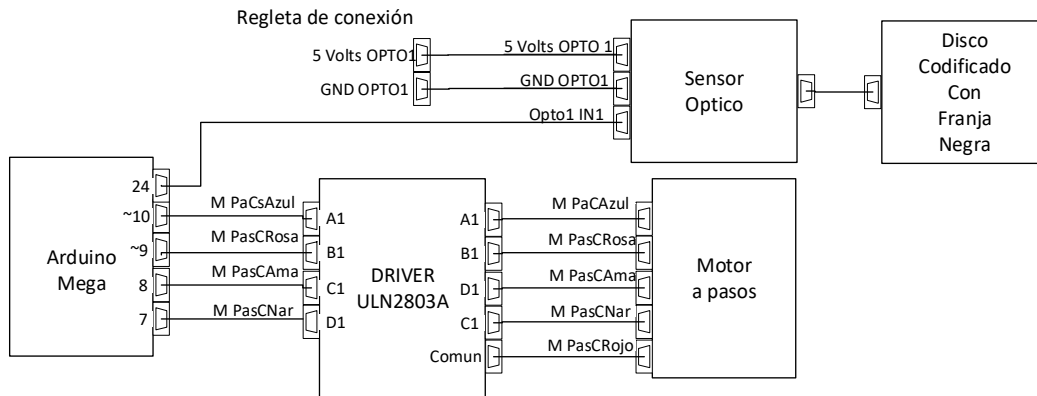


Figura 42. Esquemático de conexiones para control de giro para articulación de la muñeca.

Para el posicionamiento inicial del tercer grado de libertad; que en este caso es la articulación del codo, se realizó un programa para que el motorreductor comience a girar en sentido anti horario, aplicándole una señal digital de tipo PWM en el pin de entrada IN2 de 150 y al pin IN1 se le coloco una señal PWM de 0. El valor PWM de 150 que se le aplico a la entrada IN2 del puente H es debido a que con ese valor, el giro del motorreductor es el adecuado para que el sensor óptico detecte el cambio de tonalidad de color blanco a negro, y cuando el sensor óptico que se colocó en esta articulación detecte un pulso digital en “alto” en el pin 26 de la tarjeta Arduino Mega, el programa le va a indicar al motorreductor que cambie el sentido de giro. Esto se hace invirtiendo la secuencia en el pin de entrada IN1 aplicándole una señal PWM de 100 y al pin IN2 una señal PWM de 0; el motorreductor debe girar a una cierta posición que se le indica por software y detenerse cuando el encoder que está montado en esta articulación llegue a su posición de referencia.

En la figura 43 se muestra el esquemático, que indica cómo está la conexión de los pines de la tarjeta de Arduino Mega que se utilizó para la programación del posicionamiento inicial para la articulación del codo.

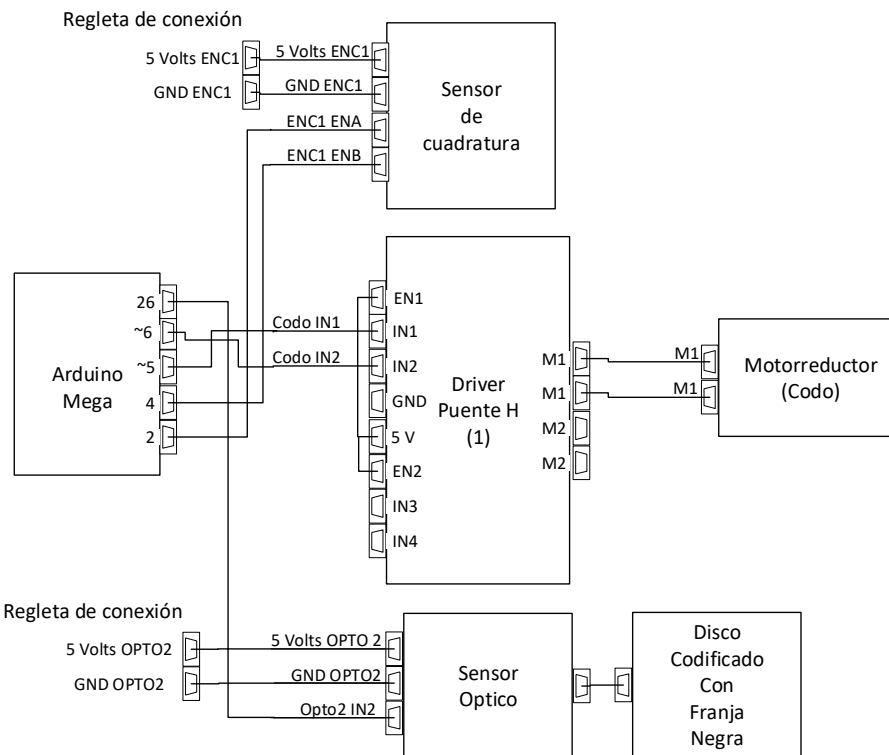


Figura 43. Esquemático de conexiones para control de giro para articulación del codo.

Para el posicionamiento inicial del segundo grado de libertad (Hombro) el programa lo que hace es que el motorreductor comience a girar en sentido anti horario, aplicando una señal digital de tipo PWM en el pin de entrada IN1 de 255 y la entrada IN2 se le coloco una señal PWM de 0. El valor PWM de 255 es el adecuado para que el motorreductor mueva la articulación del hombro. Cuando el sensor óptico que se colocó en esta articulación, detecte un pulso digital en “alto” en el pin 7 de la tarjeta Arduino UNO, el programa le va a indicar al motorreductor que cambie el sentido de giro, esto se consigue aplicándole una señal PWM de 100 al pin IN2 y al pin IN1 una de 0. Así, el motorreductor debe girar hasta llegar a su posición de referencia previamente programada.

En la figura 44 se muestra el esquemático de conexiones, en la cual indica los pines de la tarjeta de Arduino UNO que se utilizaron para realizar el posicionamiento inicial para la articulación del hombro.

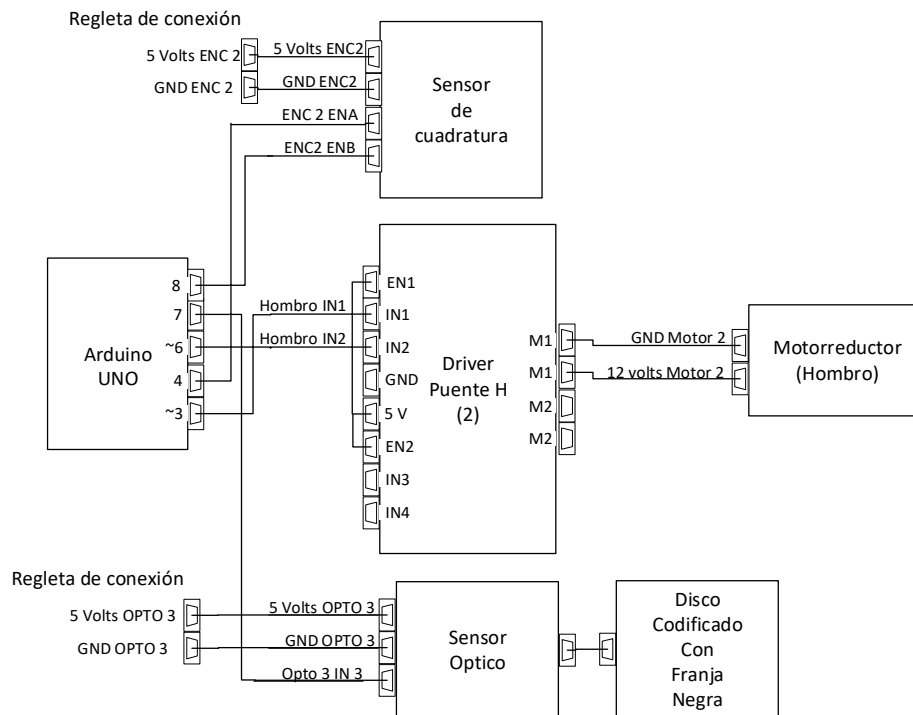


Figura 44. Esquemático de conexiones para control de giro de articulación del hombro.

Para el posicionamiento inicial del primer grado de libertad; la cintura, se realizó un programa para que el motorreductor comience a girar hacia la izquierda, aplicando una señal digital de tipo PWM en el pin IN3 con el valor de 0 y la entrada IN4 se le colocó una señal PWM de 100. El valor PWM de 100 es el adecuado para que el sensor óptico detecte las franjas de color negro y blanco que se pusieron en el disco de madera donde está montada la estructura mecánica del brazo robótico. Cuando el tope mecánico colocado en esta articulación detecte un pulso digital en "alto", en el pin 11 de la tarjeta Arduino UNO, el programa le va a indicar al motorreductor que cambie el sentido de giro, invirtiendo la secuencia en el pin IN3 con una señal PWM de 100 y al pin IN4, una señal PWM de 0. El motorreductor debe girar a una cierta posición la cual se le indicó en el programa y detenerse esto gracias al sensor óptico y al disco de la cintura que funciona como un encoder simple, indicando que la cintura llegó a su posición de referencia.

En la figura 45 se muestra el esquemático de conexiones, en la cual indica los pines de la tarjeta de Arduino UNO que se utilizaron para realizar el posicionamiento inicial de la cintura.

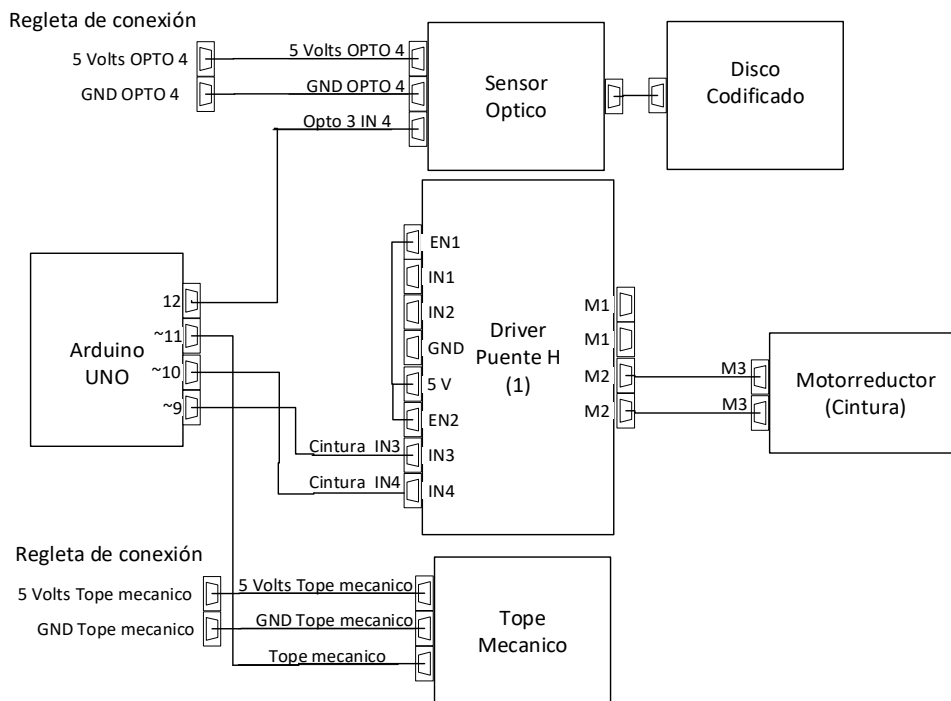


Figura 45. Esquemático de conexiones para control de giro de articulación de la cintura.

Como se vio en este apartado, las dos tarjetas de Arduino, se utilizan, para colocar en una referencia inicial a dos articulaciones del brazo robótico, en cambio, el Arduino Mega, va a encargarse de posicionar el codo y la muñeca. El Arduino UNO va a posicionar el hombro y la cintura, el orden en que deben de posicionarse estas articulaciones es el siguiente: primero el hombro, después la cintura, posteriormente el codo y por último la muñeca.

Se consideró este orden porque cuando el brazo robótico es utilizado para alguna aplicación, las articulaciones van a quedar en diferentes posiciones, es por esto que se eligió este orden. En los siguientes apartados de este capítulo se explicara cómo se realiza el posicionamiento de estas articulaciones; esto mediante una interfaz gráfica en el software de Labview.

4.6 Descripción del software de programación LabView

Es un programa que fue creado por National Instruments en 1976 para funcionar sobre MAC y salió al mercado en 1986. Lo que hace de LabView un programa distinto a otros es su entorno de programación se basa en el lenguaje Gel el cual permite trasladar casi literalmente el algoritmo en forma gráfica para la programación. El código se desarrolla mediante la interconexión de bloques funcionales, similar a un diagrama de flujo o de bloques, lo que lo convierte en un entorno de programación más amigable.

Los programas de LabView son llamados instrumentos virtuales “Vis” ya que su aspecto y funcionamiento imita a instrumentos físicos. Un instrumento virtual a diferencia de un instrumento físico, permite personalizar completamente el objeto final que conseguir. LabView permite generar instrumentos virtuales que puedan realizar distintos tipos de funciones y además se cuenta con el poder de procesamiento de un ordenador. Los Vis son módulos pueden cumplir distintas funciones a la vez, se componen de “Panel Frontal” y “Diagrama de bloques”.

La versión que se utiliza es LabView 2014, Al ingresar al programa usted se encontrará con la ventana de inicio, como se muestra en la figura 46.

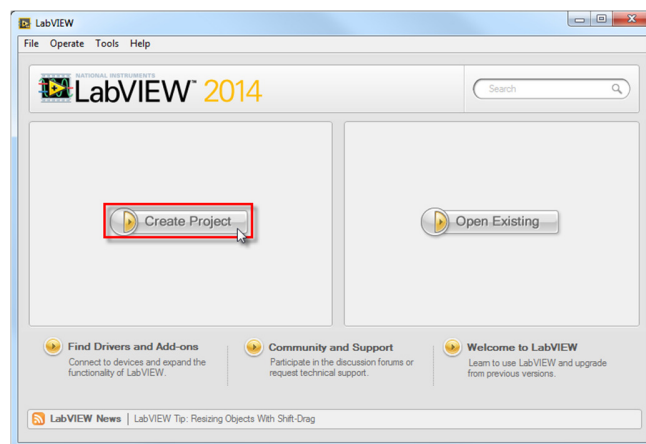


Figura 46. Ventana de inicio del programa Labview versión 2014.

4.7 Posición inicial por interfaz gráfica en Labview y Arduino

La interfaz gráfica que se implementó; fue colocar un botón llamado **POSICION INICIAL**, la función de dicho botón es posicionar a una referencia inicial cada una de las cuatro articulaciones del brazo robótico.

Las señales que va a entregar la tarjeta USB6009 es de tipo digital, para enviar una señal digital al pin 26 de la tarjeta Arduino Mega 2560 encargada del envío y de recibir señales digitales de la tarjeta Arduino UNO que se configuraron vía software como entradas y salidas, para que ejecuten subprogramas que coloquen estas articulaciones a una referencia inicial.

Las interconexiones que se hicieron en la base del robot para cumplir el objetivo de posicionar las 4 articulaciones en una referencia inicial, se describen en el anexo 1.

En cuanto a la programación de las tarjetas Arduino se realizó una configuración maestro-esclavo donde el maestro va hacer **Arduino Mega 2560** y el esclavo es **Arduino UNO**, la placa Arduino Mega se va a encargar de coordinar como se va a realizar la ejecución de cada programa almacenado incluido en los subprogramas que van a estar en la función **void loop**, asignada a cada una en las placas Arduino, para efectuar de forma secuencial el posicionamiento de cada una de las articulaciones del brazo robótico a su posición inicial conforme al orden mencionado en el tema 4.5, del capítulo 4 de la tesis, esto mediante la adquisición y envío de señales eléctricas de control de tipo digital “1” y “0” lógico provenientes de la placa Arduino UNO,

El funcionamiento de la programación de la posición inicial usando la interfaz gráfica en Labview, se realiza de la siguiente manera: al oprimir el botón virtual colocado en el panel frontal de la interfaz gráfica, es decir cuando termine de posicionar a cada una de las articulaciones del brazo robótico a una referencia, y posteriormente se esté moviendo alguna de las articulaciones mediante esta interfaz gráfica, si se vuelve a apretar este botón llamado **Posición Inicial**, el robot lo que va hacer es detener lo que se esté ejecutando en ese momento, para volver a posicionar dichas articulaciones a su referencia inicial.

