



UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL

**FACULTAD DE CIENCIAS DE LA INGENIERÍA
CARRERA DE INGENIERÍA MECATRÓNICA**

**DISEÑO Y CONSTRUCCIÓN DE UN BRAZO ROBÓTICO
ARTICULADO Y TELE-OPERADO PARA LAS OPERACIONES
DE COLOCACIÓN DE OBJETOS**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
MECATRÓNICO**

PAÚL ANDRÉS BUSTOS GUEVARA

DIRECTOR: ING. ALEXY VINUEZA, MSC

QUITO, ABRIL 2014

© Universidad Tecnológica Equinoccial. 2014
Reservados todos los derechos de reproducción

DECLARACIÓN

Yo **Paúl Andrés Bustos Guevara**, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Tecnológica Equinoccial puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

Paúl Andrés Bustos Guevara

C.I. 171626294-2

CERTIFICACIÓN

Certifico que el presente trabajo que lleva por título “**Diseño y construcción de un brazo robótico articulado y tele-operado para las operaciones de colocación de objetos**”, que, para aspirar al título de **Ingeniero Mecatrónica** fue desarrollado por **Paúl Andrés Bustos Guevara**, bajo mi dirección y supervisión, en la Facultad de Ciencias de la Ingeniería; y cumple con las condiciones requeridas por el reglamento de Trabajos de Titulación artículos 18 y 25.

Alexy Vinueza

DIRECTOR DEL TRABAJO

AGRADECIMIENTOS

A la Universidad Tecnológica Equinoccial por todos los años de aporte, formación profesional recibida y por la ética y valores impartidos en la institución.

Al Master Alexy Vinueza docente de la carrera y director de tesis, por su constante apoyo, ayuda, sugerencias y guía durante el desarrollo de este proyecto.

A mi familia por su apoyo incondicional y por ayudarme a levantar en los momentos difíciles que he cruzado.

DEDICATORIA

A Dios, por ser la luz que guía mi camino, por darme oportunidades para seguir adelante y por haberme ubicado junto a personas extraordinarias.

A mi padre Galo, por su apoyo durante toda mi vida educativa, por su esfuerzo para sacar adelante a la familia y a mí, por sus enseñanzas, consejos y sabiduría.

A mi madre Nancy, por darme el amor incondicional, la paciencia, sus enseñanzas y demostrarme como es la vida.

A mi hermano Felipe y a mi hermana Sofía, que con el amor que me han dado, han sido un eje fundamental para seguir adelante y no rendirme.

ÍNDICE DE CONTENIDOS

PÁGINA

RESUMEN	x
ABSTRACT	xi
1. INTRODUCCIÓN	1
OBJETIVO GENERAL	6
OBJETIVOS ESPECÍFICOS	6
2. MARCO TEÓRICO	8
2.1. BRAZO ROBÓTICO	8
2.1.1. Razones para utilizar robots industriales	8
2.1.2. Componentes de los robots industriales	10
2.1.3. Control de la trayectoria	12
2.1.4. Métodos de programación de robots	13
2.1.5. Selección de un robot	14
2.1.6. Ejecución de programas	16
2.1.7. Secciones de un programa	16
2.1.8. Señales electrónicas de comunicación	17
2.2. PARTE MECÁNICA.....	17
2.2.1. ESTRUCTURA	17
2.2.2. INGENIERA ESTRUCTURAL.....	18
2.2.2.1. Sujetador	18
2.2.3. ELEMENTOS ESTRUCTURALES MÁS COMUNES.....	21
2.2.3.1. Elemento tipo Cable	21
2.2.3.2. Elemento tipo Columna	22

2.2.3.3.	Elemento tipo viga	23
2.2.3.4.	Elementos tipo Arco.....	24
2.2.3.5.	Elementos tipo Cercha	24
2.2.3.6.	Elementos tipo muro.....	25
2.3.	COMPONENTES ELECTRÓNICOS	26
2.3.1.	SERVOMOTOR	26
2.3.2.	MICROCONTROLADORES	26
2.3.3.	ARDUINO	27
2.3.4.	FUENTES DE ALIMENTACIÓN	28
2.3.5.	COMUNICACIÓN INALÁMBRICA	29
2.3.5.1.	Wi-fi	29
2.3.5.2.	Cámara IP	31
2.4.	SISTEMA DE CONTROL	33
2.4.1.	BASCOM AVR	33
2.4.2.	VISUAL STUDIO.....	34
2.4.3.	VISUAL BASIC	34
3.	METODOLOGÍA.....	36
3.1.	DIAGNÓSTICO DEL SISTEMA.....	37
3.2.	REQUERIMIENTOS DEL SISTEMA	37
3.2.1.	FUNCIÓN.....	38
3.2.2.	PARÁMETROS.....	39
3.2.3.	FORMALIDAD DE LAS PRUEBAS.....	39
3.3.	RESTRICCIONES DEL SISTEMA	39
3.4.	SELECCIÓN DE ALTERNATIVAS PARA EL DISEÑO DEL BRAZO.....	41
3.4.1.	CRITERIOS DE INGENIERÍA PONDERADOS	41

3.4.2.	ALTERNATIVAS	42
3.4.3.	MÉTODO DE CRITERIOS PONDERADOS	43
4.	DISEÑO	46
4.1.	DISEÑO MECÁNICO	46
4.1.1.	ANÁLISIS DE BASTIDOR.....	46
4.1.2.	DISEÑO DEL BRAZO	48
4.1.3.	DISEÑO DEL TORNILLO	53
4.2.	DISEÑO ELECTRÓNICO	54
4.2.1.	COMPONENTES.....	54
4.2.1.1.	Cámara.....	54
4.2.1.2.	Arduino	55
4.2.1.3.	Arduino Ethernet shield	56
4.2.1.4.	Servomotor	¡Error! Marcador no definido.
4.2.1.5.	Análisis de fuente de alimentación y regulador de voltaje	58
4.3.	SISTEMA DE CONTROL	60
4.3.1.	PLATAFORMA DE INTERFAZ Y PARÁMETROS DE PROGRAMACIÓN	60
4.3.2.	PROGRAMACIÓN DE PLACA DE CONTROL	62
4.4.	CONSTRUCCIÓN	64
5.	ANÁLISIS DE RESULTADOS.....	73
5.1.	ANÁLISIS E INTERPRETACIÓN DE RESULTADOS	73
5.1.1.	MONTAJE.....	75
5.1.2.	PRUEBAS DE MOVIMIENTOS DE CADA ARTICULACIÓN	76
5.1.3.	PRUEBAS DE MOVIMIENTO DEL BRAZO COMPLETO SIN CARGA	76
5.1.4.	PRUEBAS DE MOVIMIENTO DEL BRAZO COMPLETO CON CARGA	78

5.1.5. PRUEBAS DE MOVIMIENTO DEL BRAZO COMPLETO CON CARGA MÁXIMA	80
5.1.6. DISTANCIA MÁXIMA DE CONECTIVIDAD SIN OBSTÁCULOS.....	81
5.1.7. ADAPTACIÓN AL ENTORNO GRÁFICO Y A LA MOVILIDAD DEL BRAZO ROBÓTICO.....	81
5.1.8. MOVILIDAD DEL BRAZO ROBÓTICO DENTRO DE LA INTERFAZ GRÁFICA.....	83
5.2. COMPROBACIÓN DE HIPÓTESIS.....	85
6. CONCLUSIONES Y RECOMENDACIONES.....	85
BIBLIOGRAFÍA.....	88
ANEXOS	89

ÍNDICE DE TABLAS

	PÁGINA
Tabla 1 Posibles soluciones del sistema	41
Tabla 2 Primera solución del sistema	42
Tabla 3 Segunda solución del sistema	42
Tabla 4 Tercera solución del sistema	42
Tabla 5 Cuadro de selección de alternativas del proyecto.....	43
Tabla 6 Cuadro de comparación ponderada por pares de requerimientos de ingeniería	45
Tabla 7 Cuadro de selección de alternativa por criterios ponderados	46
Tabla 8. Tiempo de carga	78
Tabla 9. Tiempo de Carga mínima.....	79
Tabla 10. Tiempo de carga máxima.....	81
Tabla 11 Resultados de sujetos de prueba.....	83

ÍNDICE DE FIGURAS

	PÁGINA
Figura 1 Brazo Robótico	2
Figura 2 Brazo robótico y sus movimientos	8
Figura 3 Robot con movimientos Cilíndricos.....	11
Figura 4 Robot con movimientos Cartesianos	11
Figura 5 Robot con movimiento Polar.....	11
Figura 6. Tipos de tornillos para destornillador	19
Figura 7. Tipos de tornillos para llave	19
Figura 8. Tornillos mariposa	20
Figura 9. Tipos de chavetas.....	20
Figura 10. Tipos de pasadores	21
Figura 11. Tipos de remaches	21
Figura 12. Cable a tensión.....	22
Figura 13 Distribución de fuerzas	22
Figura 14 Cable sujeto a fuerzas	22
Figura 15. Fuerzas de columnas.....	23
Figura 16 Elemento tipo viga con fuerzas distribuidas.....	24
Figura 17 Elemento tipo arco con carga central.....	24
Figura 18 Elemento Tipo Muro.....	25
Figura 19 Metodología Mecatrónica.....	36
Figura 20. Posibles eslabones para la construcción del brazo	40
Figura 21 Posibles articulaciones	40
Figura 22 Carga del bastidor	46
Figura 23 Base, altura y longitud de una viga.....	47
Figura 24 Pieza que conforma articulaciones	49
Figura 25 Unión en L	49
Figura 26 Pieza con conforma el bastidor.....	50
Figura 27 Pieza para colocar los servomotores	50
Figura 28 Ubicación crítica de la pieza	51
Figura 29 Fuerza para el cálculo del espesor	51

Figura 30 Cálculo en la zona de reducción del espesor	51
Figura 31 Entorno grafico de la cámara IP	55
Figura 32 Arduino Mega 2560.....	55
Figura 33 Shield Ethernet Arduino	56
Figura 34 Distribución de las fuerzas a lo largo del brazo	56
Figura 35 Diseño de la placa de regulación de voltaje en el programa Proteus	59
Figura 36 Distribución de la placa a ser construida	59
Figura 37 Interfaz Grafica	62
Figura 38 Articulación en U, base y soporte para servomotor	64
Figura 39 Unión en L, rodamiento y eslabón del eje del servomotor	65
Figura 40 Bastidor.....	65
Figura 41 Bastidor con soporte de servomotor	66
Figura 42 Colocación de servomotor	66
Figura 43 Rodamiento, tuerca y tornillo	67
Figura 44 Unión soporte de servomotor con servomotor de rotación	67
Figura 45 Conexión segundo grado de libertad	68
Figura 46 Unión de eslabones para formar la parte superior del brazo	68
Figura 47 Brazo 3 grados de libertad.....	69
Figura 48 Unión de eslabones para movimiento de la muñeca	69
Figura 49 Servomotor en eslabones de movimiento de muñeca	70
Figura 50 Brazo robot con 5 grados de libertad.....	70
Figura 51 Pinza o gripper.....	71
Figura 52 Brazo robot con 6 grados de libertad.....	71
Figura 53 Brazo robot en plataforma	72
Figura 54 Brazo robótico finalizado.....	72
Figura 55 Posición inicial	73
Figura 56 Posición Final	74
Figura 57 Posición de montaje.....	75
Figura 58. 3 Articulación	76
Figura 59 Posición inicial sin carga.....	77
Figura 60 Posición final sin carga	77

Figura 61 Objeto de prueba para carga mínima	78
Figura 62 Posición inicial con carga mínima.....	79
Figura 63 Posición final con carga mínima	79
Figura 64 Objeto de prueba para carga máxima.....	80
Figura 65 Posición inicial con carga máxima	80
Figura 66 Posición final con carga máxima	81
Figura 67 Posición inicial del usuario.....	82
Figura 68 Posición de sujeción del objeto.....	82
Figura 69 Posición final.....	83
Figura 70 Visibilidad de brazo robótico realizando movimientos	84
Figura 71 Visibilidad del brazo robótico al momento de la sujeción de un objeto	84
Figura 72 Movimiento del brazo robótico cuando está sujetando un objeto	85

ÍNDICE DE ANEXOS

	PÁGINA
Anexo 1. Programación Arduino	89
Anexo 2. Programación Visual Studio.....	100
Anexo 3. Planos y diseños del brazo robótico	114

RESUMEN

El presente proyecto, fue como objetivo el realizar el diseño y construcción de un brazo robótico articulado y tele-operado para las operaciones de colocación de objetos. El diseño de este brazo fue controlado por un microcontrolador que ejecuta las órdenes indicadas por el operador a través de un computador, además cuenta con una cámara IP que le permita al operador observar el brazo y con ello manipularlo de acuerdo a la necesidad. Fue posible diseñar con elementos que se puedan manipular remotamente. Observando que puede realizar procesos repetitivos y completar su objetivo, con movimientos precisos y rápidos en comparación al ser humano, evitando riesgos laborales. A lo largo del presente documento se presenta tanto el diseño del brazo, los cálculos necesarios para determinar las fuerzas que debe y podrá ejercer, el peso necesario que puede levantar, la longitud y diseño del brazo y los materiales, aspectos se han justificado a través de un análisis de las diferentes opciones. Se presenta también el diseño electrónico de los componentes como la cámara y el sistema robótico mismo, así como su programación, entre otros aspectos técnicos. La aplicación realizada y las pruebas sustentan la funcionalidad del brazo y muestran la utilidad del mismo que podrá ser aplicable en operaciones de gran valor para la industria, la medicina y en general de aporte al ser humano.