Capítulo 5 Programación y control

En este capítulo se describirán cada uno de los objetivos de movimiento para cada una de las articulaciones que componen el brazo robótico, y como se realizó la programación para resolver cada uno de estos problemas de movimiento de forma individual. Se explica cómo es que está conformado el panel de control virtual realizado en Labview, tanto para controlar cada uno de los movimientos de giro a la derecha e izquierda de cada uno de los actuadores de las articulaciones, como la adquisición de datos provenientes de los sensores colocados en el brazo robótico.

5.1 Objetivos de control para las articulaciones de brazo robótico.

Las 4 articulaciones que son: muñeca, codo, hombro y cintura se recapitula indicando que se tienen 5 objetivos de control:

- a) El primero es apertura y cierre de los dos dedos de la pinza que en términos de control significa mover un servo.
- b) El segundo es giro de muñeca, que implica mover el motor a pasos
- c) El tercero es codo, implica mover un motor de dc con reducción
- d) El cuarto es hombro, implica mover un motor de dc con reducción
- e) El quinto es cintura, implica mover un motor de dc con reducción

Los 5 objetivos de control se reducen a tres problemas de control, el problema de control para el brazo robótico, es que hay que mover 4 articulaciones, de estas 4 articulaciones una se reduce al control de un servomotor, a un control de motor a pasos y las otras se reducen a un control de motor de corriente directa con reducción, el control para estas articulaciones se va a realizar en lazos abiertos.

5.1.1 Control en lazo abierto de efector final – Pinza

En este apartado, se explica el procedimiento hacer girar el eje del servomotor para la apertura y cierre de la pinza, la operación del servomotor consiste en los elementos que conforman el diagrama de bloques que se muestra en la figura 47.

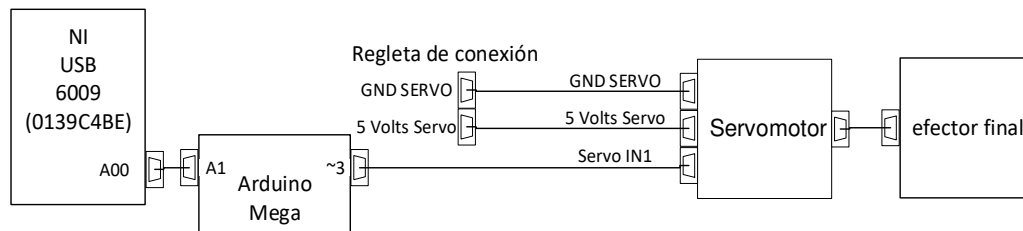


Figura 47. Diagrama de bloques para control de apertura y cierre de efector final.

El sistema utilizado en la pinza es un lazo abierto, según el diagrama de bloques de la figura 47, se tiene que inyectar una señal de referencia de tipo analógica proveniente de la tarjeta USB 6009 modelo “0139C4BE” a un pin de entrada analógica que mediante la programación realizada en el IDE de Arduino, envía una señal de control saliente del pin 3 de la tarjeta Arduino MEGA hacia el servomotor permitiéndole poder hacer girar el eje del servomotor que a su vez esta interconectado a una pinza para realizar la apertura y cierre.

El sistema utilizado en la pinza es un lazo abierto, para la programación se usó la librería `servo.h` para establecer una posición. Esta librería cuenta con las funciones **`attach(int)`** y **`write(int)`**, que en apartados posteriores se explican en qué consisten. En el programa se definen variables para almacenar la información de una entrada que en este caso va hacer de tipo analógica, proveniente de una barra virtual colocada en el panel frontal de LabView y una de las variables que se definieron se va a utilizar para almacenar el valor del Angulo, que va hacer el encargado posteriormente en el programa de mandar la señal de control PWM, para posicionar el eje del servomotor a la posición que se le indique mediante una barra virtual.

El valor de la señal de control de salida que va a enviar la tarjeta Arduino Mega al servomotor va de 0 a 100 %; para el cierre o apertura de la pinza del brazo robótico. Esta relación se ve reflejada en la siguiente grafica; de la figura 48 la cual indicar el comportamiento del servomotor.

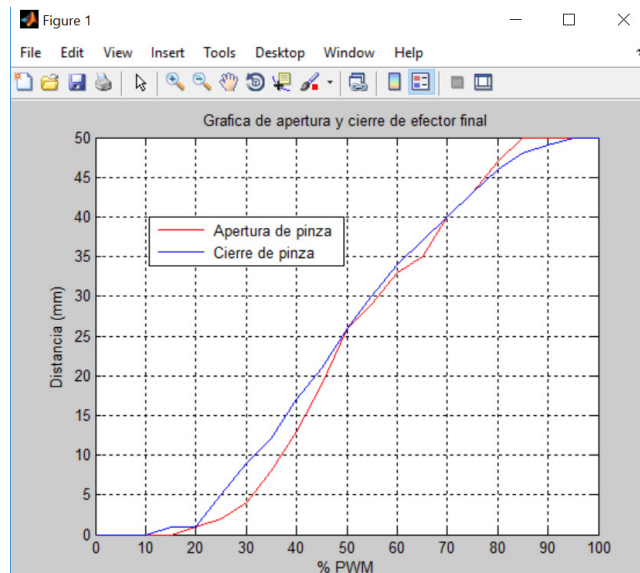


Figura 48. Grafica de apertura y cierre del efector final.

5.1.2 Dirección de giro para motor a pasos

Para controlar la dirección de giro del motor a pasos; implementado como elemento que compone la articulación de la muñeca del brazo robótico, la operación de giro de la muñeca consiste en los elementos que conforman el diagrama de bloques que se muestra en la figura 49.

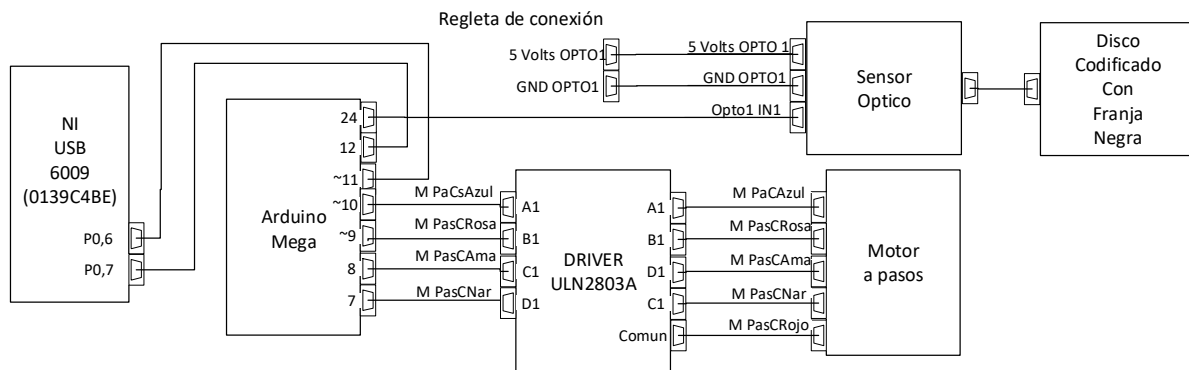


Figura 49. Diagrama para control de giro para articulación de la muñeca.

Para establecer el giro del motor, el programa consiste en inyectar dos señales digitales de dos bits, que provienen de dos botones virtuales que se colocaron en el panel frontal de la interfaz gráfica de LabView, estas dos señales digitales provenientes de los pines P0,6 y P0,7, de la tarjeta USB 6009 con número de serie "0139C4BE" están configuradas como salidas de tipo digital, y para la tarjeta Arduino Mega los pines 11 y 12 se van a configurar como entradas digitales, que dependiendo los pulsos que se envíen ya sea en alto o en bajo, el driver de potencia le va a indicar al motor a pasos que va tener un movimiento en sentido horario o anti horario. El motor a pasos solo podrá moverse de -90 a 90 grados debido a la limitación del cableado del servomotor de la pinza que impide que la articulación de la muñeca gire los 360 grados. Las secuencias de giro del motor a pasos se muestran en la tabla 3.

5.1.3 Funcionamiento del motor de corriente directa con reducción

Otro de los objetivos es el posicionamiento de las articulaciones que contienen motores de corriente directa con reductor de velocidad; estas articulaciones son codo hombro y cintura del brazo robótico. Los diagramas de bloques que conforman a cada uno de los sistemas se muestran en las siguientes figuras.

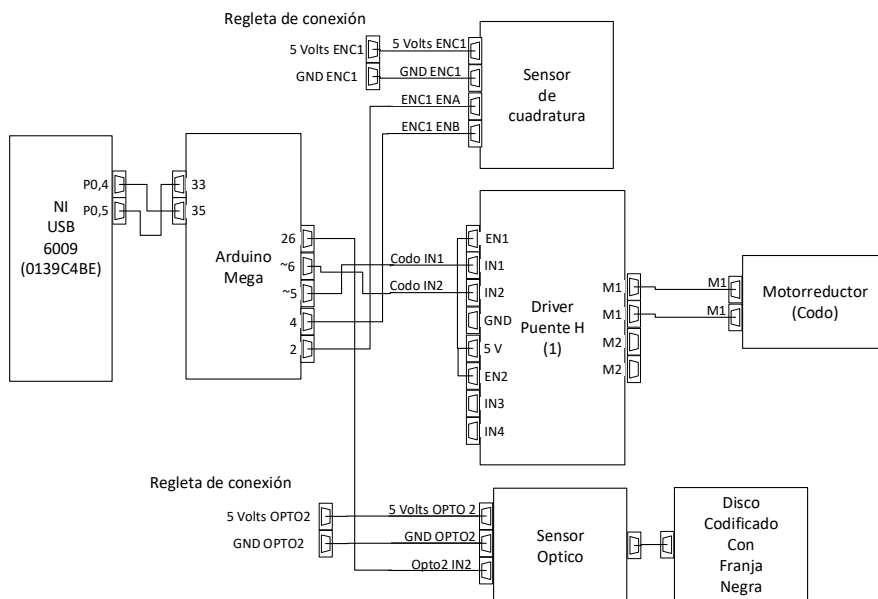


Figura 50. Diagrama para control de articulación del codo.

La figura 50 muestra las conexiones eléctricas de cada una de las etapas que componen al sistema de control de la articulación del codo.

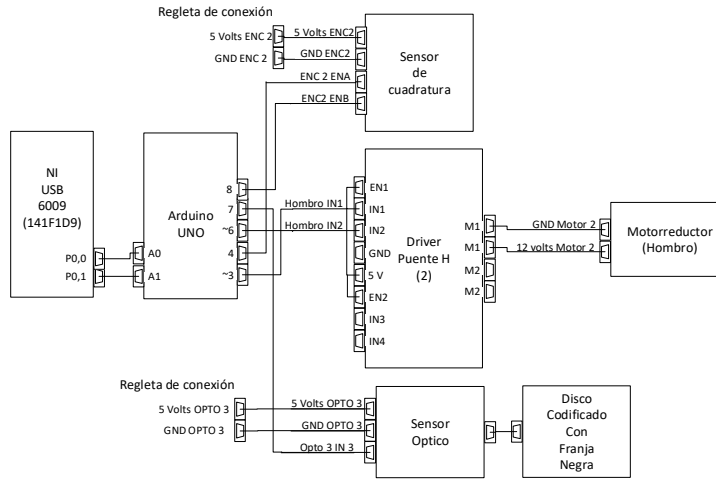


Figura 51. Diagrama para control de articulación del hombro.

En la figura 51 muestra las interconexiones que conforman cada una de las etapas que componen al sistema de control de la articulación del Hombro.

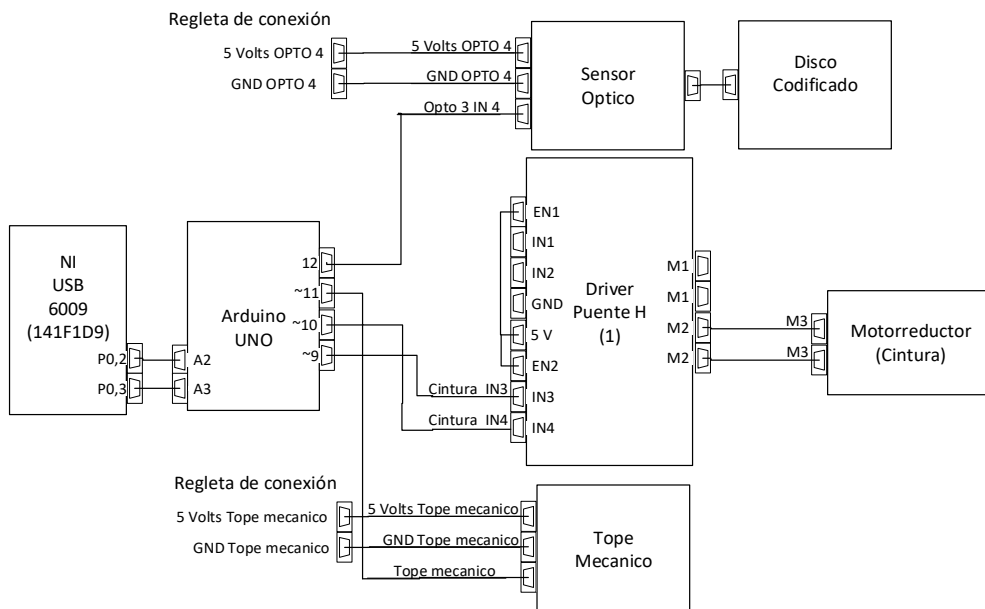


Figura 52. Diagrama para control de articulación de la cintura.

En la figura 52 muestra las interconexiones que conforman cada una de las etapas que componen al sistema de control la articulación de la Cintura.

Antes de revisar la parte de la programación, se dará una explicación de cómo está conformado el sistema de control para que mediante software y hardware en aplicaciones posteriores se realice control de posición del robot manipulador.

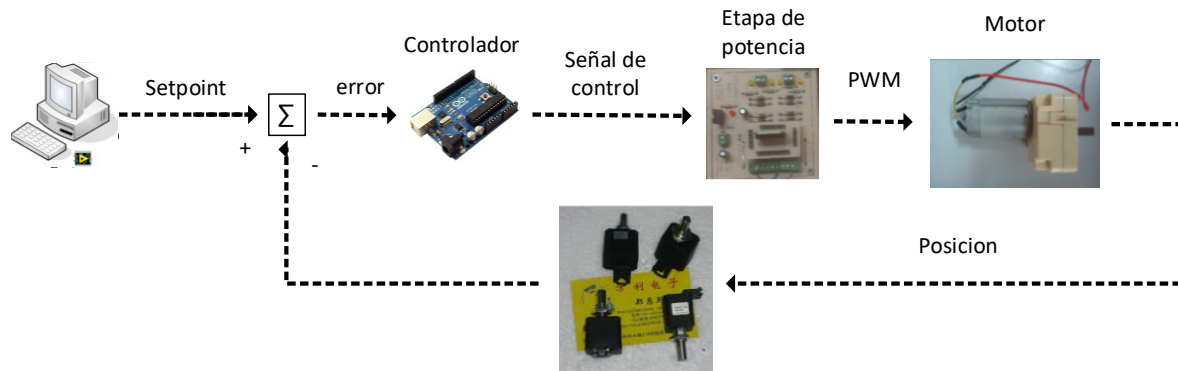


Figura 53. Diagrama en lazo cerrado para control de posición para motores de corriente directa de brazo robótico.

Los movimientos de giro para los motores de corriente directa, como se muestra en cada una de las figuras anteriores, se envían señales de entrada de tipo digital proveniente de las tarjetas de adquisición de datos USB 6009 y mediante la programación implementada en cada una de las tarjetas Arduino se manipula la dirección de giro para los actuadores de las articulaciones del CODO, HOMBRO, y CINTURA del robot.

5.2 Descripción del software del sistema

La parte de control del posicionamiento inicial para las articulaciones del brazo robótico se realizan por medio de las tarjetas Arduino que utilizan el software Arduino IDE; siendo éste una herramienta para escribir y cargar a la tarjeta el software de forma sencilla. Arduino IDE es software escrito en Java, lo cual permite exportar más fácilmente a otras plataformas y es software tipo open-source (código abierto).

En este apartado se va a describir brevemente el software que se encarga de controlar la posición inicial de la 4 articulaciones del brazo robótico, como se describió en apartados anteriores, la parte del sistema de control de posición inicial es la colocación de un botón virtual en el panel frontal de LabView, que se encarga de posicionar inicialmente las articulaciones del brazo a una referencia inicial.

Además se le colocaron botones virtuales que mueven cada una de las articulaciones el brazo, a una cierta posición a la que el usuario le indique mediante la interfaz virtual.

También la interfaz gráfica cuenta con indicadores luminosos para indicar que articulación se está moviendo y en qué dirección lo está haciendo en ese momento.

5.3 Funciones utilizadas

En la programación realizada para las tarjetas Arduino, se usaron dos librerías, que son la Servo.h. Para el control del servomotor se usó la librería Servo.h que nos ofrece diferentes funciones para controlar la posición del mismo.

Servo.h

attach(int): Para poner un pin en modo servo drive.

detach() : Libera un pin del modo servo driver.

write(int): Indica el ángulo a girar el servo, 0 a 180 grados.

read(): Obtiene el ultimo valor enviado (posición del servo).

attached(): Devuelve 1, si el servo está conectado.

refresh(): Se debe llamar a ésta función al menos cada 50ms para asegurarse que los servos estarán en su posición.

5.4 Interfaz de usuario en LabView para brazo robótico

La interfaz virtual es la conexión entre la computadora y el brazo robótico dando una comunicación entre los distintos niveles de acciones. En la PC se dan las instrucciones en el programa de LabView, estas instrucciones las recibe la interfaz en señales digitales las cuales se van a usar para el control del brazo robótico, también la programación en LabView va a enviar señales de tipo analógico y digital a las etapas que conforman la parte que va a controlar los movimientos de las articulaciones del robot manipulador. De modo general, la interfaz se compone de un solo panel principal: Modo manual.

El panel denominado *Modo manual* se desarrolló un programa en LabView cuyo objetivo es que el brazo robótico sea manipulable por un operador o usuario, en este panel se encuentran los controles asociados a cada una de las articulaciones del brazo robótico.

La interfaz gráfica va a interactuar con el usuario mediante el uso de un dibujo del brazo mecánico acompañado por una serie de una barra y botones virtuales. Esta interfaz cuenta con 4 botones virtuales, como se mencionó en el apartado 4.7 la interfaz cuenta con un botón virtual que al presionarlo produce movimiento de las articulaciones del brazo robótico para colocar cada una de ellas a una referencia inicial, la interfaz cuenta con una barra vertical, para que el usuario lo deslice, se colocaron dos leds en los extremos de la imagen de la pinza o efector final, estos leds se van a encender cada vez que el efector final del brazo mecánico se abra o se cierre, indicando al usuario que movimiento está realizando la pinza.

Para la muñeca del brazo mecánico, el usuario puede indicarle el sentido de giro a esta articulación, al apretar uno de los dos botones virtuales llamados **IZQUIERDA** y **DERECHA**, cuya función es solo indicar el sentido de giro para la articulación de la muñeca

Para la articulación del Codo, el usuario tiene que oprimir dos botones virtuales llamados **IZQUIERDA** y **DERECHA**, estos botones van a mover el codo del brazo robótico a la izquierda o derecha y el dibujo del brazo va a encender alguno de los dos indicadores leds, para mostrar la dirección de giro del motor de corriente directa.

Las articulaciones del Hombro y la Cintura tienen por igual dos botones virtuales llamados **IZQUIERDA** y **DERECHA** para cada una de estas dos articulaciones. El usuario puede oprimir alguno de los botones para mover alguna de estas articulaciones en alguna dirección deseada. En la imagen del brazo robótico solo se indicara con los leds virtuales la dirección de giro de los motores para cada una de las articulaciones.



Figura 54. Interfaz gráfica de usuario para control de giro de cada una de las articulaciones de brazo robótico.

La interfaz gráfica cuenta con los botones de **Arranque**, **Paro**, **Adquisición de señales de sensores**. La función del botón de arranque es inicializar la interfaz gráfica. El botón de paro deja que termine el movimiento que esté ejecutando el robot en ese momento y después detiene el brazo robótico, el botón de adquisición de señales de sensores nos va a mostrar en el panel frontal el monitoreo mediante graficas de las señales provenientes de los sensores ópticos, también muestra con despliegues numéricos la información proveniente de los encoders de cuadratura implementados en el robot manipulador, como son: dirección de giro proveniente del canal B del encoder, la posición en la que se encuentra el encoder en ese momento en cada una de las articulaciones etc.

En la figura 54 se muestra la interfaz hecha en LabView, la cual tiene los controles asociados a cada una de las articulaciones del brazo manipulador, que permiten mover los diferentes motores.

Capítulo 6 Pruebas y resultados

En este capítulo se hablará sobre las pruebas de funcionamiento realizadas al robot manipulador con el uso de la interfaz gráfica hecha en LabView, con la finalidad de verificar que el usuario, mediante esta interfaz, tenga la capacidad de posicionar el brazo robótico a las 4 posiciones iniciales definidas a cada una de las articulaciones del robot en su respectivo espacio de trabajo, después verificar que el usuario pueda manipular cada una de las articulaciones del robot de forma independiente mediante el uso de la botonera virtual y que, a su vez, pueda visualizar la información de las señales provenientes de cada uno de los sensores del robot mediante los despliegues numéricos y gráficos como parte de la adquisición de datos implementado en la interfaz gráfica.

El software de Arduino permite el acceso a la señal PWM, que es la señal que permite variar la velocidad de giro para cada una de las articulaciones del robot manipulador que le permita realizar los movimientos de rotación y traslación para sujetar un objeto y colocarlo en otro punto dentro de su espacio de trabajo.

6.1 Pruebas de brazo robótico con la interfaz grafica

Las pruebas que se realizaron al robot manipulador con la interfaz gráfica de LabView son:

- 1.- Prueba de posicionamiento inicial de cada una de las articulaciones del brazo robótico usando el botón virtual llamado **Posición Inicial** o **"HOME"**, que se describe en el anexo 5 de la guía de usuario.
- 2.- Prueba de uso para los botones virtuales encargados de realizar los movimientos de giro en sentido horario y anti horario para la muñeca, codo, hombro y cintura.
- 3.- Pruebas para evaluar de manera independiente cada uno de los sensores, mediante la visualización de la información en la interfaz gráfica y el monitor serial de Arduino, solo permite revisar la información uno por uno.

4.- Prueba de adquisición de datos usando las tarjetas USB 6009 y las tarjetas Arduino encargadas del monitoreo de las señales de los sensores provenientes de cada una de las articulaciones del robot, para comprobar que mediante la interfaz gráfica de LabView y el monitor serial se visualice la información al mismo tiempo conforme se mueve cada una de las articulaciones provenientes de los sensores.

5.- Prueba de uso de los botones virtuales para colocar el efector final del robot, en cualquier lugar dentro de su espacio de trabajo, para sujetar objetos y mediante la interfaz gráfica manipular los movimientos de rotación y traslación de dicho objeto para ubicarlo en otra posición.

6.2 Ajuste de la velocidad del movimiento de las articulaciones

El sistema está ajustado de manera tal que las velocidades de movimiento de las articulaciones son fijas, gracias a los ajustes realizados a las señales PWM para cada una de las articulaciones que nos permiten ajustar las variaciones de la velocidad de giro que se encuentran definidos en el software de Arduino. Como un trabajo futuro en todo lo que hay que hacer es acceder a la manipulación de la señal PWM de los Arduinos.

6.3 Resultados

Una vez que se terminaron de realizar las pruebas hechas al robot manipulador mencionadas en las secciones: 4.4, 4.5, del capítulo 4, 6.1 del capítulo 6, se hicieron los ajustes necesarios para obtener un mejor funcionamiento del robot.

Antes de describir los resultados recomendamos revisar la guía de usuario que se encuentra en el anexo 5 con la finalidad de familiarizarse con el entorno del panel frontal de la interfaz gráfica para el brazo robótico. A continuación se describen los resultados obtenidos:

El resultado de la prueba al utilizar la interfaz al apretar el botón llamado **Posición Inicial**, lo que se observa físicamente es: el robot detiene lo que esté realizando en ese momento para después regresar a su posición inicial, y en la fotografía del robot manipulador que se encuentra en el panel frontal de la interfaz se observa un indicador luminoso en color verde asociado a la activación del botón **Posición Inicial** como se muestra en la figura 55.



Figura 55. Activación de botón virtual Posición Inicial.

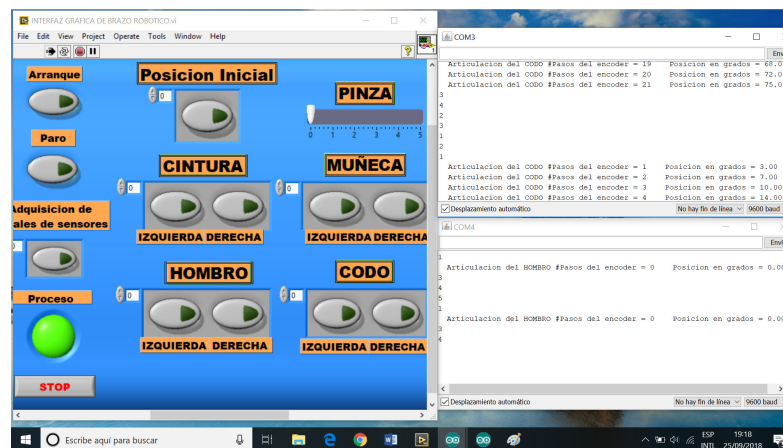


Figura 56. Visualización de Posicionamiento inicial usando interfaz gráfica inicial.

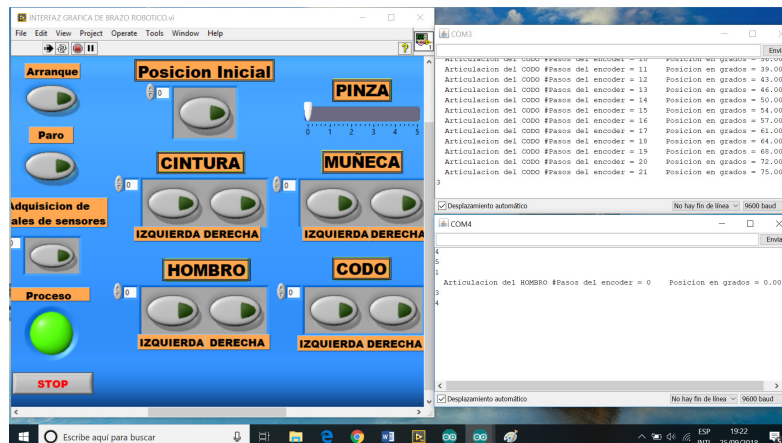


Figura 57. Visualización de Posicionamiento inicial usando interfaz gráfica final.

En la figura 56 se visualiza que al oprimir el botón virtual de **Posición Inicial**, el monitor serial **COM 3** muestra cómo se lleva a cabo la ejecución del posicionamiento de las articulaciones del codo y muñeca, y en **COM 4** se visualiza como se ejecuta la secuencia para posicionar el hombro y la cintura, en la figura 58 se observa el monitor serial **COM 3** como se termina de ejecutar el posicionamiento de la articulación del codo a una referencia asignada por software y después coloca la muñeca igualmente a referencia inicial también asignada por software, el **COM 4** se muestra el posicionamiento del hombro a su respectiva referencia inicial para posteriormente posicionar la cintura del robot a su respectiva referencia inicial.

Al realizar la prueba de usar la botonera virtual para mover cada una de las articulaciones de forma independiente, cuando se mueve la pinza lo que se observa en la fotografía del robot, es la activación y la desactivación de cada uno de los dos indicadores luminosos que se encuentran sobre la imagen de la pinza conforme se está abriendo o cerrando.

Si mediante la interfaz se mueve la articulación del codo en el panel frontal de LabView se observa lo siguiente: en la fotografía del brazo se encenderán los indicadores luminosos colocados en la imagen donde se encuentra el codo solo para indicar en qué dirección se está moviendo el codo.

Al mover el hombro usando la interfaz gráfica se observa primero que se encenderán los indicadores luminosos colocados cerca del hombro en la fotografía del robot indicando en qué dirección se está moviendo el hombro.

Cuando se mueve la cintura por medio de la interfaz gráfica, en el panel frontal de LabView se observa como los indicadores luminosos se van a encender, conforme se opriman los botones virtuales asignados para mover la cintura indicando en qué dirección se está moviendo la cintura, como se muestra en la fotografía del robot colocada en la interfaz.

En la figura 58 se observa que al ir moviendo cada una de las articulaciones usando la interfaz gráfica, se va a ir encendiendo cada indicador luminoso conforme el usuario oprima los botones virtuales indicando que articulación se quiere mover y en qué dirección lo va a realizar, y físicamente el robot se va a ir moviendo con forme se le indique mediante la interfaz gráfica.



Figura 58. Encendido de indicadores luminosos al mover cada articulación mediante la interfaz gráfica.

La prueba de adquisición de datos provenientes de las señales de los sensores optorefectivos, y sensores de cuadratura mediante el uso de la interfaz, al mover cada articulación mediante la botonera virtual en el mismo panel frontal de la interfaz los resultados son:

En la interfaz cuando se mueve la muñeca, en el panel frontal de LabView se muestra la gráfica de la señal analógica proveniente del sensor óptico, también en el panel se visualiza el despliegue numérico llamado **Sensor1 Muñeca** para observar el valor escalar (volts), en el que se encuentra en ese momento la señal proveniente del sensor óptico.

Si mediante la interfaz se mueve la articulación del codo se muestra en panel frontal a lado derecho de la botonera varios despliegues numéricos que nos van indicar el número de pasos que está dando en ese momento el encoder, la posición angular (en grados), en la que se encuentra el codo y la distancia (en centímetros) que recorre el codo. También se visualiza en el waveform chart la gráfica de la señal analógica proveniente del sensor óptico, como su valor escalar (en volts), que tiene el sensor óptico, mediante el indicador numérico que se encuentra en el panel frontal de LabView.

Al mover el hombro usando la interfaz gráfica se muestran también varios despliegues numéricos donde se visualiza el número de pasos, la posición angular (en grados) y la distancia (en centímetros) que recorre el hombro y su valor escalar (en volts), adquiridos por el encoder, También se visualiza en el waveform chart la gráfica de la señal analógica proveniente del sensor óptico, como su valor escalar (en volts), que tiene el sensor óptico, mediante el indicador numérico que se encuentra en el panel frontal de LabView.

Cuando se mueve la cintura por medio de la interfaz gráfica en la interfaz se visualizara la gráfica de la señal analógica de sensor óptico, como el despliegue numérico en el que se visualiza el valor escalar (en volts) del sensor óptico.

Al realizar la prueba de visualización de la adquisición de datos provenientes de las señales de los sensores optorefectivos, y sensores de cuadratura usando la interfaz gráfica al mover el robot el resultado obtenido fue que en la parte derecha de la botonera virtual se observa que realmente en la interfaz gráfica se puede ver que se logró poder monitorear la información adquirida proveniente de cada uno de los sensores colocados en el robot.

En la figura 59 se observa el monitoreo que hace la interfaz gráfica de LabView de las señales provenientes de los encoders, sensores ópticos al ir moviendo cada una de las articulaciones del robot por medio de la botonera virtual.

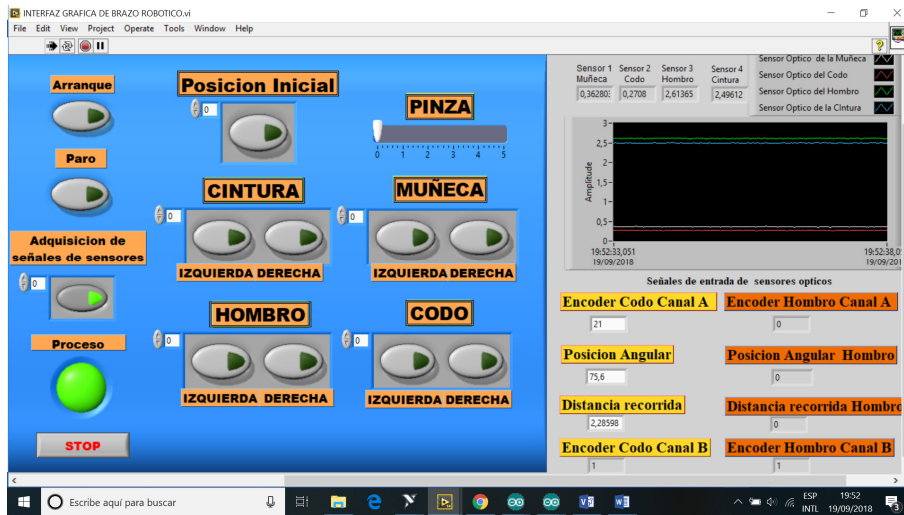


Figura 59. Monitoreo de señales de sensores en interfaz gráfica de LabView.