ABSTRACT

This project has been aimed at making the design and construction of an articulated robotic arm and tele-operated operations for object placement. The design of this arm will be controlled by a microcontroller that executes the commands given by the operator via a computer, also will have an IP camera that allows the operator to observe the arm and thereby manipulate it as needed. It was possible to design instruments that can be manipulated remotely. Noting that you can perform repetitive processes and complete their objective, accurate and rapid movement compared to humans, avoiding occupational hazards. Throughout this document both arm design , comes the necessary calculations to determine the forces that must and may exercise the necessary weight you can lift , length and arm design and materials aspects have been justified through an analysis of the different options. Electronic design of components such as the camera and the same robotic system and its programming is also presented, among other technical aspects. The application content and functionality testing support arm and show the usefulness of which may apply to transactions of value to industry, medicine and overall contribution to human.

1. INTRODUCCIÓN

El presente trabajo de investigación tratará de temas relacionados con las diferentes aplicaciones para brazos robóticos articulados y su diseño. A continuación se presentan varios conceptos que servirán de base para el entendimiento del proyecto y su desarrollo.

Aerobots

Muchos de estos sistemas responden en buena medida a los conceptos de tele manipulador o robot tele-operado, para dar una idea de las posibilidades de la robótica en estos sectores, se comentaran algunas aplicaciones concretas en las que el robot ha aportado mejoras y beneficios al proceso primitivo.

Industria Nuclear

Por sus especiales características, el sector nuclear es uno de los más susceptibles de utilizar robots de diseño específico. Entre las diversas aplicaciones se han escogido aquí, por su especial relevancia, las relativas a las operaciones de mantenimiento en zonas contaminadas y de manipulación de residuos.

Inspección de los tubos del generador de vapor en un reactor nuclear

Las operaciones de inspección y mantenimiento de las zonas más contaminadas de una central nuclear de producción de energía eléctrica son por su naturaleza largos y costosos. De realizarlas manualmente, el tiempo de exposición de los operadores a la radiación es un factor crítico que, junto con el elevado coste que supone una interrupción temporal del funcionamiento del sistema en cuestión, justifica sin lugar a dudas la utilización de sistemas robotizados, normalmente tele-operados, total o parcialmente, que sustituyan al operador.

En el generador de vapor se produce el intercambio de calor entre el fluido primario y secundario. Para ello, dentro de la vasija del generador, se encuentran dispuestos en forma matricial los tubos por los que circula el fluido receptor del calor mostrado en la figura 1.

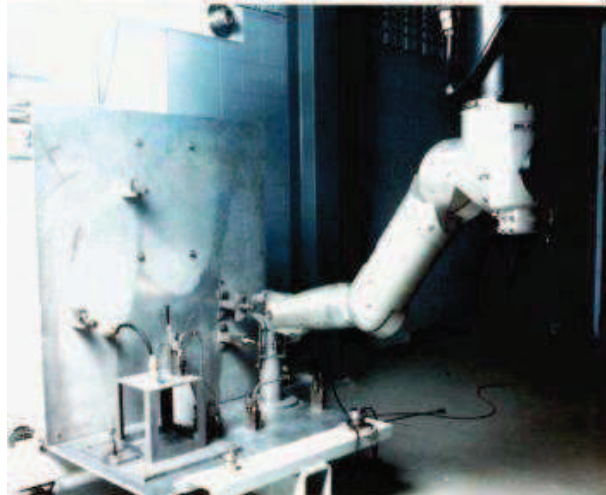


Figura 1 Brazo Robótico

Robots Nucleares. Manipulación de residuos radioactivos

La industria nuclear genera una cantidad considerable de residuos radioactivos de baja contaminación (vestimentas, envases de plástico, papel, etc.) o de alta contaminación (restos de las células del reactor, materiales en contacto directo prolongado con las zonas radioactivas, etc.). La forma, tamaño y peso de estos desechos es variable y su manipulación tiene por objeto final su envase de contenedores especiales, y posteriormente transportados y almacenados.

Para manipular remotamente estos residuos se hace uso tanto de manipuladores con unión mecánica y seguimiento directo del proceso por parte del operador a través de un cristal (en caso de baja contaminación), como con sistemas con mando remoto por radio o cable en el caso de contaminación elevada. Estos manipuladores permiten la flexibilidad necesaria para manipular elementos de peso variable y forma no definida.

Además, es preciso considerará la importancia que tiene la optimización del espacio ocupado por los residuos en su almacenamiento.

Medicina

En las aplicaciones de la robótica a la medicina destaca la cirugía. Las primeras aplicaciones de la robótica a la cirugía del cerebro datan del año 1982. En esta fecha se comenzó en Memorial Medical Center de Long Beach (California) un programa cuyo objetivo consistía en utilizar un robot comercial (Puma 260) para realizar determinadas operaciones de neurocirugía. Desde entonces se han puesto a punto varios sistemas que, con ayuda de un scanner, un ordenador registre toda la información necesaria del cerebro para que el equipo médico decida el punto exacto donde debe ser realizada la incisión, donde penetrará la sonda para obtener una muestra para realizar una biopsia. El robot, que se encuentra perfectamente situado con respecto al paciente, porta en su extremo los instrumentos necesarios para realizar la incisión, tomar la muestra, etc. La utilización de un robot conectado al ordenador permite que tanto la incisión como la toma de la muestra se realicen con la máxima precisión y en un tiempo notablemente inferior al que se consumiría en caso de emplear el sistema habitual. Además, se descarga al cirujano de la mecánica de ciertas tareas como el correcto posicionamiento de los instrumentos de cirugía con respecto al cráneo del paciente, permitiendo una mayor concentración en el seguimiento y control de la operación.

Tele-presencia

En julio de 1993 un robot SCARA en Milán (Italia) realizó sobre el hígado de un cerdo una biopsia y una incisión. El robot estaba siendo tele-operado en tiempo real desde 14000 [Km] de distancia en el Jet Propulsión Lab. De Pasadena, California. Las órdenes del cirujano y la información procedente de la sala de operaciones eran intercambiadas a través de 2 satélites de comunicaciones y redes de fibra óptica. Esta primera experiencia en

operación remota permite asegurar que este tipo de intervenciones sobre pacientes humanos podrá ser una realidad en un futuro próximo. Su justificación puede encontrarse en el tratamiento de pacientes situados en localizaciones difícilmente alcanzables (espacio, plataformas submarinas, minería, etc.). No obstante hay que considerar que aún existen importantes dificultades técnicas, como es el retraso en las comunicaciones y el elevado coste.

Construcción.

El sector de la construcción es, en la mayoría de los países industrializados, uno de los que moviliza mayor número de recursos económicos y humanos. No es de extrañar que desde hace algo más de una década se esté desarrollando gran número de sistemas robotizados, orientados a automatizar las múltiples labores que entran a formar parte de los procesos constructivos. En este tipo de aplicaciones, es Japón el país que cuenta con mayor número de sistemas en funcionamiento. En algunos casos se trata de robots parcialmente tele-operados, contruidos a partir de maquinaria convencional (grúas, excavadoras, etc.).

Si se analizan las condiciones existentes para la robotización de la construcción, se llega entre otras a las siguientes conclusiones:

- Las condiciones de trabajo son complejas.
- Los robots deben tener capacidad de locomoción y cierto grado de inteligencia.
- Deben manejar piezas pesadas y de grandes dimensiones.
- Las operaciones a realizar son complejas, variadas y poco repetitivas.
- Los robots deben ser fácilmente transportables a la obra.

Con estos condicionantes, las posibles tareas robotizables dentro de la construcción de edificios (comerciales, industriales o residenciales) podrían agruparse en:

Operaciones de colocación de elementos.

- Construcción mediante colocación repetitiva de estructuras básicas (ladrillos, Bloques, etc.).
- Posicionamiento de piezas, normalmente grandes y pesadas (vigas, etc.).
- Unión de diferentes piezas que componen una estructura (soldadura, remaches, etc.).
- Sellado de las uniones entre diferentes piezas.

Operaciones de tratamiento de superficies.

- Acabado de superficies (pulido, etc.).
- Recubrimiento de superficies con pintura, barniz, etc.
- Extensión de material sobre la superficie (cemento, espuma aislante, etc.).

Operaciones de rellenado.

- Vertido de cemento u hormigón en encofrados.
- Excavación para la preparación de terrenos y movimiento de tierras.
- Rellenado con tierra de volúmenes vacíos.

Otras.

- Inspección y control.

Fuera de la construcción de edificios, cabría destacar las realizaciones de robots para la construcción de túneles, carreteras (asfaltado), inspección de estructuras en puentes o muros de edificios difícilmente accesibles, etc.

La tabla 1, recoge alguna de las realizaciones más conocidas en esta área. Como se puede observar, destacan los desarrollos japoneses. No obstante,

debe indicarse que, como en muchas otras ocasiones, los robots japoneses son, en su mayoría, tele-operados total o parcialmente, mientras que en Europa y EU existe una mayor tendencia al desarrollo de sistemas con un mayor grado de autonomía.

OBJETIVO GENERAL

- Diseñar y construir un prototipo de un brazo robótico articulado tele-operado para las operaciones de colocación de objetos.

OBJETIVOS ESPECÍFICOS

- Realizar el análisis de requerimientos del proyecto.
- Diseñar el sistema articulado de 6 grados de libertad con herramientas CAD.
- Realizar el análisis de pesos y dimensiones de las articulaciones.
- Diseñar y construir la parte mecánica del sistema
- Diseñar y construir la parte de control
- Realizar el análisis del desempeño del brazo robótico articulado

El alcance final del presente estudio será el desarrollo del diseño y la construcción misma del brazo robótico, mismo que tendrá las siguientes características:

- Longitud de 300 – 500 [mm] cuando este estirado. Tendrá 6 grados de libertad, y tendrá como función el levantamiento de objetos pequeños tales como componentes electrónicos y con un peso máximo de 350 ± 10 [g].

- Además el brazo robótico podrá levantar piezas que tengan un máximo de 5 [cm] en alguna de sus dimensiones y que además ese objeto este ubicado en una posición que se fácil para la sujeción
- En el diseño electrónico, el brazo robótico estará controlado mediante un microcontrolador, el cual enviará las órdenes a ejecutarse por el ordenador, para los movimientos realizados por los servomotores que están acoplados a las articulaciones del brazo.
- El microcontrolador ocupará comunicación Ethernet.
- Contará con una cámara IP para que se pueda observar el brazo robótico y el entorno en el que será manipulado. Ambos la placa de control y la cámara IP estarán conectados al computador mediante un router, esto aumentará la señal de comunicación.

En el diseño mecánico, se deberá realizar cálculos de esfuerzo en los eslabones que conforma el brazo robótico y la deformación máxima que va a tener el bastidor. Además se tendrá que realizar cálculos del torque que soportaran cada uno de los servomotores y comprobar cuál será la carga máxima a ser transportada.

En la construcción se ensamblará cada una de las partes mecánicas, diseñadas y adquiridas conjuntamente con las tarjetas de control y de potencia, colocándoles en lugar donde no interfiera con el movimiento del brazo robótico.

Para verificar el correcto funcionamiento se realizarán pruebas sin carga, con carga mínima y con carga máxima, y el tiempo en que se demore cada secuencia. Además se requiere saber la distancia máxima de comunicación.

2. MARCO TEÓRICO

Esta parte de la investigación corresponde al análisis bibliográfico: tipos de materiales aptos para la construcción del brazo, robótica y sus tipos, descripción general, servomotor, microcontroladores comunicación inalámbrica y lenguajes de programación, todos estos conceptos son importantes conocer para el desarrollo del presente proyecto de titulación.

2.1. BRAZO ROBÓTICO

Es un tipo de brazo mecánico, normalmente programable, con funciones parecidas a las de un brazo humano y multifuncional, diseñado para desplazar materiales, partes, herramientas o dispositivos especiales siguiendo diferentes movimientos programados para la ejecución de una variedad de tareas y mostrado en la figura 2.