Para la prueba de adquisición de datos usando las tarjetas USB 6009 y las tarjetas Arduino para visualizar la información al mismo tiempo conforme se mueve cada una de las articulaciones el resultado obtenido es: se logró visualizar como conforme se le indica por medio de la interfaz que se mueva alguna de las articulaciones del robot, se puede observar el monitoreo de la información tanto en el panel frontal de la interfaz como en el monitor serial de Arduino provenientes de los sensores.

En la figura 60 se muestra el monitoreo de las señales del encoder y su sensor óptico que se implementó en la articulación del codo del robot manipulador como ejemplo de la adquisición de datos, realizados por la tarjeta DAQ 6009 " 0139C4BE " y la tarjeta Arduino Mega, cuando la articulación se coloca en su respectiva posición inicial.

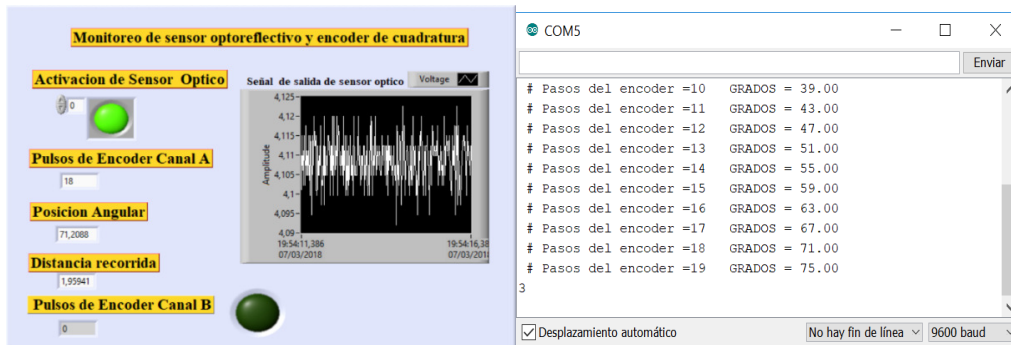


Figura 60. Monitoreo de señales de sensores con LabView y Arduino.

En la prueba de uso de los botones virtuales para colocar el efector final del robot, en un lugar dentro del espacio de trabajo, para sujetar un objeto, y mediante la interfaz gráfica realizar los movimientos de rotación y traslación de este objeto para ponerlo en otra posición el resultado obtenido es: el brazo robótico tiene la capacidad que mediante el uso de la interfaz gráfica puede moverse de tal manera que el usuario pueda colocar el efector final en una posición para poder sujetar un objeto y ponerlo en otro punto donde el usuario desee ponerlo siempre y cuando se respete el espacio de trabajo al cual se puede mover el robot manipulador. En la figura 61 se visualiza como el brazo robótico realice la sujeción del objeto para que mediante la interfaz el usuario manipule al robot para colocar este objeto en otra posición.

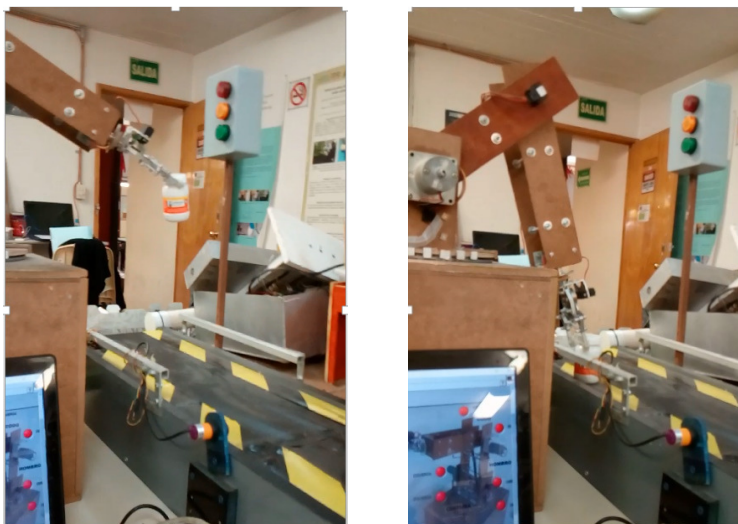


Figura 61. Prueba de sujeción de objetos por medio de la interfaz gráfica.

A lo que se refiere a la construcción de la parte mecánica del presente trabajo se pueden mencionar algunas características: el robot es de configuración angular o antropomórfica de 4 grados de libertad, su implementación mecánica del robot cuenta con motorreductores de corriente directa, motores a pasos, servomotor, tiene una capacidad de carga para sujetar objetos con pesos menores a 100 gramos, es de fácil instalación (por su tipo de alimentación y por su tamaño), el robot esta retroalimentado por encoders de cuadratura en las articulaciones del codo, y el hombro, también esta retroalimentado con un encoder simple la articulación de la cintura.

El rango de trabajo donde puede moverse cada articulación del robot es: la muñeca es de 180 grados, el codo es de 113 grados, el hombro es de 61 grados, y la cintura es 180 grados.

Conclusiones

Se entrega un prototipo de robot manipulador de 4 grados de libertad para uso didáctico y funcional. El prototipo es de bajo costo, véase **ANEXO 2**, de arquitectura abierta y de fabricación simple, entregándose una interfaz gráfica que permita al usuario interactuar con sus articulaciones y efector final utilizarlo para recoger objetos, como se menciona en la sección 3.2.1 del capítulo 3 respetando el espacio de trabajo del robot.

Se entregó un manual de usuario, **ANEXO 5**, para el correcto uso del robot manipulador. También se presentan los programas que actualmente tienen almacenados en la memoria interna de cada una de las tarjetas Arduino como se describe en el **ANEXO 4**, se entrega el diagrama esquemático de conexiones mostrado en el **ANEXO 1**, se proporcionan los diagramas de circuito impreso de la electrónica de potencia implementada en el *módulo de control y potencia* mostrado en el **ANEXO 3**, todo esto para que se puedan construir más prototipos o hacer las modificaciones que sean pertinentes.

Finalmente el robot manipulador está listo para ofrecer una plataforma en donde el usuario experimente mediante el uso de tarjetas de adquisición de datos y placas de desarrollo, tenga la oportunidad de aplicar sus conocimientos adquiridos en clase, y aplicarlos en proyectos prácticos donde el brazo robótico requiera ser implementado para realizar alguna tarea en específico.

Trabajos futuros

Unas de las mejoras que se le pueden hacer en cuanto a su construcción, es el uso de materiales ligeros, por ejemplo el aluminio es un material fácil de conseguir y es liviano, tomar en cuenta todos los pesos en cada una de las articulaciones y torques para cada uno de los eslabones para no tener problemas mecánicos, esto con la finalidad de tener un mejor diseño en cuanto a su construcción del robot manipulador.

Al brazo robótico se le puede agregar más instrumentación externa, que permita al robot tener más percepción del entorno como son: sistema de visión para aplicaciones de inspección y control de calidad, sensores de proximidad capacitivos para la detección de diferentes tipos de objetos, ya sean metálicos o no metálicos como papel, vidrio, tela, etc.

En cuanto a su control del brazo robótico, como se mencionó en apartados anteriores las tarjetas Arduino y las tarjetas Daq 6009 se encargan del monitoreo de las diferentes señales que entregan cada uno de los sensores con los que cuenta el robot manipulador se pueden diseñar diferentes controladores digitales, utilizando los toolkits de control que tiene LabView para realizar los diagramas de bloques correspondientes que se encarguen del control de posición para el sistema completo del robot o para controlar la posición de cada una de las articulaciones del robot.

Bibliografía

- [1] (s.f.) Índice de robótica. Recuperado el 10 de abril del 2018, de <http://www.etitudela.com/profesores/rpm/rpm/downloads/robotica.pdf>
- [2] Robótica. (2012). [versión electrónica]. Recuperado 12 de abril del 2018, de <https://nextcomrobotics.wordpress.com/campo-de-aplicacion/aplicacion-de-la-robotica/>
- [3] Universidad de Costa rica. Industria 4.0: la era de la ciberfísica. s.f [versión electrónica]. Recuperado 12 de abril del 2018, de <https://www.ucr.ac.cr/noticias/2018/08/13/industria-4-0-la-era-de-la-ciberfisica.html>
- [4] Víctor R. González Fernández. (2002). Aplicaciones de la robótica [versión electrónica]. Recuperado 12 de abril del 2018, de http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/aplicaciones.htm
- [5] Robótica. (2012). [versión electrónica]. Recuperado 12 de abril del 2018, de <https://nextcomrobotics.wordpress.com/campo-de-aplicacion/impacto-de-la-robotica/>
- [6] Roboticspot, “Definición y clasificación de la robótica”, (2004), España, [versión electrónica]. Recuperado 12 de abril del 2018, de https://www.researchgate.net/figure/Figura-11-Robot-industrial-tomado-de-1_fig1_46283245
- [7] Empresas. Robotización industrial. (2018). [versión electrónica]. Recuperado 17 de abril del 2018, de <http://ferrepro.mx/robotizacion-industrial/>
- [8] Definición de Mecanismo Eslabón. s.f. [Versión electrónica]. Recuperado 17 de abril del 2018, <https://conceptodefinicion.de/mecanismo/>
- [9] Víctor R. González Fdez. Antonio López Cruzado, José A. Cabero Esteban Robots. Robots industriales. s.f. [versión electrónica]. Recuperado 17 de abril del 2018, http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.4.htm
- [10] mootio components. Motorreductor DC 6V 19rpm. s.f. [versión electrónica]. Recuperado 17 de abril del 2018 http://www.mootio-components.com/motorreductor-dc-6v-19rpm_ref_011813-6.html#.W7I5qWhKg2w
- [11] Fernando Carmona. (2011). Software, Robótica y Tecnología. [Versión electrónica]. Recuperado el 17 de abril del 2018, de <http://bitsbycode.blogspot.com/2012/03/como-fabricar-poleas-dentadas.html>
- [12] Víctor R. González. (2002). Estructura de un robot industrial. [Versión electrónica]. Recuperado 17 de abril del 2018, http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/sistema/morfologia.htm

- [13] Introducción. s.f. [Versión electrónica]. Recuperado 17 de abril del 2018, <http://repositorio.upct.es/bitstream/handle/10317/133/Cap%EDtulo1.PDF;jsessionid=FD063146AC64590E6979A400B4D82B5?sequence=4>
- [14] Tipos de Movimiento y Grados de Libertad. (2010). [Versión electrónica]. Recuperado de 20 de agosto del 2018. <https://es.slideshare.net/EducaRedColombia/tipos-de-movimiento-y-grados-de-libertad>
- [15] Miguel Ángel Márquez Aguilera, Héctor Octavio Girón Bobadilla. (2017). Grados de libertad. [Versión electrónica]. Recuperado de 20 de agosto del 2018 <https://freedomforlife.wordpress.com/grados-de-libertad/>
- [16] Paco. (2013). Estructura de los robots manipuladores. [Versión electrónica]. Recuperado de 20 de agosto del 2018 <https://coparoman.blogspot.com/2013/05/estructura-de-los-robots-manipuladores.html>
- [17] (s.f.) Configuración cartesiana. [Versión electrónica]. Recuperado de 20 de agosto del 2018 <http://www.educa.madrid.org/web/ies.silveriolanza.getafe/Ens/DptoTecnologia/fprob/RobotCartesiano.htm>
- [18] Wikilibros. (2018). Robótica/Configuraciones de un robot. [Versión electrónica]. Recuperado de 20 de agosto del 2018 https://es.wikibooks.org/wiki/Rob%C3%B3tica/Configuraciones_de_un_robot
- [19] Camilo Baquero Jiménez, Jonathan Moncada Acosta, Nicolás Escobar Álzate. s.f. Estructura Mecánica de un Robot. [Versión electrónica]. Recuperado de 20 de agosto del 2018. <https://roboticapuj.es.tl/Estructura-Mec%C3%A9nica-de-un-Robot.htm>
- [20] Aníbal Ollero Baturone. (2001). Robótica Manipuladores y robots móviles. (pag. 23). Editorial marcombo.
- [21] (s.f.) Configuración cartesiana. Recuperado el 10 de abril del 2018, de <http://www.educa.madrid.org/web/ies.silveriolanza.getafe/Ens/DptoTecnologia/fprob/RobotCartesiano.htm>
- [22] Sparkfun. s.f. Robotic claw – MKII. [Versión electrónica]. Recuperado de 20 de agosto del 2018. <https://www.sparkfun.com/products/retired/11524>
- [23] tdrobotica Co. s.f. Pinza robot - Dagu. [Versión electrónica]. Recuperado de 20 de agosto del 2018. <http://tdrobotica.co/pinza-robot-dagu/476.html>
- [24] Martín Mellado Arteché (2014). Robótica Edición original publicada por Universidad Politécnica de Valencia (pag. 74 – 78). Editorial LIMUSA.
- [25] aelectronics s.f. Motor a pasos 28BYJ-48 5V. [Versión electrónica]. Recuperado de 20 de agosto del 2018. https://a-electronics.com.mx/index.php?id_product=97&controller=product

[26] AliExpress. s.f. Nuevo y Original del 100% ENA1D-481 L00100 0808 M 4 línea codificador fotoeléctrico. [Versión electrónica]. Recuperado de 20 de agosto del 2018. <https://es.aliexpress.com/item/Original-new-100-ENA1D-481-L00100-0808M-4-line-photoelectric-encoder/32615609864.html>

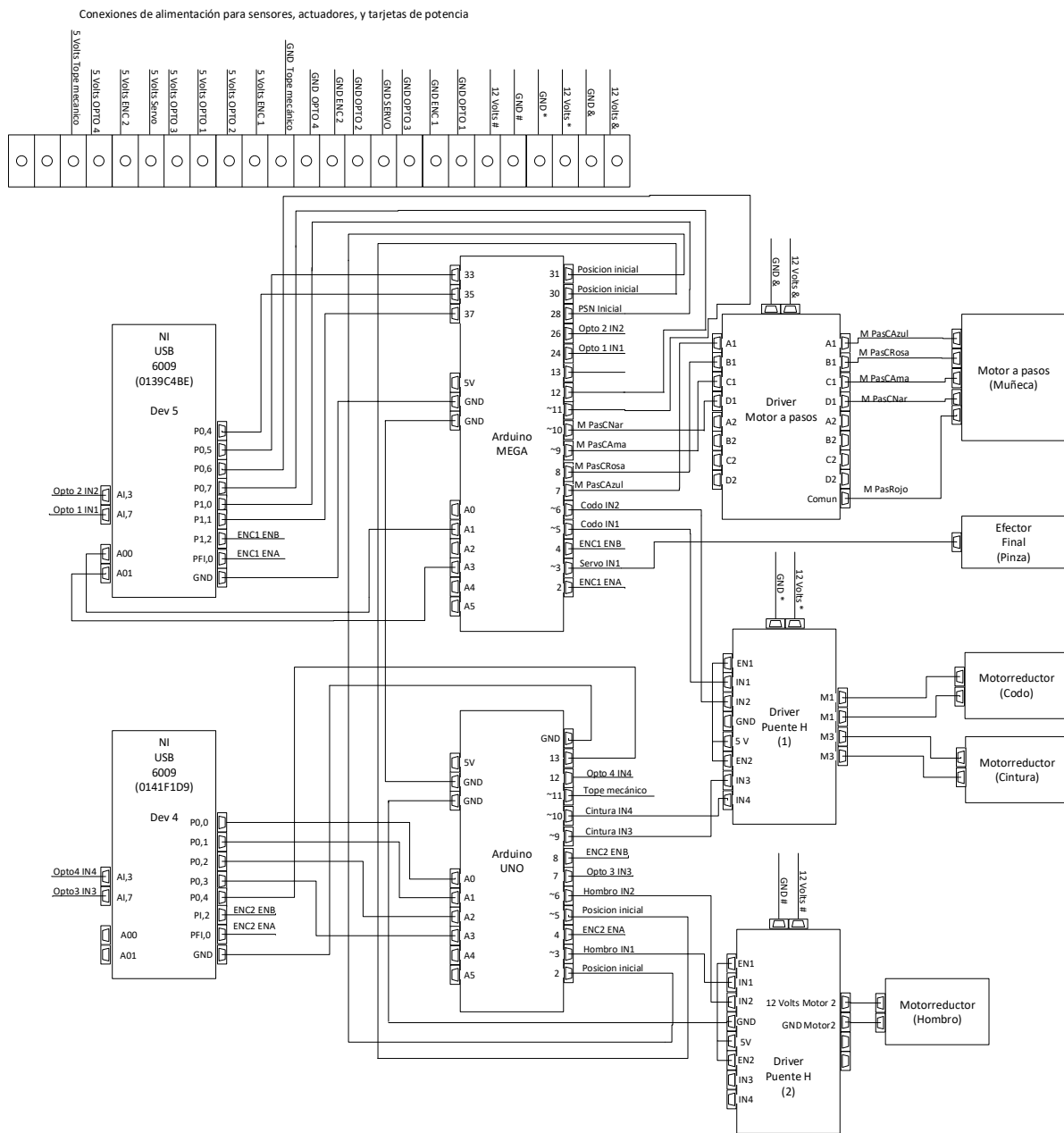
[27] RobotShop inc. S.f. [Versión electrónica]. Recuperado de 20 de agosto del 2018. <https://www.robotshop.com/blog/en/the-arduino-uno-and-mega2560-are-here-917/arduino-uno-mega2560-front>

[28] AliExpress. s.f. Excelente estado usado, la tarjeta de adquisición de datos USB NI USB-6009 analógica original DAQ 8AD 2DA 14bit 48kS / s Labview, o reemplazo [Versión electrónica]. Recuperado de 20 de agosto del 2018. <https://ru.aliexpress.com/item/1pcs-Used-Excellent-Condition-Original-NI-USB-6009-USB-data-acquisition-card-DAQ-8AD-2DA-14bit/32771734880.html>

ANEXO 1

1.1 Diagrama esquemático de conexiones eléctricas y electrónicas

Para la conformación del control del brazo robótico necesitamos conocer cada uno de los elementos que conforman el sistema y como están interconectados entre si y a continuación se muestra el diagrama de conexiones.



1.2 Lista y significado de las etiquetas colocadas en el esquemático.

Las etiquetas que se colocaron en cada uno de los cables de la partes de alimentación eléctrica, como la parte electrónica y a los diferentes dispositivos electrónicos como son sensores y actuadores que componen al brazo robótico, es necesaria conocer el etiquetado de los elementos mencionados anteriormente para facilitar la identificación de entradas y salidas que componen al sistema del robot esto para revisar e identificar más rápido algún elemento que este fallando en el sistema.

Lista y significado de las etiquetas colocadas en el esquemático

| | |
|------------------------------|---|
| <u>12 Volts &</u> | Alimentacion de driver para motor a pasos |
| <u>GND &</u> | GND para driver de motor a pasos |
| <u>12 Volts *</u> | Alimentación de Driver Puente H (1) |
| <u>GND *</u> | GND para Driver Puente H (1) |
| <u>12 Volts #</u> | Alimentación de Driver Puente H (2) |
| <u>GND #</u> | GND para Driver Puente H (2) |
| <u>GND OPTO 1</u> | GND Sensor óptico conectado a articulación de la MUÑECA |
| <u>GND ENC 1</u> | GND Encoder conectado a articulación del CODO |
| <u>GND OPTO 3</u> | GND Sensor óptico conectado a articulación del HOMBRO |
| <u>GND SERVO</u> | GND Servomotor conectado a efector final |
| <u>GND OPTO 2</u> | GND Sensor optico conectado a articulacion del CODO |
| <u>GND ENC 2</u> | GND Encoder conectado a articulación del HOMBRO |
| <u>GND OPTO 4</u> | GND Sensor optico conectado a articulacion de la CINTURA |
| <u>GND Tope mecanico</u> | GND Tope mecanico conectado a articulacion de la CINTURA |
| <u>5 Volts ENC 1</u> | 5 Volts Encoder conectado a articulación del CODO |
| <u>5 Volts OPTO 2</u> | 5 Volts Sensor óptico conectado a articulación del CODO |
| <u>5 Volts OPTO 1</u> | 5 Volts Sensor óptico conectado a articulación de la MUÑECA |
| <u>5 Volts OPTO 3</u> | 5 Volts Sensor óptico conectado a articulación del HOMBRO |
| <u>5 Volts Servo</u> | 5 Volts Servomotor conectado a efector final |
| <u>5 Volts ENC 2</u> | 5 Volts Encoder conectado a articulación del HOMBRO |
| <u>GND USB 6009 (1)</u> | GND De tarjeta USB 6009 (1) a Arduino UNO (1) |
| <u>GND USB 6009 (2)</u> | GND De tarjeta USB 6009 (2) a Arduino UNO (2) |
| <u>M.PasAzul</u> | Señal de salida para motor a pasos (cable color Azul) |
| <u>M.PasRosa</u> | Señal de salida para motor a pasos (cable color Rosa) |
| <u>M.PasAma</u> | Señal de salida para motor a pasos (cable color Amarillo) |
| <u>M.PasNar</u> | Señal de salida para motor a pasos (cable color Naranja) |
| <u>M.PasRojo</u> | Señal de salida para motor a pasos (cable color Rojo) |
| <u>Servo IN1</u> | Señal de control (pulso enviado al servomotor) |
| <u>Opto 1 IN1</u> | Señal de entrada proveniente de sensor optico conectado a articulacion de la MUÑECA |
| <u>Opto 2 IN2</u> | Señal de entrada proveniente de sensor optico conectado a articulacion del CODO |
| <u>Opto 3 IN3</u> | Señal de entrada proveniente de sensor optico conectado a articulacion del HOMBRO |
| <u>ENC 1 ENA</u> | Señal de entrada proveniente de Encoder Canal A conectado a articulacion del CODO |
| <u>ENC 1 ENB</u> | Señal de entrada proveniente de Encoder Canal B conectado a articulacion del CODO |
| <u>ENC 2 ENA</u> | Señal de entrada proveniente de Encoder Canal A conectado a articulacion del HOMBRO |
| <u>ENC 2 ENB</u> | Señal de entrada proveniente de Encoder Canal B conectado a articulacion del HOMBRO |
| <u>Codo IN1</u> | Señal de salida para motor de DC (IN1 driver Puente H (1)) conectado a articulacion del CODO |
| <u>Codo IN2</u> | Señal de salida para motor de DC (IN2 driver Puente H (1)) conectado a articulación del CODO |
| <u>Hombro IN1</u> | Señal de salida para motor de DC (IN1 driver Puente H (2)) conectado a articulacion del HOMBRO |
| <u>Hombro IN2</u> | Señal de salida para motor de DC (IN2 driver Puente H (2)) conectado a articulacion del HOMBRO |
| <u>Cintura IN3</u> | Señal de salida para motor de DC (IN3 driver Puente H (1)) conectado a articulacion del CINTURA |
| <u>Cintura IN4</u> | Señal de salida para motor de DC (IN4 driver Puente H (1)) conectado a articulacion del CINTURA |
| <u>Opto 4 IN4</u> | Señal de entrada proveniente de sensor óptico conectado a articulación de la CINTURA |
| <u>Tope mecanico</u> | Señal de entrada proveniente de la articulación de la CINTURA |
| <u>5 Volts OPTO 4</u> | 5 Volts Sensor óptico conectado a articulación de la CINTURA |
| <u>5 Volts Tope mecanico</u> | 5 Volts para tope mecánico conectado a articulación de la CINTURA |

ANEXO 2

2.1 Análisis de costos

Para la construcción e implementación de este proyecto se utilizaron los siguientes materiales, los cuales generan costos que están relacionados directamente con la fabricación del mismo entre estos tenemos los costos mecánicos, eléctricos, electrónicos y de mano de obra.

Lista de materiales mecánicos

| ítem | Descripción | Cantidad | P.Unitario Pesos | P.Total Pesos |
|------|--|----------|---------------------|------------------|
| 1 | Madera MDF | 1 | 214 | 214 |
| 2 | Madera Triplay | 1 | 30 | 30 |
| 3 | Madera Aglomerado | 1 | 40 | 40 |
| 4 | Base giratoria con baleros de 12 pulgadas | 1 | 140 | 140 |
| 5 | Tornillos con tuerca para coples de aluminio | 26 | 2 | 52 |
| 6 | Soporte angular de 90 grados de 4 agujeros | 4 | 7.50 | 30 |
| 7 | Cople de aluminio de 6mm | 4 | 75 | 300 |
| 8 | Cople de aluminio de 8mm | 1 | 70 | 70 |
| 9 | Cople de aluminio de 3 mm | 1 | 70 | 70 |
| 10 | Soporte angular de 90 angular de 2 agujeros | 4 | 5 | 20 |
| 11 | Varilla roscada | 1 | 20 | 20 |
| 12 | Tuercas con rondana de 3/8 | 32 | 2 | 64 |
| 13 | Tornillos para fijar los motores a la estructura mecánica del brazo robótico | 9 | 1.50 | 13.50 |
| | | | Total | \$1,063.50 |

Tabla 10. Costo de materiales mecánicos

Lista de materiales eléctricos

| item | Descripción | Cantidad | P.unitario Pesos | P.total Pesos |
|------|---|----------|---------------------|------------------|
| 1 | Conectores hembra de tres pines | 4 | 5 | 20 |
| 2 | Conector macho de cinco pines | 1 | 5 | 5 |
| 3 | Conectores hembra de 4 pines | 2 | 5 | 10 |
| 4 | Servomotor con pinza | 1 | 574 | 574 |
| 5 | Motor a pasos | 1 | 70 | 70 |
| 6 | Motorreductores (Hombro y Cintura) | 2 | 641 | 1,282 |
| 7 | Motorreductor (Codo) | 1 | 450 | 450 |
| 8 | Tope mecanico | 1 | 21 | 21 |
| 9 | Regletas de conexión eléctrica de 6 mm de 12 entradas | 2 | 20 | 40 |
| 10 | Canaleta de cableado de plástico de una vía | 1 | 30 | 30 |
| 11 | Espiral de plástico para cobertura de cables | 1 | 17.28 | 17.28 |
| 12 | Fuente de PC de escritorio | 1 | 220 | 220 |
| | | | Total | \$2,739.28 |

Tabla 11. Costo de materiales eléctricos.

Lista de materiales electrónicos

| item | Descripción | Cantidad | P.Unitario Pesos | P. Total Pesos |
|------|---|----------|---------------------|----------------------|
| 1 | Sensores ópticos QTR-1A | 4 | 81 | 324 |
| 2 | Circuito impreso de etapa de potencia para motor a pasos | 1 | 122 | 122 |
| 3 | Circuito impreso de etapa de potencia para motor de corriente directa | 2 | 186 | 372 |
| 4 | DAQ 6009 | 2 | 7,940 | 15,880 |
| 5 | Tarjeta Arduino Mega | 1 | 1,249 | 1,249 |
| 6 | Tarjeta Arduino UNO | 1 | 550 | 550 |
| 7 | Encoder ENA1D-481-L00100 | 2 | 468 | 936 |
| | | | Total | \$19,433 |

Tabla 12. Costo de materiales electrónicos.

El costo total corresponde a la sumatoria de los costos mecánicos eléctricos y electrónicos los cuales están ligados a la construcción del prototipo.

| ítem | Descripción | P. Total Pesos |
|------|---------------------|-------------------|
| 1 | Costos mecánicos | \$ 1,063.50 |
| 2 | Costos eléctricos | \$ 2,739.28 |
| 3 | Costos electrónicos | \$ 19,433 |
| 4 | Total | \$ 23,235.78 |

Tabla 13. Costo total para la construcción de brazo robótico.

ANEXO 3

3.1 Circuitos impresos de etapas de potencia

Los diagramas para cada uno de los circuitos impresos se hicieron en el programa que se llama Diptrace versión 3.0.1, el primero que se empezó a diseñar fue el driver para el motor a pasos y posteriormente el driver para los motorreductores de corriente directa como se muestra en la figura 62.

En la figura 62 se ven los diagramas de las interconexiones internas, tanto de la cara de abajo (Cooper), como la cara de arriba (Top), que se implementaron en la base del robot (**módulo de control y potencia**), estos circuitos impresos se pueden replicar para reemplazar alguna de las etapas de potencia en dado que se requiera por cuestiones de mantenimiento, o para hacer modificaciones que se crean pertinentes.

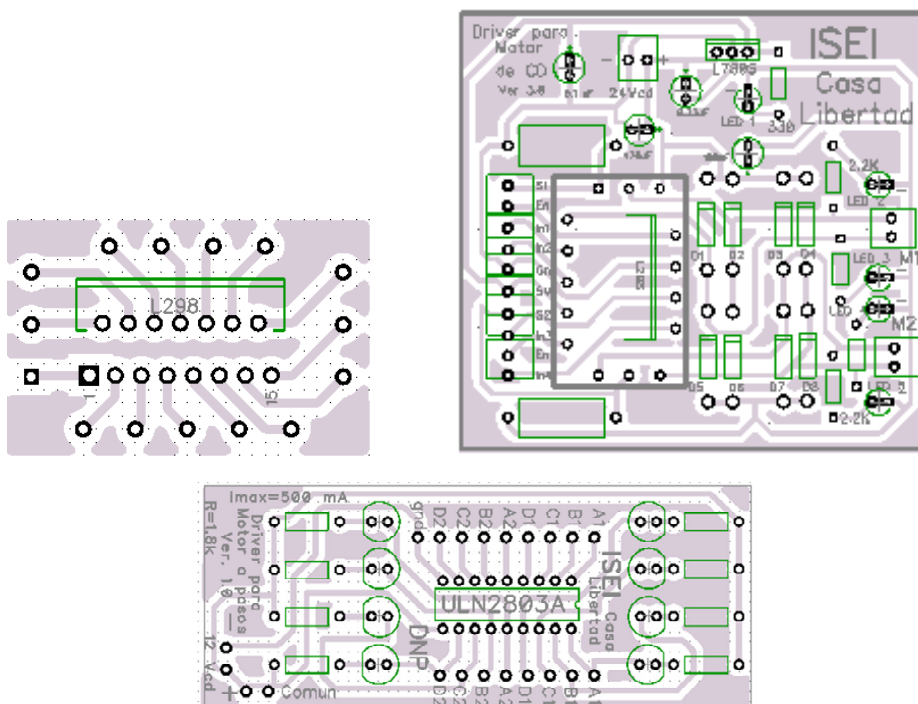


Figura 62. Circuitos impresos de etapas de potencia para motores de corriente directa.

ANEXO 4

4.1 Programa de posición inicial para el brazo robótico

En esta sección haremos referencia a las líneas de código que se utilizaron para la programación del posicionamiento inicial del robot manipulador, está dividida en dos partes:

```
// Programa de posición inicial para articulaciones de codo y muñeca de brazo robótico.
#include <Servo.h>

Servo myservo;

int potpin = A1;

int valor;

// Declaración de variables para lectura de encoder de cuadratura

int val;
int encoder0PinA = 2;
int encoder0PinB = 4;
int encoder0Pos = 0;
int encoder0PinALast = LOW;
int n = LOW;
float grados=0;
float grados1;

// Declaración de variables para motor del CODO

int valor = 0;
int motorAvance = 5;
int motorRetroceso = 6;

// Declaración de variables para sensor óptico

int sensor = 26;
int valorSensor = 0;
int estadoAnterior = 0;
int salida = 0;
```

```

// Declaración de pines para motor a pasos
int previous = 0;
int p1 = 7;
int p2 = 8;
int p3 = 9;
int p4 = 10;

// Variables para almacenar el conteo de número de pasos

int temp = 10;
int vuelta = 1024;
int cuenta = 0;

// Declaración de variables para sensor óptico y almacenamiento

int sensor1 = 24;
int valorSensor1 = 0;
int estadoAnterior1 = 0;
int salida1 = 0;

// Declaración de variables para botón virtual de posición inicial
// para articulación del codo y muñeca de Brazo Robótico

int boton = 28;
int valorSensor3 = 0;
int estadoAnterior3 = 0;
int salida3 = 0;
int contador = 0;

// Declaración de la variable para señal digital para posicionar
// las siguientes articulaciones HOMBRO y CINTURA

int boton2 = 31;

// Declaración de variables para señal digital para posicionar
// las siguientes articulaciones CODO y MUNECA

int ArduinoMega = 30;
int valorSensor4 = 0;
int estadoAnterior4 = 0;
int contador1 = 0;

```

```

// Variables para control de dirección de giro de motor de articulación
// de CODO, con botones virtuales en LabView

int botonCodolzq = 33;
int botonCodoDer = 35;
int estadoCodolzq;
int estadoCodoDer;

// Declaración de variable para activar el monitoreo de señales de los sensores
// mediante la interfaz gráfica

int senaldeparo = 37;
int estadosenaldeparo;

// Declaración de variables para los botones virtuales que
// van a mover la articulación de la MUÑECA en LabView

int buttonPin = 11;
int buttonPin1 = 12;
int estado;
int estado1;

int led = 13;

void setup() {

valorSensor3 = digitalRead(boton);
if(valorSensor3== HIGH)

{
contador == 1;
digitalWrite(led,HIGH);
delay(1000);
}

else

{
contador++;
digitalWrite(led,HIGH);
delay(1000);
}
}