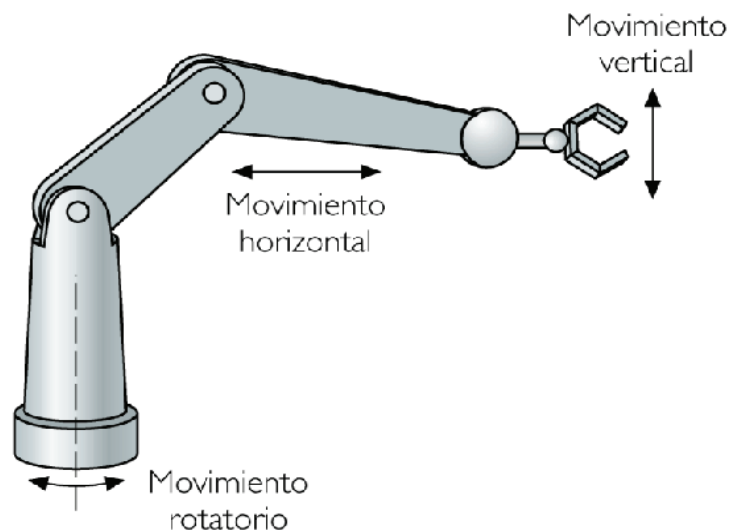


Figura 2 Brazo robótico y sus movimientos

2.1.1. Razones para utilizar robots industriales

En la manufactura moderna existen muchas ventajas en el uso de robots, en lugar de la mano de obra humana o de la automatización fija. Las ventajas potenciales de la robotización se pueden maximizar efectuando elecciones de aplicación sabias. Los robots industriales obtienen todo su potencial económico en aplicaciones donde el volumen de productos es suficientemente

grande para recuperar el desembolso en costos de equipo, programación e ingeniería, aunque lo suficientemente bajos para que no justifiquen un equipo dedicado de uso especial.

Ventajas:

- Los robots funcionan a un nivel constante de calidad.
- Se minimiza el desperdicio, las materias primas estropeadas y los re trabajos.
- Pueden trabajar en áreas que sean riesgosas o poco agradables para los seres humanos.
- Ningún trabajo es aburrido, cansado o fatigoso para los robots.
- Es posible una producción continua de 24 horas durante muchos días.
- Representa una sola inversión; no es necesario pagar salarios cada año en cantidades crecientes, y no existen costos indirectos como planes de pensiones y de seguros, vacaciones, pago durante enfermedad, etc.
- La inversión en un robot significa un desembolso de capital por una sola vez; en tanto que la mano de obra humana requiere un costo continuo por salarios, que se incrementan de manera anual.
- Los robots tienen ventajas cuando se requiere de la fuerza; y en muchas aplicaciones son más rápidos que los seres humanos
- Los robots son reprogramables con mayor facilidad para poder aceptar nuevos productos o modificaciones al diseño existente.
- A menudo los componentes del producto tienen que ser rediseñados, dando como resultado un diseño simplificado, mejor y más económico para el producto.
- Se mejorará la calidad en muchas áreas.
- Se pueden reducir el tiempo entre planeación y la terminación de la fabricación del producto.
- El trabajo en proceso puede ser menor.

2.1.2. Componentes de los robots industriales

En esencia un robot industrial está formado de dos elementos, el manipulador (o “brazo”) y el controlador del robot. El controlador contiene el sistema microprocesador y las unidades de control de energía. Los robots hidráulicos y neumáticos también tienen bombas y compresores, respectivamente. El brazo será accionado por medio de energía eléctrica, hidráulica o neumática, y se controlará con o sin servo; puede ser programado en línea, fuera de línea o en ambas formas, es capaz de moverse punto a punto, de punto a punto con trayectoria coordinada o con trayectoria continua.

Unidad de control. Governa la conexión con sensores internos y externos del robot, las unidades propulsoras, el equipo periférico, el programador y el operador. Es por lo tanto, capaz de manejar transmisiones de datos en serie y en paralelo a diversas velocidades, y puede efectuar conversión digital a analógica y analógica a digital, según se requiera. La comunicación con el programador es vía una unidad de visualización (monitor), un teclado, o una caja de enseñanza. También puede incluir unidades de disco flexibles para carga y almacenamiento de los programas.

Sistemas de propulsión y control. Muchos de los primeros robots industriales eran propulsados hidráulicamente. Sin embargo, la mayor parte de los robots producidos ahora están propulsados eléctricamente.

La definición dada de un robot implica un control relativamente complejo y esto se puede probar con el hecho de que prácticamente todos los robots de calidad industrial están totalmente controlados por servo. Los robots no controlados por servo están justo en el límite de la definición entre robots y simples unidades de tomar y colocar, por lo general son propulsados neumáticamente. Algunos dispositivos de trabajo muy ligero utilizan motores de pasos no servo.

Anatomía de los Robots. El brazo del robot está compuesto de eslabones y articulaciones. Normalmente las articulaciones, tendrán cada una de ellas sólo un grado de libertad. Para cualquier cuerpo libre en el espacio existe un máximo disponible de seis grados de libertad, como se muestra en las figuras 3, 4 y 5.

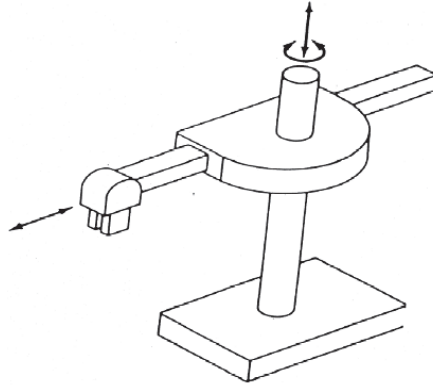


Figura 3 Robot con movimientos Cilíndricos

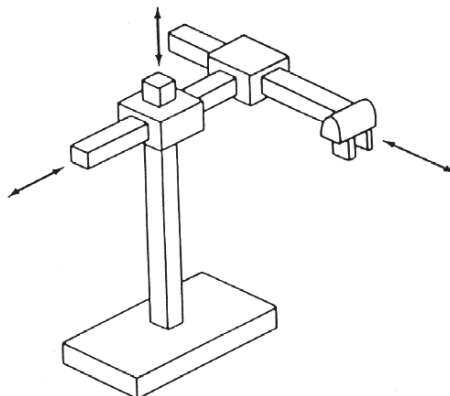


Figura 4 Robot con movimientos Cartesianos

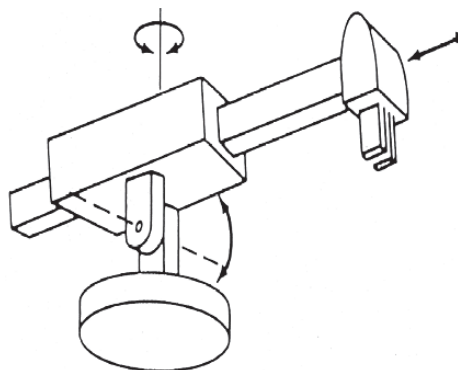


Figura 5 Robot con movimiento Polar.

2.1.3. Control de la trayectoria

En el control de punto a punto el robot se mueve entre puntos definidos, independientemente de la trayectoria seguida entre ambos. En algunos robots un software adicional permite la elección del movimiento entre puntos, ya sea en el tiempo de recorrido más corto o en línea recta. En robots revolventes, también se puede seleccionar el movimiento en modo de “codo hacia arriba” o “codo hacia abajo”. Cuando se utilizan robots con un control sencillo PTP (punto a punto) por sus siglas en Inglés, deberá tenerse especial cuidado, cuando está operando el robot, de asegurarse que se evitan colisiones con obstáculos.

En el control punto a punto con trayectoria coordinada por software, permite determinar la trayectoria que la extremidad seguirá entre los puntos. Pueden definirse líneas rectas, círculos, arcos y otras curvas. Si se trata de una línea sólo es necesario definir dos puntos; un movimiento circular se puede programar especificando tres puntos en una circunferencia, o un punto central y un radio.

El control de trayectoria totalmente continua se obtiene con mayor frecuencia al reproducir la información grabada al momento de guiar físicamente al robot a través de la tarea deseada. Cada uno de los movimientos del brazo queda grabado en tiempo real, al muestrear en alta frecuencia las posiciones de las articulaciones y esto es utilizado en programación en línea. También en algunos sistemas se emplea el control de trayectoria continua en programación fuera de línea, donde exista la posibilidad de insertar ecuaciones matemáticas para las curvas deseadas. Estas curvas posteriormente serán seguidas por la extremidad actuadora.

2.1.4. Métodos de programación de robots

Los robots industriales pueden programarse utilizando varias técnicas o métodos. Los métodos más básicos empleados en algunos de los primeros robots hidráulicos, utilizaban tambores rotatorios. Estos métodos de programación mecánica obtenían el control de la secuencia y de la distancia de recorrido. Más recientemente, para sincronizar en secuencia los movimientos de los robots neumáticos y de las unidades modulares ha sido aplicado el uso de controladores lógicos programables. Se tratan todos ellos de sistemas de control no servo, que utilizan simple retroalimentación proveniente de interruptores limitadores o de detectores de proximidad. Los robots totalmente controlados por servo emplean unidades controladoras dedicadas, basadas en microprocesador. Esto permite el empleo de técnicas complejas en y fuera de línea de programación.

Programación en línea. Aquí se utiliza el brazo del robot mismo durante la operación directa de programación. Cuando la tarea que se está programando es compleja el tiempo de programación puede resultar exageradamente largo. Básicamente existen dos métodos de programación en línea: aprendizaje mediante conducción guiada a través de la tarea y aprendizaje mediante caja de enseñanza. También es posible programar directamente desde una terminal de computadora conectada al controlador, lo que en principio es similar al método de aprendizaje por caja de enseñanza.

Programación fuera de línea. Programar fuera de línea significa la creación de un programa para una tarea robótica. Sin la necesidad de estar físicamente conectado con el robot, ni cerca de él. De hecho, cuando se combina con técnicas de simulación, la programación fuera de línea puede llevarse a cabo antes de decidir qué robot adquirir para una aplicación específica. La simulación gráfica del robot y de su entorno ofrece muchos beneficios al ingeniero industrial y al programador. Los paquetes de simulación tienen la capacidad de modelar el robot de manera cinemática y a veces de manera

dinámica, para dar una representación animada, visual y en tiempo real de cómo funcionará el robot bajo condiciones programadas. La simulación se está convirtiendo en una herramienta de programación de robots de popularidad creciente en la industria, en la educación y en la investigación.

2.1.5. Selección de un robot

La selección de un robot deberá llevarse a cabo después de en listar las demandas de la tarea como duración del ciclo, carga útil requerida, precisión requerida y costo. Estas demandas se comparan contra las especificaciones dadas por el proveedor o fabricante del robot, algunas de las cuales se muestran a continuación:

Velocidad. La velocidad máxima no necesariamente es una información muy útil, porque un brazo de robot deberá acelerarse y desacelerarse, partiendo de dicha velocidad. Para algunas aplicaciones, particularmente de ensamble, un tiempo "límite" es una especificación más útil. Este es el tiempo proporcionado por el fabricante del robot, que deberá tomar el robot para completar una serie de movimientos, llevando una carga.

Carga útil. Al haberse determinado la carga máxima que se espera manejar, se debe seleccionar un robot con una fuerza suficiente para hacer frente a una carga considerablemente mayor. La especificación deberá mostrar si la capacidad máxima de carga ocurre cuando el brazo está cerca del cuerpo o en plena extensión, donde la capacidad deberá de ser mucho menor debido a la palanca.

Precisión. Está constituida por tres elementos: resolución, capacidad de repetición y exactitud. La resolución es por lo general una característica evidente para el usuario, y por lo tanto no está incluida en las hojas de especificación estándar. Se refiere al movimiento más pequeño controlado que la extremidad hacedora es capaz de efectuar, esto se determina mediante

la resolución del controlador de la computadora (es decir el número de bits utilizados para definir la posición en un rango dado), la resolución del sistema de propulsión (es decir el número de pasos por revolución proporcionados por un motor de pasos y su engranaje asociado). La capacidad de repetición de un robot queda determinada por su resolución. La diferencia entre el punto de destino enseñado y la posición conseguida por el robot en una repetición, constituye la "exactitud". Está quedará determinada por la resolución, por las "inexactitudes del modelo en la memoria del robot y otros factores como la flexión o expansión térmica del brazo del robot.

Costo. Se considera desde el del robot básico. El robot seleccionado puede influenciar este costo total. La facilidad de programación y las capacidades de su interfaz influirán en los costos de ingeniería. También en el total deberá incluirse el costo de los accesorios, dispositivos de presentación y orientación de piezas y el herramental en la extremidad del brazo.