```

```

// Variables de salida para la pinza

myservo.attach(3);
pinMode (encoder0PinA,INPUT);
pinMode (encoder0PinB,INPUT);
pinMode(motorAvance, OUTPUT);
pinMode(motorRetroceso,OUTPUT);
pinMode(sensor,INPUT);
pinMode(p1,OUTPUT);
pinMode(p2,OUTPUT);
pinMode(p3,OUTPUT);
pinMode(p4,OUTPUT);
pinMode(sensor1,INPUT);
pinMode(boton,INPUT);
pinMode(boton2, OUTPUT);
pinMode(ArduinoMega, INPUT);
pinMode(buttonPin, INPUT);
pinMode(buttonPin1,INPUT);
pinMode(botonCodoIzq,INPUT);
pinMode(botonCodoDer,INPUT);
pinMode(senaldeparo,INPUT);
pinMode(led, OUTPUT);

Serial.begin (9600);

}

void(*resetFunc)(void) = 0;

void loop() {

// Programación para el botón virtual proveniente de la interfaz gráfica para posicionar
// las articulaciones del codo y cintura a una referencia inicial

valorSensor3 = digitalRead(botón);
if(valorSensor3!=estadoAnterior3)

{

if(valorSensor3==HIGH)

{

contador++;
Serial.println(contador);
delay(10);

}

}
}

```

```

}

estadoAnterior3=valorSensor3;

// Programa para adquirir la señal digital que va a posicionar
// Para posicionar la articulaciones del codo y la muñeca

valorSensor4 = digitalRead(ArduinoMega); {

if(valorSensor4 == HIGH)

{
    contador1++;
    Serial.println(contador1);
    delay(10);
}

estadoAnterior4=valorSensor4;

}

switch (contador){

case 1:

digitalWrite(boton2,HIGH);
delay(1);

break;

case 2:

digitalWrite(boton2,LOW);
delay(1);

break;

case 3:

resetFunc();
delay(1000);

break;

delay(1);

}

```

```
switch (contador1) {  
  
case 1:  
  
codo();  
  
if ((grados == 75) && (valorSensor1 == LOW))  
  
    {  
        digitalWrite(valorSensor4,HIGH);  
        delay(10);  
        contador1++;  
    }  
  
break;  
  
case 2:  
  
muneca();  
  
if ((cuenta == 512) && (valorSensor1 == LOW))  
  
    {  
        contador1++;  
        Serial.println(contador1);  
        delay(10);  
  
    }  
  
break;  
  
case 3:  
  
InterfazLabview ();  
  
break;  
  
    }  
  
delay(1);  
  
}
```

```

void codo() {

    {

        analogWrite(motorAvance,LOW);
        analogWrite(motorRetroceso,150);

    }

    valorSensor = digitalRead(sensor);
    if ((valorSensor == HIGH)&&(estadoAnterior == LOW)) {
    salida = 1 - salida;
    delay(20);

    }

    estadoAnterior = valorSensor;

    if (salida == 1) {
        encoder();
        analogWrite(motorAvance,110);
        analogWrite(motorRetroceso,LOW);
        delay(20);

    }

// Posición inicial a la que va a colocarse la articulación del CODO

    if(grados == 75)

        {

            analogWrite(motorAvance,LOW);
            analogWrite(motorRetroceso,LOW);
            delay(1000);

        }

    }

```

```

void encoder()

{

n = digitalRead(encoder0PinA);
if ((encoder0PinALast == LOW) && (n == HIGH)) {
if (digitalRead(encoder0PinB) == LOW) {
encoder0Pos--;
grados = ((encoder0Pos*360/100));
grados1 = abs(grados);

if (encoder0Pos == -100){(encoder0Pos = 0); }
} else {

encoder0Pos++;
grados = ((encoder0Pos*360/100));

if (encoder0Pos == 100){(encoder0Pos = 0);}

}

Serial.print(" Articulacion del CODO #Pasos del encoder = ");
Serial.print(encoder0Pos);

Serial.print(" Posicion en grados = ");
Serial.println(grados);

}

encoder0PinALast = n;

}

void muneca() {

{

// Giro a la DERECHA para motor a pasos

// Primer paso

digitalWrite(p4,HIGH) ;
digitalWrite(p3,HIGH) ;
digitalWrite(p2,LOW) ;
digitalWrite(p1,LOW) ;
delay(temp);

```

```
// Segundo paso
```

```
digitalWrite(p4,LOW) ;  
digitalWrite(p3,HIGH) ;  
digitalWrite(p2,HIGH) ;  
digitalWrite(p1,LOW) ;  
delay(temp);
```

```
// Tercer paso
```

```
digitalWrite(p4,LOW) ;  
digitalWrite(p3,LOW) ;  
digitalWrite(p2,HIGH) ;  
digitalWrite(p1,HIGH) ;  
delay(temp);
```

```
// Cuarto paso
```

```
digitalWrite(p1,HIGH) ;  
digitalWrite(p2,LOW) ;  
digitalWrite(p3,LOW) ;  
digitalWrite(p4,HIGH) ;  
delay(temp);
```

```
}
```

```
valorSensor1 = digitalRead(sensor1);
```

```
if ((valorSensor1 == HIGH)&&(estadoAnterior1 == LOW)) {  
salida1 = 1 - salida1;  
delay(20);
```

```
}
```

```
estadoAnterior1 = valorSensor1;
```

```

if (salida1 == 1) {
    for (int i=0; i<4; i++)
    {
        horario();
    }
    if (cuenta == 512) {
// Posición inicial

digitalWrite(p1,LOW);
digitalWrite(p2,LOW);
digitalWrite(p3,LOW);
digitalWrite(p4,LOW);
delay(1000);

break;

    }
    }
}

// Giro a la izquierda motor a pasos

void horario() {

// Giro en sentido horario para posición inicial para motor a pasos
// de la articulación de la muñeca

// Primer paso

digitalWrite(p4,LOW) ;
digitalWrite(p3,LOW) ;
digitalWrite(p2,HIGH) ;
digitalWrite(p1,HIGH) ;
cuenta++;
delay(10);

```

```

// Segundo paso

digitalWrite(p4,LOW) ;
digitalWrite(p3,HIGH) ;
digitalWrite(p2,HIGH) ;
digitalWrite(p1,LOW) ;
cuenta++;
delay(10);

// Tercer paso

digitalWrite(p4,HIGH) ;
digitalWrite(p3,HIGH) ;
digitalWrite(p2,LOW) ;
digitalWrite(p1,LOW) ;
cuenta++;
delay(10);

// Cuarto paso

digitalWrite(p4,HIGH) ;
digitalWrite(p3,LOW) ;
digitalWrite(p2,LOW) ;
digitalWrite(p1,HIGH) ;
cuenta++;
delay(10);

}

void InterfazLabview ()

{

// Programa para mover servomotor

valr = analogRead(potpin);
valr = map(valr, 0, 1023, 0, 179);
myservo.write(valr);
delay(15);

// Programa para control de dirección de giro para motor
// de articulación CODO

estadoCodoIzq = digitalRead(botonCodoIzq);
estadoCodoDer = digitalRead(botonCodoDer);

```

```

// Secuencia de paro de motor del CODO

if(estadoCodolzq==LOW && estadoCodoDer==LOW)

{
  digitalWrite(motorAvance, LOW);
  digitalWrite(motorRetroceso, LOW);
}

if(estadoCodolzq==HIGH && estadoCodoDer==HIGH)

{
  digitalWrite(motorAvance, LOW);
  digitalWrite(motorRetroceso, LOW);
}

// Secuencia de giro a la Izquierda del motor del CODO
// de articulación de brazo robótico

else if (estadoCodolzq==HIGH && estadoCodoDer==LOW)

{

// Giro en sentido horario para posición inicial para motor a pasos
// de la articulación de la muñeca

// encoder();
  analogWrite(motorAvance,LOW) ;
  analogWrite(motorRetroceso,170) ;

// Serial.print ("GRADOS = ");
// Serial.println (grados1);

}

```

```

// Secuencia de giro a la Derecha del motor del CODO
// de articulación de brazo robótico

else if (estadoCodoIzq==LOW && estadoCodoDer==HIGH)

    {

//    encoder();
//    analogWrite(motorAvance,110);
//    analogWrite(motorRetroceso,LOW);

//    Serial.print ("GRADOS = ");
//    Serial.println (grados);

    }

// Programa para los dos botones virtuales para giro a la derecha
// y giro a la izquierda para motor a pasos

estado = digitalRead(buttonPin);
estado1 = digitalRead(buttonPin1);

// Secuencia de paro de motor a pasos

if(estado==LOW && estado1==LOW)

    {
        digitalWrite(p4, LOW);
        digitalWrite(p3, LOW);
        digitalWrite(p2, LOW);
        digitalWrite(p1, LOW);
    }

if(estado==HIGH && estado1==HIGH)

    {
        digitalWrite(p4, LOW);
        digitalWrite(p3, LOW);
        digitalWrite(p2, LOW);
        digitalWrite(p1, LOW);
    }

```

```
// Secuencia para giro a la derecha motor a pasos
```

```
    else if(estado== LOW && estado1== HIGH) {
```

```
// Primer paso
```

```
    digitalWrite(p4,HIGH) ;  
    digitalWrite(p3,HIGH) ;  
    digitalWrite(p2,LOW) ;  
    digitalWrite(p1,LOW) ;  
    delay(temp);
```

```
// Segundo paso
```

```
    digitalWrite(p4,LOW) ;  
    digitalWrite(p3,HIGH) ;  
    digitalWrite(p2,HIGH) ;  
    digitalWrite(p1,LOW) ;  
    delay(temp);
```

```
// Tercer paso
```

```
    digitalWrite(p4,LOW) ;  
    digitalWrite(p3,LOW) ;  
    digitalWrite(p2,HIGH) ;  
    digitalWrite(p1,HIGH) ;  
    delay(temp);
```

```
// Cuarto paso
```

```
    digitalWrite(p4,HIGH) ;  
    digitalWrite(p3,LOW) ;  
    digitalWrite(p2,LOW) ;  
    digitalWrite(p1,HIGH) ;  
    delay(temp); }
```

```
// Secuencia de giro a la Izquierda de motor a pasos
```

```
    else if (estado==HIGH && estado1==LOW)
```

```
    {
```

```
// Primer paso
```

```
    digitalWrite(p4,LOW) ;  
    digitalWrite(p3,LOW) ;  
    digitalWrite(p2,HIGH) ;  
    digitalWrite(p1,HIGH) ;  
    delay(10);
```

```
// Segundo paso
```

```
    digitalWrite(p4,LOW) ;  
    digitalWrite(p3,HIGH) ;  
    digitalWrite(p2,HIGH) ;  
    digitalWrite(p1,LOW) ;  
    delay(10);
```

```
// Tercer paso
```

```
    digitalWrite(p4,HIGH) ;  
    digitalWrite(p3,HIGH) ;  
    digitalWrite(p2,LOW) ;  
    digitalWrite(p1,LOW) ;  
    delay(10);
```

```
// Cuarto paso
```

```
    digitalWrite(p4,HIGH) ;  
    digitalWrite(p3,LOW) ;  
    digitalWrite(p2,LOW) ;  
    digitalWrite(p1,HIGH) ;  
    delay(10);
```

```
    }  
}
```

```
// Programa de posición inicial para articulaciones de hombro y cintura de brazo robótico.
```

```
// Declaración de variables para lectura de ENCODER DE CUADRATURA
```

```
int val;  
int encoder0PinA = 4;  
int encoder0PinB = 8;  
int encoder0Pos = 0;  
int encoder0PinALast = LOW;  
int n = LOW;  
float grados=0;  
float grados1
```

```
// Declaración de variables para motor del Hombro
```

```
int valor = 0;  
int motorAvance = 3;  
int motorRetroceso = 6;
```

```
// Declaración de variables para sensor óptico del HOMBRO
```

```
int sensor = 7;  
int valorSensor = 0;  
int estadoAnterior = 0;  
int salida = 0;
```

```
// Declaración de variables para motor de articulación de la Cintura
```

```
int motorAvance1 = 9;  
int motorRetroceso1 = 10;
```

```
// Declaración de variables para tope mecánico de la Cintura
```

```
int topemecanico = 11;  
int valorSensor2 = 0;  
int estadoAnterior2 = 0;  
int salida2 = 0;
```

```
// Declaración de variables de sensor óptico de la cintura
```

```
int sensor1 = 12;  
int contador = 0;  
int estadoAnterior1 = 0;  
int valorSensor1 = 0;
```

```

// Declaración de variables para botón de posición inicial
// para articulación del Hombro y Cintura de Brazo Robótico

int boton = 2;
int valorSensor3 = 0;
int estadoAnterior3 = 0;
int contador1 = 0 ;

// Declaración de variable para enviar señal digital encargada
// de posicionar las articulaciones de CODO y MUÑECA de brazo robótico

int boton2 = 5;

// Variables para control de dirección de giro de motor
// de articulación de CINTURA, con botones virtuales en LabView

int botonCinturalzq = 16;
int botonCinturaDer = 17;
int estadoCinturalzq;
int estadoCinturaDer;

// Variables para control de dirección de giro de motor
// de articulación de Hombro, con botones virtuales en LabView

int botonHombrolzq = 14;
int botonHombroDer = 15;

int estadoHombrolzq;
int estadoHombroDer;

// Declaración del pin led de la tarjeta Arduino UNO,
// Indica que se puede ejecutar la interfaz gráfica en LabView y mover
// la articulación del HOMBRO y CINTURA.

int senalParo1 = 13;

void setup() {

pinMode (encoder0PinA,INPUT);
pinMode (encoder0PinB,INPUT);
pinMode(motorAvance, OUTPUT);
pinMode(motorRetroceso,OUTPUT);
pinMode(motorAvance1, OUTPUT);
pinMode(motorRetroceso1,OUTPUT);
pinMode(sensor,INPUT);
pinMode(sensor1,INPUT);

```

```

pinMode(topemecanico,INPUT);
pinMode(boton,INPUT);
pinMode(boton2, OUTPUT);
pinMode(botonCinturalzq,INPUT);
pinMode(botonCinturaDer,INPUT);
pinMode(botonHombrolzq,INPUT);
pinMode(botonHombroDer,INPUT);
pinMode(senalParo1, INPUT);
Serial.begin (9600);

}

void(*resetFunc)(void) = 0;

void loop() {

valorSensor3 = digitalRead(boton);
if(valorSensor3!=estadoAnterior3)

{

if(valorSensor3==HIGH)

{

    contador1++;
    Serial.println(contador1);
    delay(10);
}

}

estadoAnterior3=valorSensor3;
switch (contador1){

case 1:

hombro();

if((grados1 == 18) && (valorSensor2 == LOW))

{
    digitalWrite(valorSensor3,HIGH);
    delay(10);
    contador1++;
}
}
}

```

```
break;

case 2:

cintura();

if (contador == 13 && valorSensor2 == LOW)

    {
        contador1++;
        Serial.println(contador1);
        delay(10);
    }

break;

case 3:

digitalWrite(boton2, HIGH);
delay(1);
contador1++;
Serial.println(contador1);
delay(10);

break;

case 4:

digitalWrite(boton2,LOW);
delay(1500);

InterfazLabview1 ();

break;

case 5:

resetFunc();
delay(1000);

break;

    }

}
```

```

void cintura()
{
// Giro a la izquierda de la articulación de la CINTURA del brazo robótico
{
    analogWrite(motorAvance1,LOW);
    analogWrite(motorRetroseso1,110);
}
valorSensor2 = digitalRead(topemecanico);

if ((valorSensor2 == HIGH)&&(estadoAnterior2 == LOW)){
    salida2 = 1 - salida2;
    delay(20);
}

estadoAnterior2 = valorSensor2;

// Parte del programa para mover el motor de la cintura hacia la derecha
if (salida2 == 1)
{
    Contador();
    analogWrite(motorAvance1,90);
    analogWrite(motorRetroseso1,LOW);
    delay(20);
}

if (contador == 13)
{
    analogWrite(motorAvance1,LOW);
    analogWrite(motorRetroseso1,LOW);
    delay(1000);
}
}

```

```

void hombro()
{
// Giro a la izquierda de la articulación del Hombro del brazo robótico
{
    analogWrite(motorAvance,255);
    analogWrite(motorRetroceso,LOW);
}
valorSensor = digitalRead(sensor);
if ((valorSensor == HIGH)&&(estadoAnterior == LOW))
{
    salida = 1 - salida;
    delay(20);
}
estadoAnterior = valorSensor;
// Cambio de giro de motor de articulación de hombro
if (salida == 1)
{
    encoder();
    analogWrite(motorAvance,LOW);
    analogWrite(motorRetroceso,60);
    delay(20);
}
}