Sistema de control y método de programación. La aplicación tiene influencia en la elección del sistema de control de la trayectoria del robot. Los robots con control simple de punto a punto son adecuados para ensamble, paletizado y otras tareas de manejo de materiales. El control punto a punto con trayectoria coordinada es adecuado para tareas tales como soldadura de arco, aplicación de cordones selladores y soldadura por puntos de componentes en movimiento (carrocerías de automóviles en un sistema de banda transportadora) se utiliza control continuo de trayectoria donde se deba imitar el movimiento diestro del operador humano (en la pintura con pistola o donde sean necesarios movimientos de contorno complejos). Para la programación de trayectoria coordinada mediante la conducción guiada pudiera estar disponible un brazo auxiliar y para enseñanza punto a punto, o punto a punto con trayectoria coordinada, y puede utilizarse con métodos de caja de enseñanza. Para muchos robots también estará disponible programación utilizando una terminal de la computadora y un lenguaje de alto nivel.

Configuración. El proveedor proporciona información sobre la configuración y las dimensiones geométricas de la envolvente de trabajo efectiva del robot. Estas pueden ser usadas para construir plantillas ya sea en tarjetas o en computadora, que permitan diseñar una disposición apropiada del trabajo.

Información

- Software (manejo y programación)
- Lenguajes de programación
- Señales de comunicación

2.1.6. Ejecución de programas

Para poder llevar a cabo la ejecución de programas en un robot, se requieren 4 elementos:

1. Diagrama de puntos: Representación gráfica que contiene la información referente a la localización de cada punto, muestra el área en la cual el robot se desplazará de acuerdo a la programación asignada.
2. Algoritmo de programación: Secuencia de actividades que se desea realice el robot expresadas en lenguaje cotidiano.
3. Puntos: Localizaciones o posiciones físicas por las cuales el robot debe conducirse de acuerdo a la secuencia asignada en el programa.
4. Programa: Secuencia de actividades que se desea realice el robot expresadas en el lenguaje de programación correspondiente.

2.1.7. Secciones de un programa

- Inicialización: Apagar señales, enviar a su posición de home, etc.
- Preparación: Esperar pallet, colocarse en un área específica, tomar herramienta, etc.

- Operación: Ensamble, soldado, carga y descarga, etc.
- Finalización: Liberar pallet, dejar herramienta, regresar al inicio del programa, etc.

2.1.8. Señales electrónicas de comunicación

Medios con los cuales se establece comunicación entre dos estaciones de la celda de manufactura para su coordinación y control. Existen dos tipos de señales:

- **Señales de entrada:** Son aquellas que se reciben de alguna estación o dispositivo externo, proporcionan la información de la ocurrencia de un evento y condicionan la realización de la siguiente actividad del robot, no se tiene control sobre ellas.
- **Señales de salida:** Son aquellas que se envían a alguna estación o dispositivo externo de manera que provoque algún evento, se tiene control sobre ellas (Tecnológico de Monterrey, 2012).

2.2. PARTE MECÁNICA

2.2.1. ESTRUCTURA

Entidad física de carácter unitario, concebida como una organización de cuerpos dispuestos en el espacio de modo que el concepto del todo domina la relación entre las partes.

Según esta definición se observa que una estructura en un ensamblaje de elementos que mantiene su forma y su unidad.

Sus objetivos son: resistir cargas resultantes de su uso y de su peso propio y darle forma a un cuerpo, obra civil o máquina.

Ejemplos de estructuras son: puentes, torres, edificios, estadios, techos, barcos, aviones, maquinarias, presas y hasta el cuerpo humano.

Sistema estructural es un ensamblaje de miembros o elementos independientes para conformar un cuerpo único y cuyo objetivo es darle solución (cargas y forma) a un problema civil determinado.

La manera de ensamblaje y el tipo de miembro ensamblado definen el comportamiento final de la estructura y constituyen diferentes sistemas estructurales. En algunos casos los elementos no se distinguen como individuales sino que la estructura constituye en sí un sistema continuo como es el caso de domos, losas continuas o macizas y muros, y se analizan siguiendo los conceptos y principios básicos de la mecánica (Escuela de ingeniería de Antioquia, 2009).

2.2.2. INGENIERA ESTRUCTURAL

La ingeniería estructural es una rama clásica de la ingeniería civil que se ocupa del diseño y cálculo de la parte estructural en elementos y sistemas estructurales tales como edificios, puentes, muros (incluyendo muros de contención), presas, túneles y otras obras civiles. Su finalidad es la de conseguir estructuras seguras, resistentes y funcionales. En un sentido práctico, la ingeniería estructural es la aplicación de la mecánica de medios continuos para el diseño de estructuras que soporten su propio peso (cargas muertas), más las cargas ejercidas por el uso (cargas vivas), más las cargas producidas por eventos de la naturaleza, como vientos, sismos, nieve o agua. Los ingenieros estructurales se aseguran que sus diseños satisfagan un estándar para alcanzar objetivos establecidos de seguridad (por ejemplo, que la estructura no se derrumbe sin dar ningún aviso previo) o de nivel de servicio (por ejemplo, que la vibración en un edificio no moleste a sus ocupantes) (Universidad de Stuttgart, 2014).

2.2.2.1. Sujetador

Los sujetadores tienen como función mantener unidas las partes de una máquina o de un producto manufacturado. Los dispositivos de sujeción son

tan importantes en la industria, pues la igual que se los puede encontrar en la máquina más grande, un trasatlántico por ejemplo, también se puede encontrar en el reloj más pequeño.

Hay dos tipos de sujetadores de uso general: Sujetadores removibles, tales como pernos roscados, tornillos, chavetas y pasadores, y sujetadores permanentes, como los remaches.

Los tornillos

Un tornillo es un dispositivo mecánico de fijación cilíndrico o cónico, por lo general metálico, formado esencialmente por un plano inclinado enroscado alrededor.

Son dispositivos sujetadores empleados para sujetar dos piezas de material pasando a través de un agujero hecho en un miembro y enroscándolo en un agujero roscado hecho en el otro miembro. Las cabezas de tornillo se fabrican en cinco formas básicas: redonda, plana, cilíndrica, hexagonal y cuadrada, mostradas en las figuras 6 y 7.

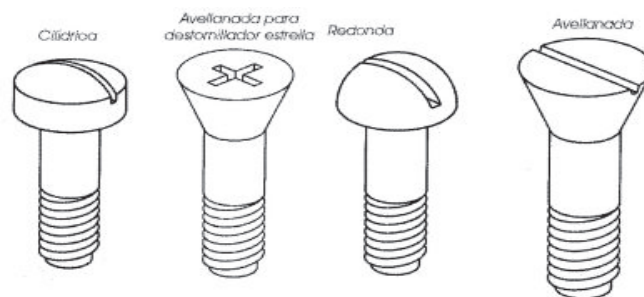


Figura 6. Tipos de tornillos para destornillador



Figura 7. Tipos de tornillos para llave

También se fabrican tornillos de presión directa o de apretar con la mano; estos reciben el nombre de tornillo mariposa o palomilla, como se muestra en la figura 8. Son muy cómodos para su manejo debido a sus dos orelas; se utilizan en artículos eléctricos, como lámparas de escritorio.

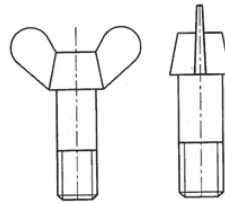


Figura 8. Tornillos mariposa

Chavetas

Las chavetas (figura 9) se utilizan como medio para sujetar poleas, volantes, manivelas, etc., a los ejes. Hay muchos tipos y tamaños de chavetas para diversas clases de trabajos. La chaveta de uso más común actualmente es la chaveta Woodruff.

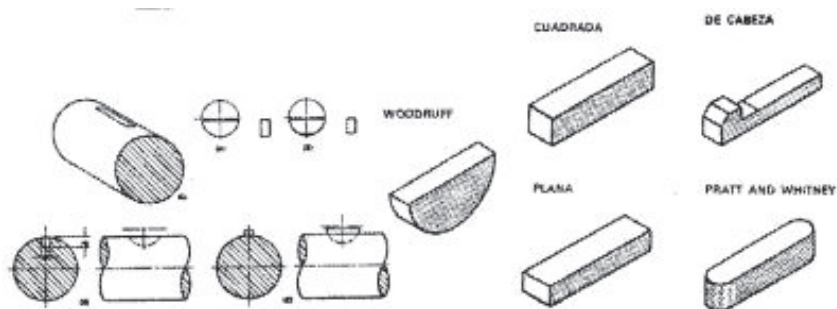


Figura 9. Tipos de chavetas

Pasadores

Son elementos de fijación mecánica desmontable, de forma cilíndrica o cónica, cuyos extremos pueden variar en función de la aplicación (figura 10).



Figura 10. Tipos de pasadores

Remaches

Los remaches (figura 11) se utilizan ampliamente en trabajo estructural, particularmente para puentes y para calderas. Sin embargo, el remachado está siendo reemplazado lentamente por la soldadura eléctrica, la cual es más barata debido a que ahorra tiempo y esfuerzo. Las cabezas de los remaches se producen en una variedad de estilos, dependiendo del tipo de trabajo a realizar.

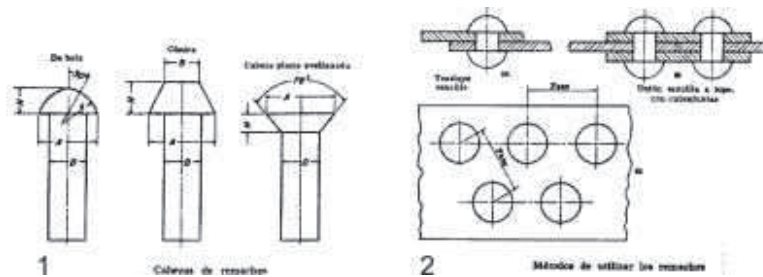


Figura 11. Tipos de remaches

2.2.3. ELEMENTOS ESTRUCTURALES MÁS COMUNES

2.2.3.1. Elemento tipo Cable

No posee rigidez para soportar esfuerzos de flexión, compresión o cortantes. Al someter a cargas a un cable este cambia su geometría de tal manera que las cargas son soportadas por esfuerzos de tracción a lo largo del elemento (presentado en la figura 12). Siempre se encuentra que cuando se aplica una fuerza el cable tendrá otra geometría.



Figura 12. Cable a tensión

Un cable bajo su propio peso adquiere la forma del diagrama de momentos de tal manera que al encontrar las fuerzas internas en cualquiera de sus puntos el valor del momento sea cero y solo presente componente de tracción (figuras 13 y 14).

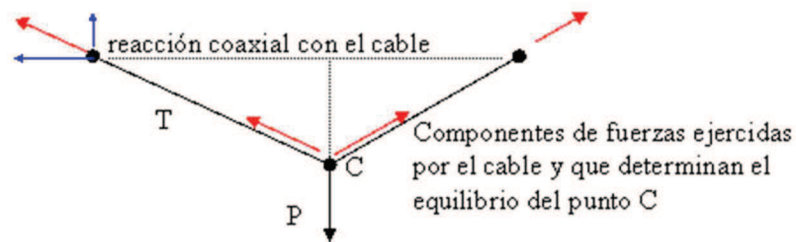


Figura 13 Distribución de fuerzas

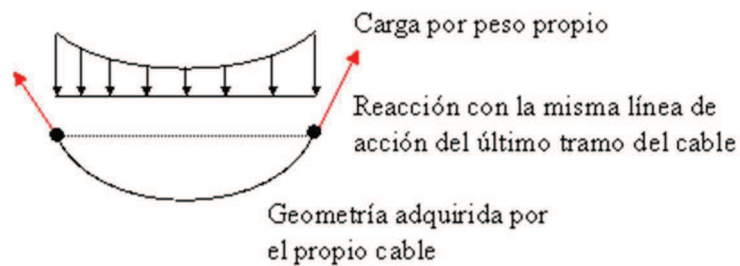


Figura 14 Cable sujeto a fuerzas

Los cables no tienen rigidez a flexión. Es un elemento con poca inercia y poca área transversal pero con una gran resistencia a la tracción.

2.2.3.2. Elemento tipo Columna

Es un elemento con dos dimensiones pequeñas comparadas con la tercera dimensión. Las cargas principales actúan paralelas al eje del elemento y por lo tanto trabaja principalmente a compresión, como se muestra en la figura

15. También puede verse sometido a esfuerzos combinados de compresión y flexión.

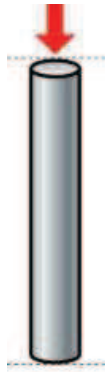


Figura 15. Fuerzas de columnas

2.2.3.3. Elemento tipo viga

Es un elemento que tiene dos de sus dimensiones mucho menores que la otra y recibe cargas en el sentido perpendicular a la dimensión mayor (figura 17). Estas características geométricas y de carga hacen que el elemento principalmente esté sometido a esfuerzos internos de flexión y de cortante. Es un elemento que debe tener la suficiente I (inercia transversal) y A (área transversal) para soportar estos tipos de esfuerzos. Se debe recordar que los esfuerzos de flexión dependen directamente de la inercia de la sección.

(ecuación de esfuerzo de flexión [2-1])

$$(\delta = \frac{M * c}{I}) \quad [2-1]$$

y los de cortante indirectamente del área (ecuación de esfuerzo cortante [2-2])

$$\tau = \frac{V * Q}{I * t} \quad [2-2]$$

Donde Q , es el primer momento del área.

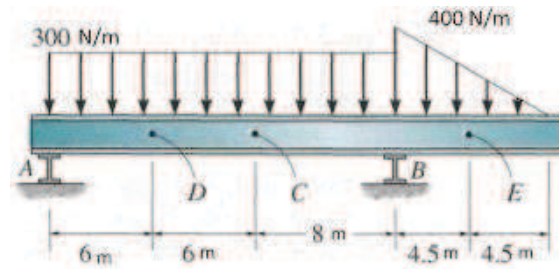


Figura 16 Elemento tipo viga con fuerzas distribuidas

2.2.3.4. Elementos tipo Arco

Se comporta o es similar a un cable invertido aunque posee rigidez y resistencia a flexión (figura 17). Esta característica lo hace conservar su forma ante cargas distribuidas y puntuales. Debido a su forma los esfuerzos de compresión son mucho más significativos que los de flexión y corte.

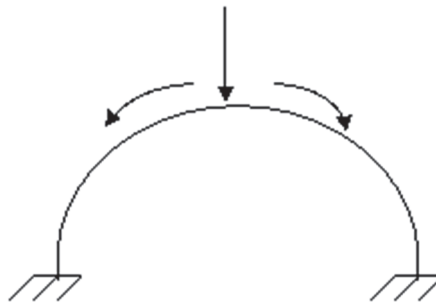


Figura 17 Elemento tipo arco con carga central

Sus esfuerzos principales son compresión y esto permite que su sección transversal sea pequeña relacionada con la luz o claro entre sus apoyos. En el caso de cargas asimétricas el esfuerzo de flexión empieza a ser notable y el arco debe tornarse más grueso.

2.2.3.5. Elementos tipo Cercha

Es un elemento cuya área transversal es pequeña comparada con su longitud y está sometido a cargas netamente axiales aplicadas en sus extremos. Por su geometría y tipo de cargas actuantes soporta solamente fuerzas de tracción y de compresión.

Su comportamiento netamente axial exige que sus conexiones a otros elementos o soportes sean rotulas sin rozamiento. Sin embargo en la práctica se construyen uniones rígidas que obligan a mantener la geometría de la sección y la posición de los nudos. Esto hace que las pequeñas deformaciones de alargamiento o acortamiento de los elementos por sus tensiones axiales, no se disipen en deformaciones de los nudos y producen entonces esfuerzos de flexión en los elementos.

Estos esfuerzos de flexión son muy pequeños comparados con sus grandes fuerzas axiales y no se tienen en cuenta en su análisis y diseño.

2.2.3.6. Elementos tipo muro

Estos elementos se caracterizan por tener dos de sus dimensiones mucho más grandes que la tercera dimensión y porque las cargas actuantes son paralelas a las dimensiones grandes (observar figura 18). Debido a estas condiciones de geometría y carga, el elemento trabaja principalmente a cortante por fuerzas en su propio plano. Adicionalmente a esta gran rigidez a corte los muros también son aptos para soportar cargas axiales siempre y cuando no se pandeen (Escuela de ingeniería de Antioquia, 2009).

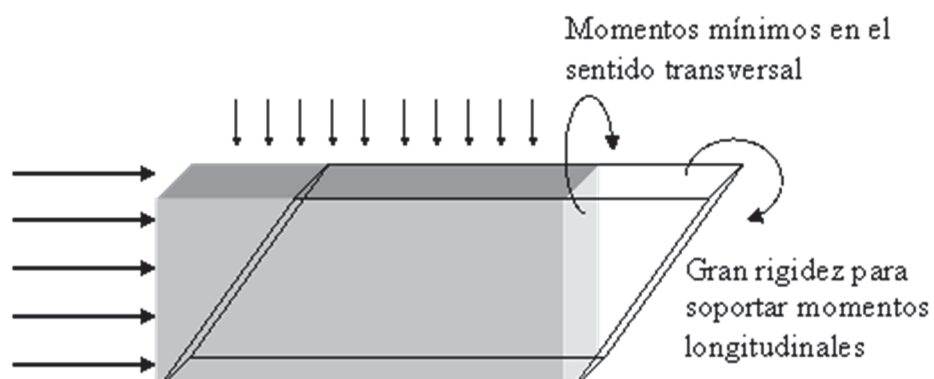


Figura 18 Elemento Tipo Muro

2.3. COMPONENTES ELECTRÓNICOS

2.3.1. SERVOMOTOR

Es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.

Es un motor eléctrico que puede ser controlado tanto en velocidad como en posición.

La corriente que requiere depende del tamaño del servo. Normalmente el fabricante indica cuál es la corriente que consume.

En otras palabras, es un motor especial al que se ha añadido un sistema de control (tarjeta electrónica), un potenciómetro y un conjunto de engranajes. Con anterioridad los servomotores no permitían que el motor girara 360 grados, solo aproximadamente 180, sin embargo, hoy en día existen servomotores en los que puede ser controlada su posición y velocidad en los 360 grados. Los servomotores son comúnmente usados en modelismo como aviones, barcos, helicópteros y trenes para controlar de manera eficaz los sistemas motores y los de dirección (Irving L. Kosow, 1993).

2.3.2. MICROCONTROLADORES

Es un circuito integrado programable y su abreviatura es μC , UC o MCU, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

Algunos microcontroladores pueden utilizar palabras de cuatro bits y funcionan a velocidad de reloj con frecuencias tan bajas como 4 kHz, con un consumo de baja potencia (mW o microvatios). Por lo general, tendrá la

capacidad para mantener la funcionalidad a la espera de un evento como pulsar un botón o de otra interrupción, el consumo de energía durante el sueño (reloj de la CPU y los periféricos de la mayoría) puede ser sólo nanovatios, lo que hace que muchos de ellos muy adecuados para aplicaciones con batería de larga duración. Otros microcontroladores pueden servir para roles de rendimiento crítico, donde sea necesario actuar más como un procesador digital de señal (DSP), con velocidades de reloj y consumo de energía más altos.

Cuando es fabricado, el microcontrolador no contiene datos en la memoria ROM. Para que pueda controlar algún proceso es necesario generar o crear y luego grabar en la EEPROM, el cual puede ser escrito en lenguaje ensamblador; sin embargo, para que el programa pueda ser grabado en la memoria del microcontrolador, debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que hace trabajar al microcontrolador cuando éste es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento (Vienna University of Technology, 2007).

2.3.3. ARDUINO

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida.⁴ Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa.

Desde octubre de 2012, Arduino se usa también con microcontroladores CortexM3 de ARM de 32 bits, que coexistirán con las más limitadas, pero

también económicas AVR de 8 bits. ARM y AVR no son plataformas compatibles a nivel binario, pero se pueden programar con el mismo IDE de Arduino y hacerse programas que compilen sin cambios en las dos plataformas. Eso sí, las microcontroladoras CortexM3 usan 3.3V, a diferencia de la mayoría de las placas con AVR que generalmente usan 5V. Sin embargo ya anteriormente se lanzaron placas Arduino con Atmel AVR a 3.3V como la Arduino Fio y existen clones de Arduino Nano y Pro como Meduino en que se puede conmutar el voltaje.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP, Pure Data) (Arduino, 2014).

Arduino también consta de shields. Las "Shields" son placas que pueden ser conectadas encima de la placa Arduino extendiendo sus capacidades. Las diferentes "shields" siguen la misma filosofía que el conjunto original: son fáciles de montar, y baratas de producir (Arduino, 2010).

2.3.4. FUENTES DE ALIMENTACIÓN

Una fuente de alimentación es un dispositivo que convierte la corriente alterna, en una o varias corrientes continuas, que alimentan los distintos circuitos del aparato electrónico al que se conecta (ordenador, televisor, impresora, router, etc.).

Las fuentes de alimentación, para dispositivos electrónicos, pueden clasificarse básicamente como fuentes de alimentación lineal y conmutada. Las lineales tienen un diseño relativamente simple, que puede llegar a ser más complejo cuanto mayor es la corriente que deben suministrar, sin embargo su regulación de tensión es poco eficiente. Una fuente conmutada, de la misma potencia que una lineal, será más pequeña y normalmente más eficiente pero será más compleja y por tanto más susceptible a averías (Edgardo Faletti, 2013).

2.3.5. COMUNICACIÓN INALÁMBRICA

2.3.5.1. Wi-fi

Wi-Fi es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Los dispositivos habilitados con Wi-Fi, tales como: un ordenador personal, una consola de videojuegos, un smartphone o un reproductor de audio digital, pueden conectarse a Internet a través de un punto de acceso de red inalámbrica. Dicho punto de acceso (o hotspot) tiene un alcance de unos 20 metros en interiores y al aire libre una distancia mayor. Pueden cubrir grandes áreas la superposición de múltiples puntos de acceso.

Wi-Fi es una marca de la Wi-Fi Alliance (anteriormente la WECA: Wireless Ethernet Compatibility Alliance), la organización comercial que adopta, prueba y certifica que los equipos cumplen los estándares 802.11 relacionados a redes inalámbricas de área local.

Aunque se tiende a creer que el término Wi-Fi es una abreviatura de Wireless Fidelity (Fidelidad inalámbrica), equivalente a Hi-Fi, High Fidelity, término frecuente en la grabación de sonido, la WECA contrató a una empresa de publicidad para que le diera un nombre a su estándar, de tal manera que fuera fácil de identificar y recordar. Phil Belanger, miembro fundador de Wi-Fi Alliance que apoyó el nombre Wi-Fi escribió:

Wi-Fi" y el "Style logo" del Yin Yang fueron inventados por la agencia Interbrand. (WiFi Alliance) contrató a Interbrand para que hiciera un logotipo y un nombre que fuera corto, tuviera mercado y fuera fácil de recordar. Se necesitaba algo que fuera algo más llamativo que "IEEE 802.11b de Secuencia Directa". Interbrand creó nombres como "Prozac", "Compaq", "OneWorld", "Imation", por mencionar algunos. Incluso inventaron un nombre para la compañía: VIATO (Phil Belanger, 2009).

Estándares que certifica Wi-Fi

Artículo principal: IEEE 802.11.

Existen diversos tipos de Wi-Fi, basado cada uno de ellos en un estándar IEEE 802.11 aprobado, los cuales son los siguientes:

Los estándares IEEE 802.11b, IEEE 802.11g e IEEE 802.11n disfrutaron de una aceptación internacional debido a que la banda de 2.4 GHz está disponible casi universalmente, con una velocidad de hasta 11 Mbit/s, 54 Mbit/s y 300 Mbit/s, respectivamente.

En la actualidad ya se maneja también el estándar IEEE 802.11a, conocido como WIFI 5, que opera en la banda de 5 GHz y que tiene una operatividad con canales relativamente limpios. La banda de 5 GHz ha sido recientemente habilitada y, además, no existen otras tecnologías (Bluetooth, microondas, ZigBee, WUSB) que la estén utilizando, por lo tanto existen muy pocas interferencias. Su alcance es algo menor que el de los estándares que trabajan a 2.4 GHz (aproximadamente un 10%), debido a que la frecuencia es mayor (a mayor frecuencia, menor alcance).

Existe un primer borrador del estándar IEEE 802.11n que trabaja a 2.4 GHz y a una velocidad de 108 Mbit/s. Sin embargo, el estándar 802.11g es capaz de alcanzar ya transferencias a 108 Mbit/s, gracias a diversas técnicas de aceleramiento. Actualmente existen ciertos dispositivos que permiten utilizar esta tecnología, denominados Pre-N.

Existen otras tecnologías inalámbricas como Bluetooth que también funcionan a una frecuencia de 2.4 GHz, por lo que puede presentar interferencias con Wi-Fi. Debido a esto, en la versión 1.2 del estándar Bluetooth por ejemplo se actualizó su especificación para que no existieran interferencias con la utilización simultánea de ambas tecnologías, además se necesita tener 40 000 k de velocidad.