```

```
// Condición para poner la articulación de la Cintura en su posición inicial
```

```
if (grados1 == 18)
```

```
{
```

```
    digitalWrite(motorAvance,LOW);  
    digitalWrite(motorRetrosceso,LOW);  
    delay(1000);
```

```
}
```

```
}
```

```
void encoder()
```

```
{
```

```
n = digitalRead(encoder0PinA);  
if ((encoder0PinALast == LOW) && (n == HIGH)) {  
if (digitalRead(encoder0PinB) == LOW) {  
    encoder0Pos--;  
grados = ((encoder0Pos*360/100));  
grados1 = abs(grados);
```

```
if (encoder0Pos == -100){(encoder0Pos = 0);}
```

```
    } else {
```

```
encoder0Pos++;  
grados = ((encoder0Pos*360/100));
```

```
if (encoder0Pos == 100){(encoder0Pos = 0);}
```

```
}
```

```
// Serial.print(" Articulacion del HOMBRO #Pasos del encoder = ");
```

```
// Serial.print(encoder0Pos);
```

```
// Serial.print(" Posicion en grados = ");
```

```
// Serial.println(grados1);
```

```
}
```

```
encoder0PinALast = n;
```

```
}
```

```

void Contador()
{
valorSensor1 = digitalRead(sensor1);
if(valorSensor1!=estadoAnterior1)
{
if(valorSensor1==HIGH)
{
contador++;
// Serial.println(contador);
// delay(100);
}
}

estadoAnterior1=valorSensor1;
}

void InterfazLabview1 ()
{
// Programa para control de dirección de giro para motor
// de Articulacion CINTURA

estadoCinturalzq = digitalRead(botonCinturalzq);
estadoCinturaDer = digitalRead(botonCinturaDer);

// Secuencia de paro de motor de la CINTURA

if(estadoCinturalzq==LOW && estadoCinturaDer==LOW)
{
digitalWrite(motorAvance1, LOW);
digitalWrite(motorRetroceso1, LOW);
}
}

```

```

if(estadoCinturalzq==HIGH && estadoCinturaDer==HIGH)
{
    digitalWrite(motorAvance1, LOW);
    digitalWrite(motorRetroceso1, LOW);
}

// Secuencia de giro a la Izquierda del motor del CINTURA
// de articulacion de brazo robotico

else if (estadoCinturalzq==HIGH && estadoCinturaDer==LOW)
{
    analogWrite(motorAvance1,LOW) ;
    analogWrite(motorRetroceso1,110) ;
}

// Secuencia de giro a la Derecha del motor del CINTURA
// de articulación de brazo robótico

else if (estadoCinturalzq==LOW && estadoCinturaDer==HIGH)
{
    analogWrite(motorAvance1,110) ;
    analogWrite(motorRetroceso1,LOW) ;
}

// Programa para control de dirección de giro para motor
// de articulación HOMBRO

estadoHombrolzq = digitalRead(botonHombrolzq);
estadoHombroDer = digitalRead(botonHombroDer);

```

```

// Secuencia de paro de motor del HOMBRO

if(estadoHombrolzq==LOW && estadoHombroDer==LOW)

{

    digitalWrite(motorAvance, LOW);
    digitalWrite(motorRetroceso, LOW);

}

if(estadoHombrolzq==HIGH && estadoHombroDer==HIGH)

{

    digitalWrite(motorAvance, LOW);
    digitalWrite(motorRetroceso, LOW);

}

// Secuencia de giro a la Izquierda del motor del HOMBRO
// de articulación de brazo robótico

else if (estadoHombrolzq==HIGH && estadoHombroDer==LOW)

{

    digitalWrite(motorAvance,HIGH) ;
    digitalWrite(motorRetroceso,LOW) ;

}

// Secuencia de giro a la Derecha del motor del HOMBRO
// de articulación de brazo robótico

else if (estadoHombrolzq==LOW && estadoHombroDer==HIGH)

{

    analogWrite(motorAvance,LOW) ;
    analogWrite(motorRetroceso,70) ;

}
}

```

ANEXO 5

Guía de usuario

Antes de encender el brazo robótico primero se debe conectar los cables USB tipo B que vienen conectados en cada una de las tarjetas DAQ USB 6009 y las tarjetas Arduino a una PC, después abrir los siguientes archivos **INTERFAZ GRAFICA DE BRAZO ROBOTICO**, realizado en LabView, **POSICION INICIAL BRAZO ROBOTICO CODO Y CINTURA** y **POSICION INICIAL BRAZO ROBOTICO HOMBRO Y CINTURA** hechos en Arduino, a continuación hay que hacer la configuración de las tarjetas Arduino correspondientes para cada programa como se va a describir a continuación:

Para el programa **POSICION INICIAL BRAZO ROBOTICO CODO Y CINTURA** en el sketch se va a configurar la tarjeta a ocupar, nos vamos a la barra de menú y le damos click en Herramientas, nos movemos con el mouse en el submenú y escogemos el modelo de la placa a utilizar en este caso va hacer Arduino Mega 2560 como se muestra en la figura 63.

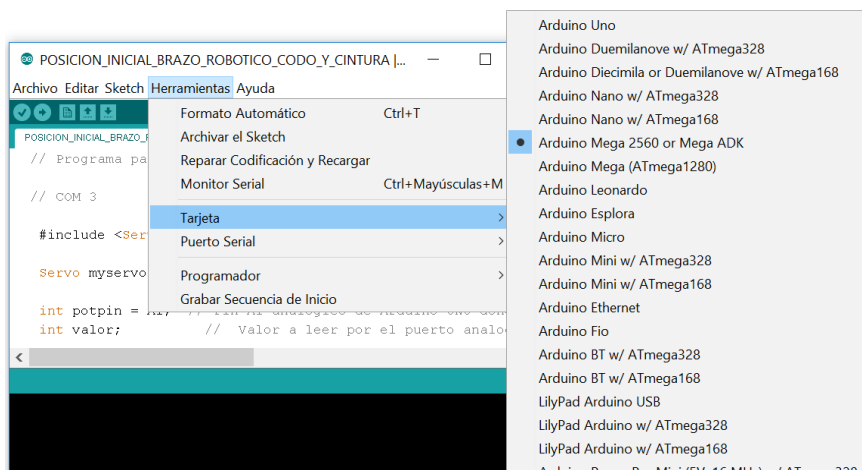


Figura 63. Configuración de tarjeta Arduino mega 2560.

Se debe configurar también el puerto serie, es la forma principal de comunicar una placa Arduino con una PC, antes de configurar el puerto serial en la tarjeta Arduino primero debemos averiguar que puerto COM detecto la computadora, para esto nos vamos a **Panel de control, Administrador de dispositivos** de la PC como se observa en la figura 64.

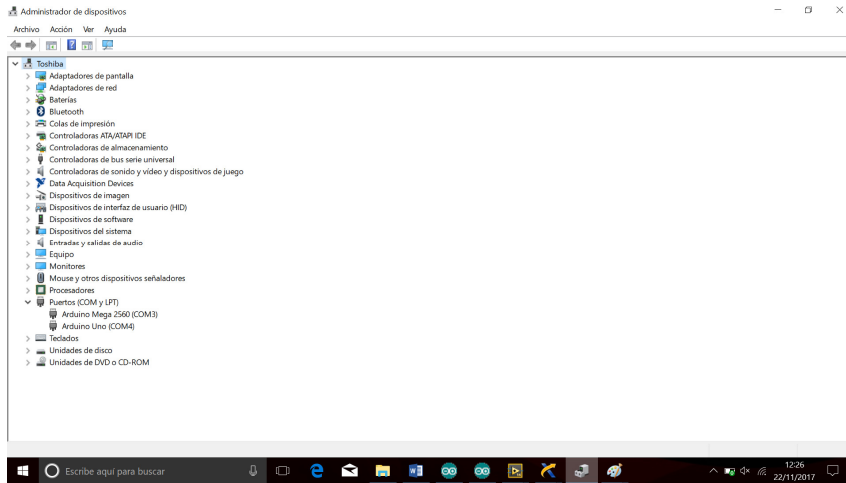


Figura 64. Puertos de comunicación para tarjetas Arduino.

En el IDE de Arduino el puerto serie se configura de la siguiente forma, en el menú Herramienta se da click y en el submenú que aparece después, dar el click seleccionamos Puerto Serial le damos en el puerto COM que detecte la computadora en este caso fue el COM3 como lo muestra la figura 65.

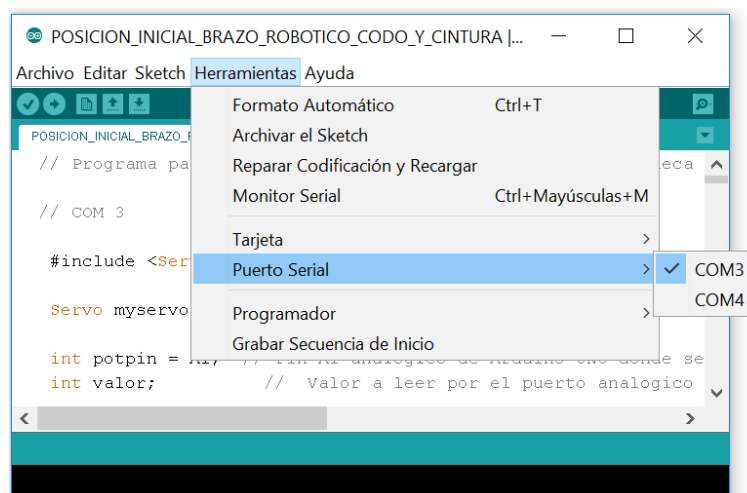


Figura 65. Configuración de puerto serial para tarjeta Arduino mega.

El programa **POSICION INICIAL BRAZO ROBOTICO HOMBRO Y CINTURA**, para la configuración del modelo de la placa a utilizar, el procedimiento de cómo hacerlo se muestra en la figura 66 y la asignación de en qué puerto COM se debe colocar para esta tarjeta depende cual es el que detecte la computadora del usuario para este caso fue COM 4 como muestra la imagen de la figura 67.

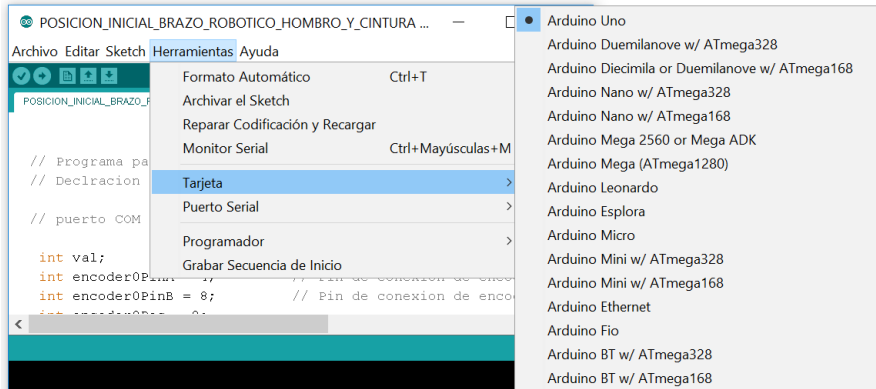


Figura 66. Configuración de tarjeta Arduino UNO.

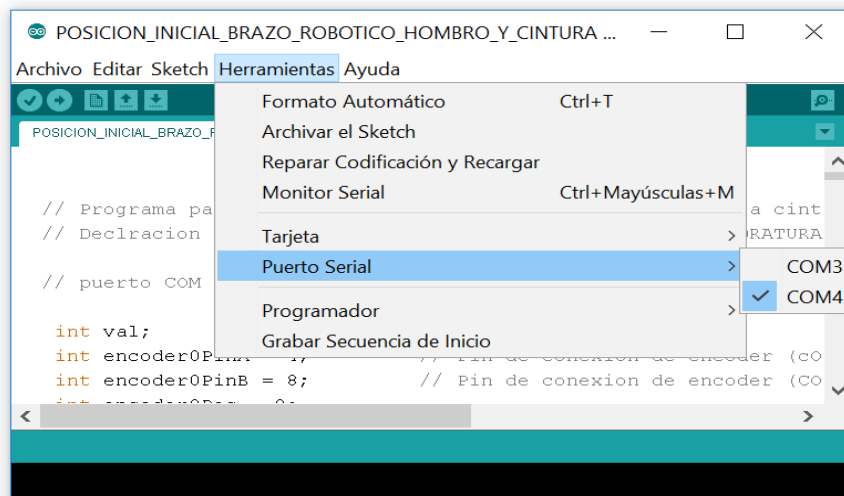


Figura 67. Configuración de puerto serial para tarjeta Arduino UNO.

Ya terminada la configuración de cada una de las tarjetas Arduino, ahora hay que abrir el archivo **INTERFAZ GRAFICA DE BRAZO ROBOTICO**, conectar las dos tarjetas DAQ USB6009 a los puertos USB de la computadora, abrir el programa NI MAX, en la ventana configuración que aparece, ir a la pestaña de **Device and Interface** y verificar primero que las tarjetas USB 6009 las reconozca el programa de LabView.

El Programa NI MAX al reconocer las tarjetas USB 6009 “0139C4BE” y “141F1D9” aparecerán identificadas con un símbolo de USB o una carpeta, dependiendo la versión utilizada, NI MAX al realizar la detección de las tarjetas DAQ 6009, si en la versión que se esté utilizando aparecen dos símbolos de una carpeta, visualizadas en color verde, si en la versión de NI MAX aparece el símbolo de USB dicho símbolo solo se visualizara activado, de lado izquierdo de los símbolos antes mencionados se visualiza NI USB-6009 “DevX”, el número de dispositivo “Dev X” lo asigna la computadora que se esté utilizando en ese momento, en nuestro caso los números que les asigno son: NI USB-6009 “Dev5”, NI USB-6009 “Dev4”. En la figura 68 de lado izquierdo se visualiza la versión 14.0 de NI MAX, y de lado derecho se visualiza la versión 17.0, la versión que se utilizó en este proyecto de tesis fue la versión 17.0.

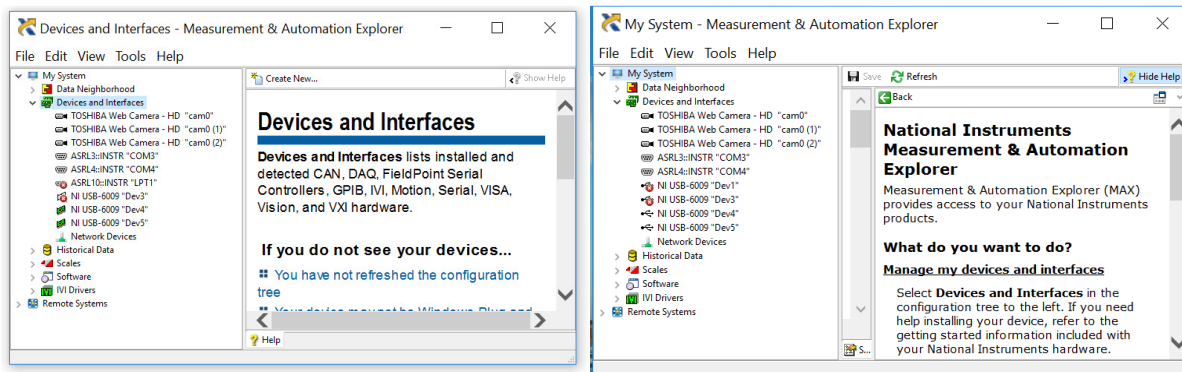



Figura 68. Reconocimiento de tarjetas DAQ 6009 en el programa NI MAX.

Si se utiliza la interfaz gráfica en otra PC, se tiene que ir al diagrama de bloques del programa de la interfaz y hacer modificaciones en cada una de las líneas de interconexión de los bloques llamados DAQ Assistant configurar el número “Dev X” de la tarjeta que reconozca la PC en ese momento, respetando las conexiones del esquemático eléctrico que viene en la parte del Anexo 1, que indica que pines deben utilizarse y como deben declararse para cada una de las dos tarjetas USB 6009, si como entradas o salidas, de tipo analógicas o digitales.

Ahora corremos el programa de la interfaz gráfica, nos situamos en el panel frontal se pulsara el botón de Run  , situado en la barra de herramientas, el programa comenzara a ejecutarse. A continuación oprimimos el botón virtual que lleva por título **Arranque** este botón está configurado como push button no es necesario mantenerlo apretado nada más hay que oprimirlo una sola vez.