Dispositivos

Existen varios dispositivos Wi-Fi, los cuales se pueden dividir en dos grupos: Dispositivos de Distribución o Red, entre los que destacan los routers, puntos de acceso y Repetidores; y Dispositivos Terminales que en general son las

tarjetas receptoras para conectar a la computadora personal, ya sean internas (tarjetas PCI) o bien USB.

Dispositivos de Distribución o Red:

- Los puntos de acceso son dispositivos que generan un "set de servicio", que podría definirse como una "Red Wi-Fi" a la que se pueden conectar otros dispositivos. Los puntos de acceso permiten, en resumen, conectar dispositivos en forma inalámbrica a una red existente. Pueden agregarse más puntos de acceso a una red para generar redes de cobertura más amplia, o conectar antenas más grandes que amplifiquen la señal.
- Los repetidores inalámbricos son equipos que se utilizan para extender la cobertura de una red inalámbrica, éstos se conectan a una red existente que tiene señal más débil y crean una señal limpia a la que se pueden conectar los equipos dentro de su alcance. Algunos de ellos funcionan también como punto de acceso.
- Los router inalámbricos son dispositivos compuestos, especialmente diseñados para redes pequeñas (hogar o pequeña oficina). Estos dispositivos incluyen, un Router (encargado de interconectar redes, por ejemplo, la red del hogar con internet), un punto de acceso (explicado más arriba) y generalmente un switch que permite conectar algunos equipos vía cable (Ethernet y USB). Su tarea es tomar la conexión a internet, y brindar a través de ella acceso a todos los equipos que conectemos, sea por cable o en forma inalámbrica (Wi-Fi Alliance, 2014).

2.3.5.2. Cámara IP

Una cámara IP (en inglés "IP cameras") es una cámara que emite las imágenes directamente a la red.

Una cámara de red incorpora su propio miniordenador, lo que le permite emitir vídeo por sí misma.

Además de comprimir el vídeo y enviarlo, puede tener una gran variedad de funciones:

- Envío de correos electrónicos con imágenes.
- Activación mediante movimiento de la imagen.
- Activación mediante movimiento de sólo una parte de la imagen.
- Creación una máscara en la imagen, para ocultar parte de ella o colocar un logo. O simplemente por adornar.
- Activación a través de otros sensores.
- Control remoto para mover la cámara y apuntar a una zona.
- Programación de una secuencia de movimientos en la propia cámara.
- Posibilidad de guardar y emitir los momentos anteriores a un evento.
- Utilización de diferente cantidad de fotogramas según la importancia de la secuencia. Para conservar ancho de banda.
- Actualización de las funciones por software.

Las cámaras IP permiten ver en tiempo real qué está pasando en un lugar, aunque esté a miles de kilómetros de distancia. Son cámaras de vídeo de gran calidad que tienen incluido un ordenador a través del que se conectan directamente a Internet.

Una cámara IP (o una cámara de red) es un dispositivo que contiene:

- Una cámara de vídeo de gran calidad, que capta las imágenes
- Un chip de compresión que prepara las imágenes para ser transmitidas por Internet
- Un ordenador que se conecta por sí mismo a Internet

Visión en vivo

Con las cámaras IP se puede ver qué está pasando en este preciso momento. La cámara se conecta a través de Internet a una dirección IP que tienen sus cámaras IP.

Las cámaras IP permiten al usuario tener la cámara en una localización y ver el vídeo en vivo desde otro lugar a través de Internet.

El acceso a estas imágenes está totalmente (en el caso que este cifrado) restringido: sólo las personas autorizadas pueden verlas. También se puede ofrecer acceso libre y abierto si el vídeo en directo se desea incorporar al web site de una compañía para que todos los internautas tengan acceso.

Microordenador

- Una cámara IP tiene incorporado un ordenador, pequeño y especializado en ejecutar aplicaciones de red. Por lo tanto, la cámara ip no necesita estar conectada a un PC para funcionar. Esta es una de sus diferencias con las denominadas cámaras web.
- Una cámara ip tiene su propia dirección IP y se conecta a la red como cualquier otro dispositivo; incorpora el software necesario de servidor de web, servidor o cliente FTP, de correo electrónico y tiene la capacidad de ejecutar pequeños programas personalizados (denominados scripts).
- También incluye entradas para alarmas y salida de relé.
- Las cámaras de red más avanzadas pueden equiparse con muchas otras funciones de valor añadido como son la detección de movimiento y la salida de vídeo analógico (Alberto LLorente, 2014).

2.4. SISTEMA DE CONTROL

2.4.1. BASCOM AVR

Bascom AVR es el compilador Basic para los AVR que sobresale entre los de su clase, aunque, la verdad, no tiene mucha competencia. Soporta los AVR de 8 bits, desde los tinyAVR hasta los XMEGA. La sintaxis de sus funciones tiene ciertas diferencias respecto de otros compiladores Basic como MBasic o Proton Plus, pero en general es fácil de asimilar.

Bascom AVR ofrece aceptables librerías, incorpora un sencillo simulador, un terminal serial y un excelente software programador de AVR que soporta casi todos dispositivos conocidos como el USB-ISP, PROGGY, FLIP, USBprog/AVR ISP mkII (AVRISP), KamProg for AVR, STK600, ARDUINO, etc.

Mark Alberts desarrolló Bascom AVR como una adaptación de su compilador Basic para los microcontroladores 8051. Actualmente lo mantiene desde su empresa MCS electronics y ofrece un versión demo que la puedes descargar desde de aquí. Es completamente funcional y solo tiene el limitante de generar códigos ejecutables de hasta 4 kbytes (Bascom Avr, 2013).

2.4.2. VISUAL STUDIO

Microsoft Visual Studio es un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta multiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, Phyton, Ruby, php; al igual que entornos de desarrollo web como ASP.NET MVC, Django, et., a lo cual sumarle las nuevas capacidades online bajo Windows Azure en forma del editor Monaco.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos, consolas (la xbox 360 y xbox one), etc. (Visual Studio, 2013).

2.4.3. VISUAL BASIC

Visual Basic es un lenguaje de programación dirigido por eventos, desarrollado por Alan Cooper para Microsoft. Este lenguaje de programación es un dialecto de BASIC, con importantes agregados. Su primera versión fue presentada en 1991, con la intención de simplificar la programación utilizando

un ambiente de desarrollo completamente gráfico que facilitará la creación de interfaces gráficas y, en cierta medida, también la programación misma.

La última versión fue la 6, liberada en 1998, para la que Microsoft extendió el soporte hasta marzo de 2008.

En 2001 Microsoft propuso abandonar el desarrollo basado en la API Win32 y pasar a un framework o marco común de librerías, independiente de la versión del sistema operativo, .NET Framework, a través de Visual Basic .NET (y otros lenguajes como C Sharp (C#) de fácil transición de código entre ellos); fue el sucesor de Visual Basic 6.

Aunque Visual Basic es de propósito general, también provee facilidades para el desarrollo de aplicaciones de bases de datos usando Data Access Objects, Remote Data Objects o ActiveX Data Objects.

Visual Basic contiene un entorno de desarrollo integrado o IDE que integra editor de textos para edición del código fuente, un depurador, un compilador (y enlazador) y un editor de interfaces gráficas o GUI (Visual Basic Express Edition, 2012).

3. METODOLOGÍA

La ingeniería mecatrónica integra sistemáticamente y en forma simultánea el diseño mecánico, electrónico y el de control. La metodología de la mecatrónica considera el análisis, el diseño y la implementación de la máquina.

A continuación, en la figura 19 se presenta a manera de esquema las partes del diseño e implementación de sistemas mecatrónicos:

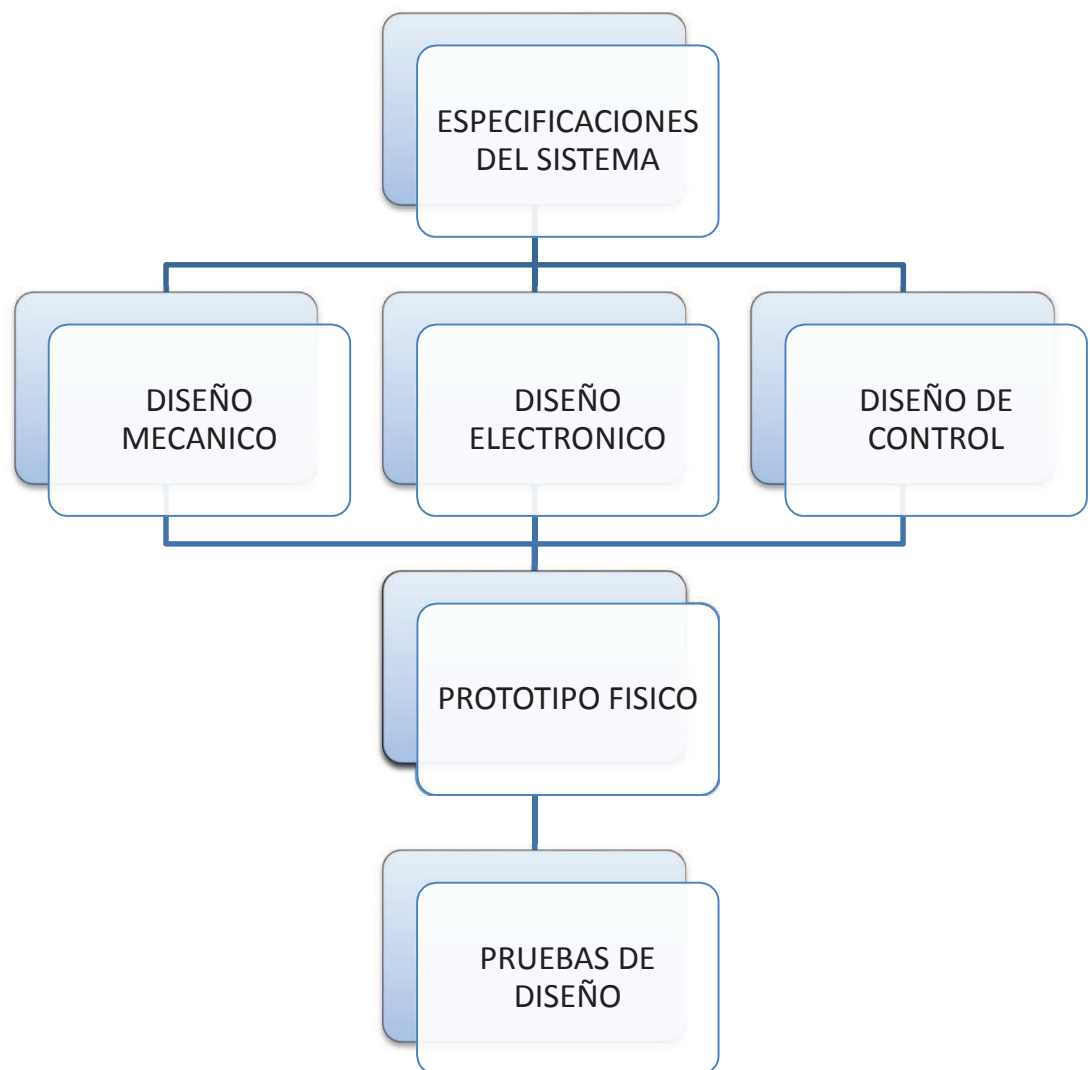


Figura 19 Metodología Mecatrónica

La presente investigación inicia con el estudio de torque de cada uno de los servomotores aplicados al brazo, para la estructura y sus cargas,

adicionalmente se ha realizado el estudio de fuerzas máximas que van a soportar, el brazo y los servomotores.

Para el diseño electrónico se realizará un análisis de los dispositivos aptos para la comunicación entre la computadora y el brazo robot. El diseño de control requiere realizar un análisis de los software aptos que permitan un entorno gráfico claro y a su vez sea posible la programación que gobernará los movimientos y posicionamiento del brazo

3.1. DIAGNÓSTICO DEL SISTEMA

Uno de los objetivos de la realización del presente proyecto ha sido desarrollar una herramienta que pueda llegar a lugares donde el ser humano no puede llegar, es por ello que se ha pensado la instalación de la cámara y el brazo para que sea posible observar el entorno en donde se está trabajando y el mismo se transmita a la pantalla.

Otra funcionalidad que tendrá el equipo es que, al momento de monitorear el brazo robot, si este no está en buen funcionamiento, el operario puede parar el sistema, acomodarlo y volver a ponerlo en funcionamiento, esta operación se la puede realizar remotamente.

3.2. REQUERIMIENTOS DEL SISTEMA

El equipo estará controlado remotamente, la persona va a observar en la posición en la que se encuentra el brazo y los objetos alrededor de él. Este al tener 6 grados de libertad va a poder mover con facilidad al objeto, cada grado de libertad corresponde a cada articulación movida por cada servomotor, este brazo consta con una pinza que va a ser encargada de la sujeción de los objetos.

Para el proceso de desarrollo de los conceptos de diseño se detallan los siguientes requerimientos técnicos de ingeniería:

Fácil Uso: Que permita su manipulación de forma sencilla al operador

Facilidad de montaje y fabricación: Que no exista dificultad al momento de ensamblar y fabricar las piezas.

Peso: Que el peso de toda la estructura esté acorde a las necesidades de uso

Facilidad de mantenimiento: Que permita su eventual reparación o actualización de la maquina

Funcionalidad: Que el uso permita el cumplimiento de los objetivos para los que ha sido diseñado y que cumpla en precisión y velocidad.

Complejidad de control: Que sea factible su programación y manipulación

Costo: Que su valor de fabricación se optimice a través de su uso.

Requerimientos de esfuerzos por parte del operador: Que no requiera de un esfuerzo mayor por parte del operador al momento de manipular la maquina

Carga máxima: Que pueda soportar la carga que va a usualmente mover, con la tolerancia calculada.

3.2.1. FUNCIÓN

El usuario manipulará el brazo mediante un entorno gráfico que presenta el software en la pantalla, este enviará los datos que el usuario ingreso mediante la conexión Wi-Fi establecida hacia la tarjeta de control. La programación en la tarjeta de control codificará los datos recibidos y posicionará el brazo con los datos recibidos.

La placa de control constará de un shield Ethernet, esto servirá para la conexión hacia el router que va a transmitir los datos hacia el computador que estará manipulando el usuario. La cámara IP también estará conectada al router para tener un mayor alcance.

3.2.2. PARÁMETROS

Los movimientos que va a realizar el brazo serán controlados mediante un programa de computadora, en el programa se podrá ver el brazo y se podrá realizar los movimientos del brazo.

3.2.3. FORMALIDAD DE LAS PRUEBAS

Las pruebas de cargas del brazo robótico se lo realizo bajo techo, la pruebas del Wi-Fi se las realizó al medio ambiente sin obstáculos para así poder medir cuál es el máximo alcance de comunicación.

Para un mejor uso es recomendable la práctica bajo techo, ya que si recibe directamente el sol a las placas o la caja donde están todas las placas de control, se puede sobrecalentar y dañar a los circuitos, los mismo ocurriría con la lluvia, en cuyo caso si se moja, podría provocarse un corto circuito tanto a las placas como a los servomotores.

3.3. RESTRICCIONES DEL SISTEMA

Cada sección del brazo robot tendrá orificios, esto es para que ciertas piezas sean las mismas y puedan ser ocupadas en cualquier posición, además los orificios pequeños son para evitar la sujeción de otros elementos que se requieran.

Como este brazo robot es un prototipo, se lo va a construir con materiales que no sean muy costosos y que sean fáciles de manufacturar, en este caso el aluminio. La tolerancia que será ocupada en los orificios debe ser ± 0.2 [mm] para que pueda entrar con facilidad los tornillos de unión.

Para la construcción del brazo robot, las formas de las conexiones de los miembros pueden ser de diferentes maneras, una es la conexión en “U” como se puede observar en la figura 20.



Figura 20. Posibles eslabones para la construcción del brazo

La ventaja de ocupar este tipo de piezas es que al momento de unirlos se forman dos eslabones que están conectados en el medio de cada pieza, como se puede observar en la figura 21.



Figura 21. Posibles articulaciones

Va a haber 2 tipos de servomotor, lo cual va a disminuir la carga máxima, ya que el servo hace mover la primera sección del brazo y soportará el peso del brazo más el peso del objeto que será desplazado. Se estima que la carga

máxima será de 0.2 [Kg]. El peso de esos servomotores será de 0.055 [Kg] y de 0.078 [Kg] respectivamente.

3.4. SELECCIÓN DE ALTERNATIVAS PARA EL DISEÑO DEL BRAZO

Para la selección de alternativas para el diseño del brazo, se consideran los aspectos técnicos, material, movimiento, costos, peso, criterio de ingeniería, definición una línea por cada criterio.

3.4.1. CRITERIOS DE INGENIERÍA PONDERADOS

En la tabla 1 se colocan las diferentes opciones a través de las cuales se puede resolver cada una de las características que tiene el proyecto.

Tabla 1 Posibles soluciones del sistema

Tipos de soluciones		Opción 1	Opción 2	Opción 3
Dispositivo de Control	de Arduino 2560	Mega	PLC s7-300	Micro controlador PIC 18F4550
Material de la estructura	de la	Aluminio	Acero	Madera
Servomotores	Turnigy MG959r 30 [Kg.cm]		TowerPro MG996r 10 [Kg.cm]	TowerPro MG946r 12 [Kg.cm]
Fuente de alimentación	de	700 W	300 W	150W
Conexión Wi-Fi		1 antena	2 antenas	3 antenas

3.4.2. ALTERNATIVAS

Una vez que ya se dispone de las opciones para cada característica se va a desarrollar 3 tipos de soluciones (tablas 2, 3, y 4) que se pueden presentar:

Tabla 2 Primera solución del sistema

Tipos de soluciones	Opción 1
Dispositivo de Control	Arduino Mega 2560
Material de la estructura	Aluminio
Servomotores	Turnigy MG959r 30 [Kg.cm]
Fuente de alimentación	700 W
Conexión Wi-Fi	2 antenas

Tabla 3 Segunda solución del sistema

Tipos de soluciones	Opción 2
Dispositivo de Control	PLC s7-300
Material de la estructura	Acero
Servomotores	TowerPro MG946r 12 [Kg.cm]
Fuente de alimentación	300 W
Conexión Wi-Fi	3 antenas

Tabla 4 Tercera solución del sistema

Tipos de soluciones	Opción 3
Dispositivo de Control	Micro controlador 18F4550
Material de la estructura	Madera
Servomotores	TowerPro MG996r 10 [Kg.cm]

Fuente de alimentación	150 W
Conexión Wi-Fi	1 antenas

3.4.3. MÉTODO DE CRITERIOS PONDERADOS

Para definir los criterios se utiliza la tabla No. 5, donde se califica con 5 como máximo y 1 como mínimo. La calificación de los criterios dados se basó en los conocimientos técnicos y de experiencia del creador de la tesis.

Tabla 5 Cuadro de selección de alternativas del proyecto

<u>CUADRO DE SELECCIÓN DE ALTERNATIVAS DEL PROYECTO</u>	FACIL USO	FACILIDAD DE MONTAJE Y FABRIM	PESO	FACILIDAD DE MANTENIMIENTO	FUNCIONALIDAD	COMPLEJIDAD DE CONTROL	REQUERIMIENTO DE ESFUERZOS	BAJO COSTO	CARGA MAXIMA	TOTAL
	ALTERNATIVA 1	3	4	4	3	4	4	3	4	5
ALTERNATIVA 2	3	3	2	3	2	3	3	3	3	25
ALTERNATIVA 3	3	3	5	3	3	2	4	5	4	32

En la tabla 6 se califica un criterio con respecto a otro, si hay similitudes entre cada criterio se coloca 1, si no hay mucha similitud se coloca 0.5 y si no hay similitud se coloca 0. Una vez realizado este proceso se obtiene una ponderación para cada criterio, mediante el cual se procede a redistribuir el puntaje en cada criterio.

En la tabla 7 se observa la multiplicación de los valores de la tabla 5 con su respectiva ponderación de acuerdo a cada criterio, esto ayuda a equilibrar la importancia en cada uno.

Al realizar los criterios ponderados se puede observar que la alternativa 1 es la que tiene el mayor puntaje y por tanto la seleccionada, ya que se va a optimizar el uso de materiales para la construcción de este dispositivo, además se observa que los criterios más importantes han sido: el fácil uso y

la funcionalidad de la máquina. Esta funcionalidad en la alternativa seleccionada es mejor porque su carga máxima es mayor que las otras.

Tabla 6 Cuadro de comparación ponderada por pares de requerimientos de ingeniería

CRITERIOS	Fácil uso	Facilidad de Montaje y Fabricación	Peso	Facilidad de Mantenimiento	Funcionalidad	Complejidad de control	Requerimientos de esfuerzos por parte del operador	Costo	Carga máxima	$\Sigma + 1$	Ponderación
Fácil uso	1	0.5	1	1	1	1	0.5	1	0.5	7.5	0.17
Facilidad de Montaje y Fabricación	1	0	0.5	0.5	0.5	0.5	1	1	0	4.5	0.10
Peso	0	0	0.5	0.5	1	0	0.5	1	1	5	0.11
Facilidad de Mantenimiento	0	1	0	0	1	0	0.5	1	0.5	5	0.11
Funcionalidad	0.5	0	1	0	0.5	0.5	1	1	1	5.5	0.12
Complejidad de control	0.5	0.5	0	0	1	0.5	1	0.5	0	4	0.09
Requerimientos de esfuerzos por parte del operador	0	1	0	0	1	1	0	0	0.5	4.5	0.10
Costo	0	0	1	0	0	1	0.5	0	0.5	4	0.09
Carga máxima	0.5	0	1	0.5	1	0	0.5	0.5	4.5	4.5	0.10
								SUMA	44.5	1.00	

Tabla 7 Cuadro de selección de alternativa por criterios ponderados

Criterios	Fácil uso	Facilidad de montaje y fabricación	Peso	Facilidad de mantenimiento	Funcionalidad	Complejidad de control	Requerimientos de esfuerzos por parte del operador	Costo	Carga Maxima	Σ	Prioridad
Alternativa 1	0.51	0.41	0.45	0.34	0.50	0.36	0.31	0.45	0.45	3.34	1
Alternativa 2	0.51	0.31	0.23	0.34	0.25	0.27	0.31	0.27	0.27	2.49	3
Alternativa 3	0.51	0.31	0.57	0.34	0.38	0.18	0.41	0.45	0.36	3.15	2

4. DISEÑO

4.1. DISEÑO MECÁNICO

Para elaborar el diseño mecánico del brazo robot se requiere crear un esquema computarizado del modelo, para ello se ha seleccionado el diseño en Solid Works que permite modelar piezas en 3D y realizar el ensamble de las piezas y mecanismos para poder observar la construcción completa del brazo.

4.1.1. ANÁLISIS DE BASTIDOR

El bastidor estará anclado en la base metálica por su seguridad, éste soportará todo el peso del brazo y el peso de los objetos que van a ser manipulados.

La fuerza que soportará será de 7.349 [N], pero por seguridad se aumentará en un 50%, proporcionando así una fuerza máxima de 11.024 [N].

Se necesita observar cuál es la deformación máxima del bastidor, para ello se utilizarán los siguientes datos:

Longitud del soporte: 135 [mm]

Peso máximo: 11.024 [N]

Tipo de material: Aluminio 1050-0 con un módulo de elasticidad de 70 [GPa] (Delmetal, 2012)

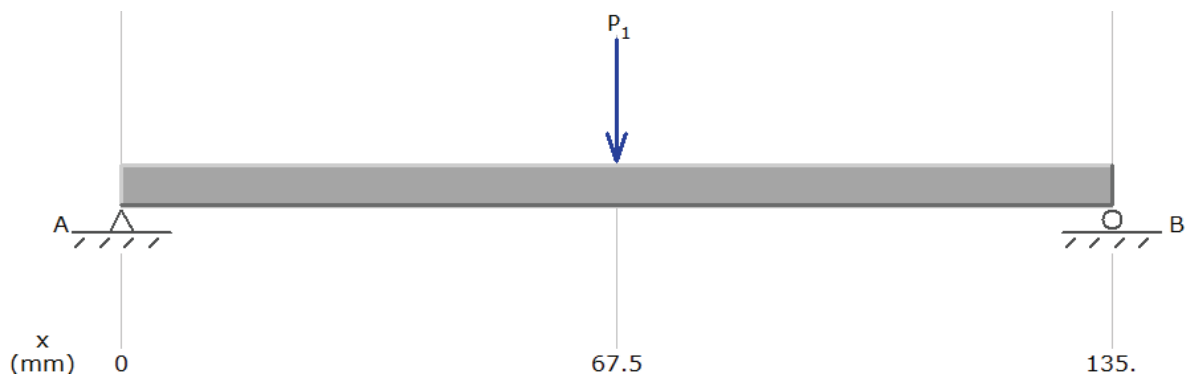


Figura 22 Carga del bastidor

La fórmula para determinar la deformación máxima es la siguiente (ecuación 4.1):

$$y_{max} = -\frac{P * L^3}{3 * E * I} \quad [4-1]$$

Donde,

P= fuerza máxima

L= longitud del soporte

E= módulo de elasticidad

I= inercia del soporte

Para calcular la inercia se requiere de la ecuación [4-2]:

$$I = \frac{B * H^3}{12} \quad [4-2]$$

Donde:

B= es la medida de la base (como se muestra en la figura 23)

H= es la medida de la altura (como se muestra en la figura 23)

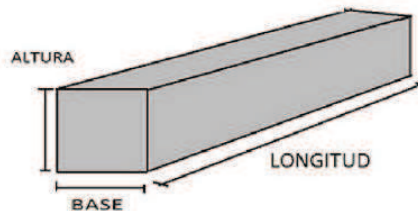


Figura 23 Base, altura y longitud de una viga

Por tanto los cálculos serán:

$$I = \frac{90.2 [mm] * (29 [mm])^3}{12}$$
$$I = 183323.983 [mm^4] = 183.323 * 10^{-9} [m^4]$$

Una vez obtenida la inercia se procede a calcular la flexión máxima

$$y_{max} = - \frac{11.024[N] * (0.135 [m])^3}{3 * 70 * 10^9 [\frac{N}{m^2}] * 183.323 * 10^{-9} [m^4]}$$

$$y_{max} = -7.045 * 10^{-7} [m] \cong 7.045 * 10^{-4} [mm]$$

4.1.2. DISEÑO DEL BRAZO

En el diseño del brazo robot se optó por escoger como material el aluminio, debido al costo y al peso que va a tener la estructura en relación a los aceros. No se optó por polímeros ya que el movimiento de los servos motores puede romper las piezas que conforman el brazo.