Al hacer esta ejecución el panel frontal de la interfaz se encenderá un indicador luminoso en color verde, para avisar al usuario que ya está en ejecución la programación interna de la interfaz gráfica como lo muestra la figura 69.



Figura 69. Ejecución de la interfaz gráfica para mover cada una de las articulaciones de brazo robótico.


El botón llamado **Arranque** lo que hace es iniciar la ejecución la estructura de programación implementada en el panel del diagrama de bloques de la interfaz gráfica y además inicializa a las dos tarjetas Arduino para correr la programación implementada internamente en dichas tarjetas Arduino para el posicionamiento inicial del brazo robótico.

El botón llamado **Paro**, es un paro automático virtual del brazo mecánico y en el sistema físico el robot manipulador detiene lo que este ejecutando en ese momento para quedar completamente detenido.

El botón llamado **Adquisición de señales de sensores** al oprimirlo en el panel frontal de la interfaz gráfica se podrá visualizar la información de la adquisición de datos mediante los despliegues numéricos y la gráfica de las señales de los sensores colocados en la interfaz.


El botón de **Posición Inicial** al oprimirlo el robot manipulador vuelve a una referencia inicial o “home”.

Ahora comenzaremos a describir los botones virtuales encargados de controlar el movimiento de giro para los actuadores de las articulaciones cintura, hombro, codo, muñeca, y efector final.

Comencemos con el efector final, en el área del panel frontal de la interfaz gráfica, se encuentra una botonera virtual, donde se visualiza una etiqueta llamada **PINZA**, abajo de dicha etiqueta se encuentra una barra virtual , el usuario al realizar el barrido de la barra virtual se observaran dos cosas: la primera es en la imagen del brazo robótico colocada en el panel frontal de la interfaz localizada de lado izquierdo se observan dos indicadores luminosos se encenderán o apagaran conforme el usuario este haciendo el barrido indicando la apertura y cierre del efector final, y físicamente se observa la apertura y cierre de los dos dedos de la pinza.

Para mover las articulaciones del codo, hombro, y cintura, el panel frontal de la interfaz cuenta con una serie de botones virtuales mostrados en la imagen de la figura 64, la función de dichos botones es la de controlar la dirección de giro a la que deben moverse cada actuador de cada una de las articulaciones, la interfaz cuenta con un par de botones por cada articulación, donde el usuario mediante la interfaz indique que articulación se va utilizar y la dirección a la que debe moverse.

Además la interfaz gráfica cuenta con una foto del brazo robótico, en su interior de la foto se le colocaron indicadores luminosos nada más para indicar cuando se está ejecutando la programación de la posición inicial del robot, e indicar cuál de las articulaciones se está ejecutando en ese momento y en qué dirección lo hace, los indicadores se enciende conforme se está moviendo cada eslabón del brazo robótico.

Lo que a continuación sigue es abrir nuevamente los programas **POSICION INICIAL BRAZO ROBOTICO CODO Y CINTURA** y **POSICION INICIAL BRAZO ROBOTICO HOMBRO Y CINTURA** de Arduino encargados de la posición inicial para las articulaciones del robot, ir a la barra de botones de cada uno de los sketch y dar click en el icono de monitor serie  y posteriormente se abrirán en cada uno de los programas una ventana que mostrara la información que en las placas Arduino estén transmitiendo en ese momento. En la figura 70 y 71 se muestran el sketch de programación de posición inicial para las articulaciones del robot, junto con su respectiva ventana de monitor serial.

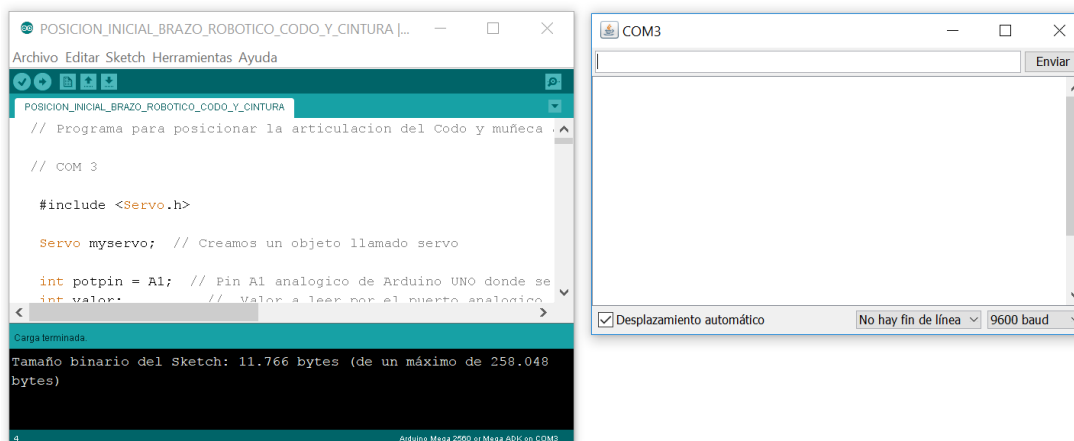


Figura 70. Posición inicial para articulación de codo y cintura de brazo robótico.

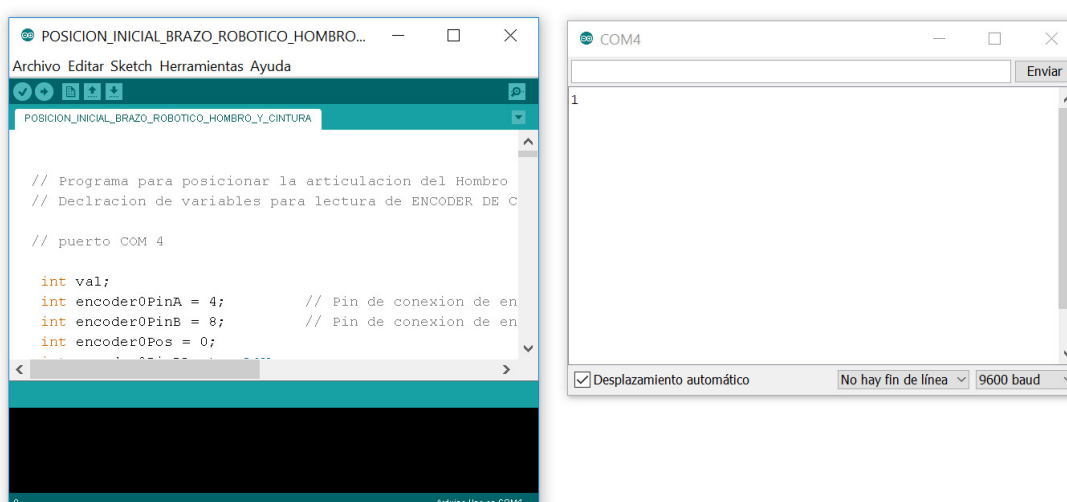


Figura 71. Posición inicial para hombro y cintura de brazo robótico.

El monitor serial del programa **POSICION INICIAL BRAZO ROBOTICO HOMBRO Y CINTURA** despliega un valor de 1, por la estructura de programación que se aplicó que tiene un comportamiento secuencial, es decir que se van a ejecutar cada uno de los subprogramas de forma secuencial para ir colocando cada articulación a una posición inicial.

Ahora ya podemos apretar el botón de encendido de la fuente de alimentación del robot, que se encuentra marcada con un círculo en color rojo, como se observa en la figura 72, al realizar esta acción se comenzara a posicionar primero la articulación del Hombro a una referencia inicial y después se irán posicionando cada una de las articulaciones restantes hasta que el robot se encuentre en posición cero o home.

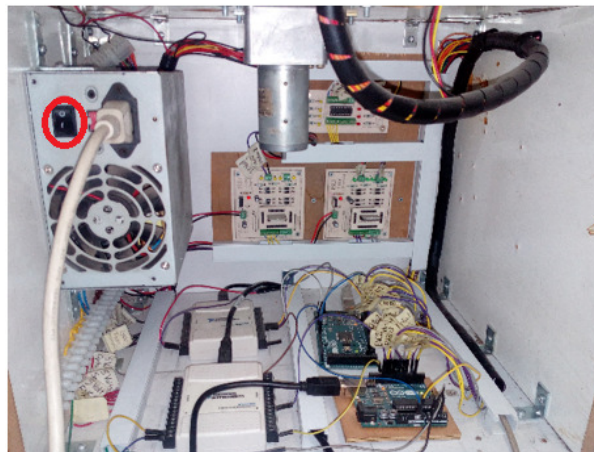


Figura 72. Botón de encendido de fuente de alimentación de brazo robótico.

Cada uno de los monitores seriales de los dos programas de Arduino van a desplegar valores numéricos secuencialmente en cada monitor serial que solo indican como se está ejecutando la programación para posicionar cada articulación a una referencia inicial en cada tarjeta Arduino. La figura 73 muestra los dos monitores seriales, el monitor serial **COM 3**, al comenzar la ejecución del posicionamiento inicial este monitor serial no va a desplegar ningún valor numérico hasta que la tarjeta Arduino UNO termine de ejecutar el posicionamiento de las articulaciones del hombro y cintura y a continuación envíe una señal digital de control a la otra tarjeta Arduino indicándole que tiene que ejecutar el posicionamiento del codo y la muñeca, entonces en el monitor serial van aparecer valores numéricos,

El monitor serial **COM 4** es el que al principio va a inicializar con un valor numérico 1 indicando con esto que se tiene que ejecutar primero la programación para posicionar la articulación del hombro del robot, y al terminar de posicionar esta articulación va a comenzar a posicionar la articulación de la cintura, al terminar de posicionar estas dos articulaciones, se va a enviar una señal digital a la tarjeta Arduino MEGA para continuar con el posicionamiento de las articulaciones del codo y la muñeca, cuando inicie la ejecución de la programación para posicionar las articulaciones antes mencionadas es cuando el monitor serial **COM 3** va a mostrar el primer valor numérico que es 1 indicando que debe de comenzar a posicionarse la articulación del codo, para posteriormente posicionar la muñeca a una referencia inicial con esto finalizando el posicionamiento total del brazo mecánico.

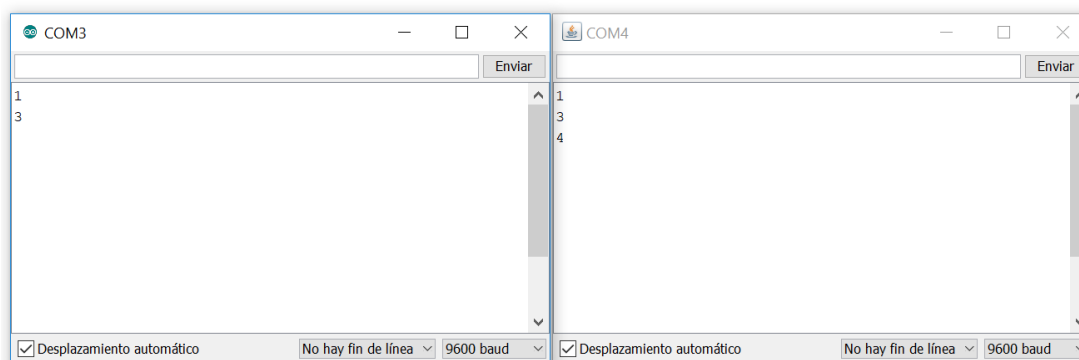


Figura 73. Monitores seriales que muestran la ejecución para posicionar cada articulación a su posición inicial.

Después de esto el usuario puede comenzar a mover cada una de las articulaciones del robot con ayuda de la interfaz gráfica.

Cuando el usuario oprima el botón virtual de **Posición Inicial** una sola vez el robot no comenzara a realizar ninguna acción, hasta que se vuelva a apretar nuevamente este botón virtual, porque en la programación en Arduino no permite que se ejecute el posicionamiento nuevamente de las articulaciones del robot al oprimir una sola vez el botón virtual, esto por las pruebas físicas con el brazo y se tiene que apretar dos veces.

En el monitor serial **COM 3** aparecerá el valor numérico 2, indica que ya se apretó una sola vez el botón de posición inicial, al apretarlo por segunda vez el valor numérico en el monitor serial se incrementa a 3 y el brazo robótico detendrá lo que se está ejecutando en ese momento y a su vez le mandara una señal digital a la tarjeta Arduino UNO para comenzar a posicionar nuevamente a una referencia inicial las articulaciones del hombro y cintura como muestra el monitor serial **COM 4** con el valor numérico 5.

Los valores 1, 3 y 4 solo indican la ejecución del posicionamiento de las articulaciones antes mencionadas, en el **COM 3** solo se van a desplegar los valores numéricos 1 y 3 que como se mencionó anteriormente solo indican como se está ejecutando la secuencia para posicionar las articulaciones del codo y muñeca.

En la figura 74 se muestran los dos monitores seriales **COM 3** y **COM 4**.

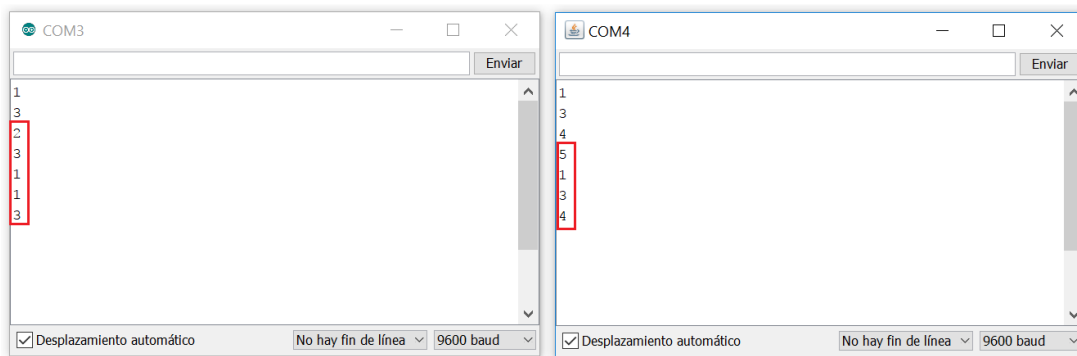


Figura 74. Posición inicial de brazo robótico al oprimir botón **Posición Inicial** en interfaz de LabView.

En la programación de las tarjetas Arduino en la parte de la función **void loop** como parte de la programación se utilizó la estructura **switch case** y se dejó un case para que el usuario agregue ahí la programación que desee para aplicaciones prácticas como ejemplo pueden ser PID para control de posición para cada una de las articulaciones del robot, como lo muestra la figura 75.

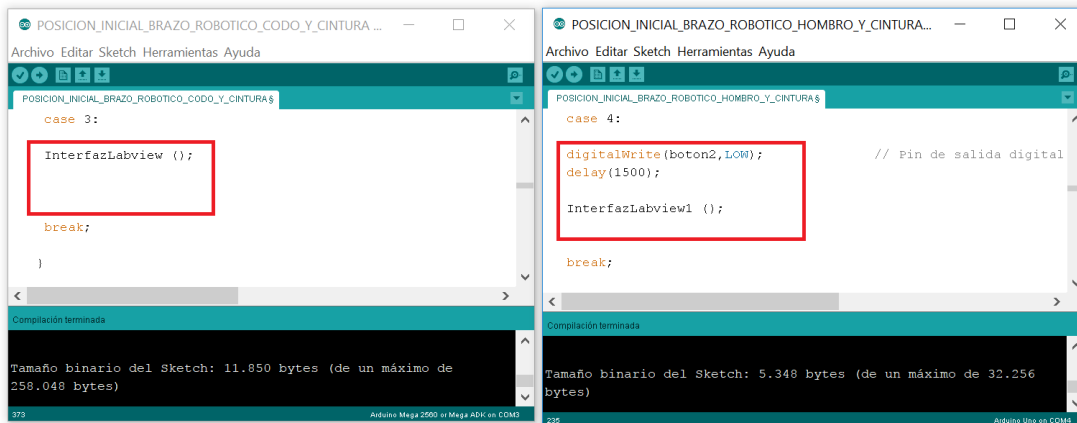


Figura 75. Espacio para programación extra para aplicaciones prácticas.

Para el apagado del sistema nos vamos al panel de control de LabView y oprimimos el botón llamado **Paro** para detener la simulación en el diagrama de bloques de LabView, después oprimir el botón de **stop** para detener la simulación de LabView.

Después oprimir el botón de apagado de la fuente de alimentación del sistema, para posteriormente desconectar primero las dos tarjetas DAQ 6009 de la PC y a continuación desconectar primero la tarjeta Arduino MEGA y luego desconectar la tarjeta Arduino UNO.