Se va a ocupar el aluminio 1050-0, cuyas propiedades son:

Módulo de elasticidad aproximado:	70 GPa
Resistencia máxima a la ruptura:	90 MPa
Temperatura de fusión:	646-657 °C

(Delmetal, 2012)

Basándose en modelos con articulaciones, se realizará el estudio de las piezas y tensiones que deben soportar. El brazo consta de diferentes piezas, de las cuales se va a indicar el modelado de dichas piezas. En la figura 24 se muestra uno de los eslabones que se requiere para la construcción del brazo.

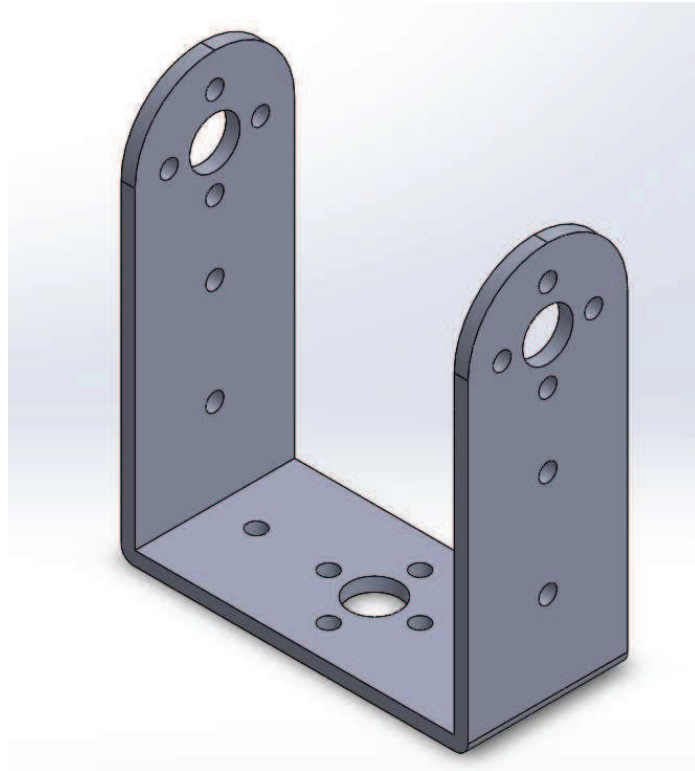


Figura 24 Pieza que conforma articulaciones

En la figura 25 se muestra una pieza que sirve para la unión de la parte superior del brazo.

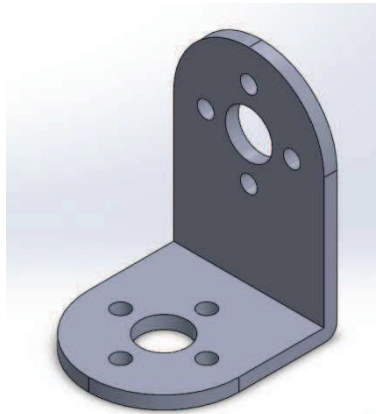


Figura 25 Unión en L

La figura 26 muestra una de las piezas de las que está conformada el bastidor del brazo.

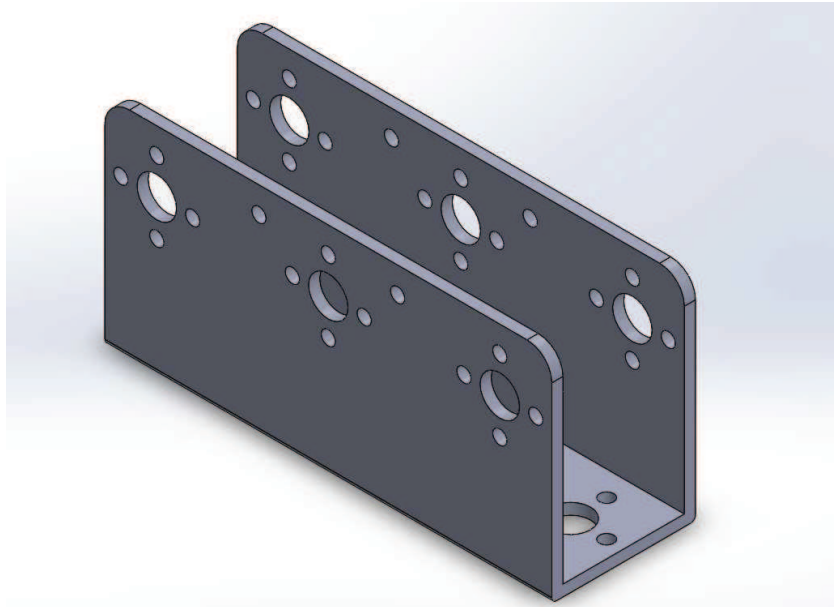


Figura 26 Pieza con conforma el bastidor

La figura 27 corresponde a la pieza que sirve como articulación, donde los servomotores estarán colocados y atornillados.

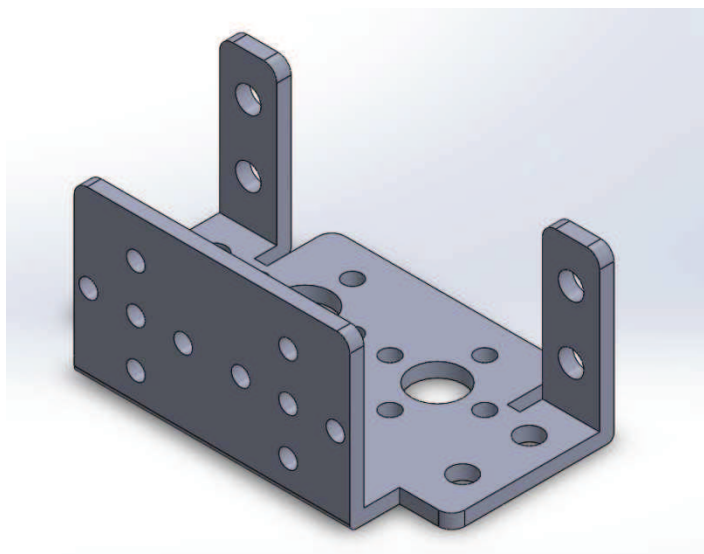


Figura 27 Pieza para colocar los servomotores

Para realizar el cálculo de las piezas, se necesita determinar el espesor mínimo que se necesita para el peso requerido. La estructura pesa 0.55 ± 0.005 [Kg], y el peso máximo que va a levantar será de 200g, el máximo peso del sistema será de 0.750 ± 0.005 [Kg], lo cual en fuerza será 7.349 [N].

El cálculo de esta pieza será la que tenga la mayor carga dentro de todo el sistema, para ello se hará 2 cálculos.

El cálculo 1, la fuerza máxima será colocado al extremo de la barra, esta barra va a estar cantiliber sujeta con un servomotor al otro extremo como se muestra en las figuras 28 y 29.

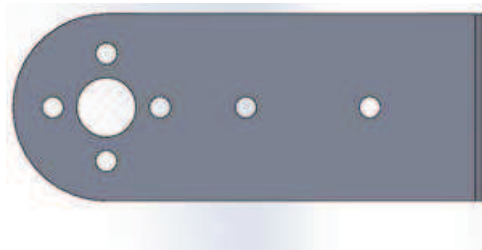


Figura 28 Ubicación crítica de la pieza



Figura 29 Fuerza para el cálculo del espesor

El cálculo 2, la fuerza será colocada en el cambio de sección mayor como se muestra la figura 30.

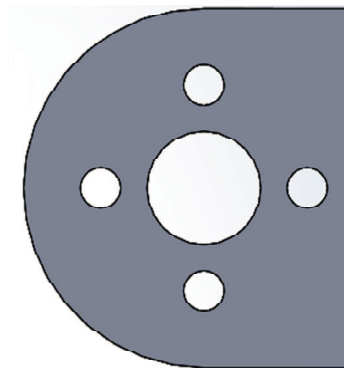


Figura 30 Cálculo en la zona de reducción del espesor

Para hacer el primer cálculo del espesor que necesita la estructura, se requiere de la ecuación [4-3] (ecuación de esfuerzo máximo):

$$\delta = \frac{F}{A} \quad [4-3]$$

Donde:

δ = Esfuerzo máximo del material

F= Fuerza que va a ser aplicada

A= Área que soportará la fuerza

El esfuerzo máximo de este material es de $90 \left[\frac{N}{mm^2} \right]$

La fuerza que va a soportar es 7.349 [N], a la cual se le aumentará un 30% más por seguridad, la fuerza total será de 9.554 [N].

El área será la multiplicación del espesor y del ancho de la pieza. El ancho es de 25.18 mm.

Reemplazando datos se tiene lo siguiente:

$$90 \frac{N}{mm^2} = \frac{9.554 N}{\text{espesor} * 25.18 mm}$$

$$\text{espesor} = 0.00421 mm$$

Para hacer el segundo cálculo se ocupa la fórmula [4-3], la misma fuerza y esfuerzo, el ancho de la pieza se reduce ya que ahí se encuentran realizados los orificios por donde van a pasar los tornillos.

Por lo tanto, la ecuación reemplazada los valores quedaría de la siguiente manera:

$$90 \frac{N}{mm^2} = \frac{9.554 N}{\text{espesor} * (25.18 - 2 * .85 - 7.95) mm}$$

$$\text{espesor} = 0.0092 mm$$

Al realizar los cálculos, se puede observar que ambos espesores son demasiado pequeños como para encontrar en el mercado actual, por ello se optará por el espesor similar que ha sido factible encontrar, que es de 2.4 mm.

Bastidor: carga solo brazo, carga máxima + 20%

Estructura: estático, dinámico

Articulaciones

4.1.3. DISEÑO DEL TORNILLO

Para armar el brazo se necesita unir las diferentes piezas que compone el brazo, para ello se necesita tornillos según el tipo de estructura que se va a utilizar.

Para esto se debe encontrar el diámetro que requiere la unión de las piezas, para eso el tornillo siempre va a estar sometido a un esfuerzo cortante, por lo que se procede a calcular el diámetro del tornillo con la formula [4-4].

$$S_{sut} = \frac{F}{A} \quad [4-4]$$

Dónde:

F= es la fuerza de corte aplicada al tornillo

A= es el área del tornillo ($\pi * \frac{D^2}{4}$)

Ssut= es la tercera parte del 100% del esfuerzo máximo del material del tornillo

F=7.349 [N]

Basándose en el libro de diseño de elementos de máquina de Mott, se ocupará un perno de grado 4.8, ya que se puede encontrar el tamaño del perno necesario que es M2.5x0.4

Su= 400 [MPa]

Ssut= 0.75*Su

Ssut= 300 [MPa]

(Delmetal, 2012)

Con estos datos se calcula:

$$S_{sut} = \frac{F}{\pi * \frac{D^2}{4}}$$

$$D = \sqrt{\frac{4 * F}{\pi * S_{sut}}}$$

$$D = 1.76 * 10^{-4}$$

4.2. DISEÑO ELECTRÓNICO

4.2.1. COMPONENTES

4.2.1.1. Cámara

La cámara IP consta del protocolo TCP/IP y requiere un puerto de entrada http. La dirección IP configurada es 192.168.0.249 y puerto http es el 81. Para ingresar al entorno de la cámara se debe colocar <http://192.168.0.249:81>

Como se puede observar en la figura 31 se muestra el entorno de la cámara, esta consta de 2 motores a paso, el motor de la base gira 270 grados y el motor de elevación lo hace 180 grados.

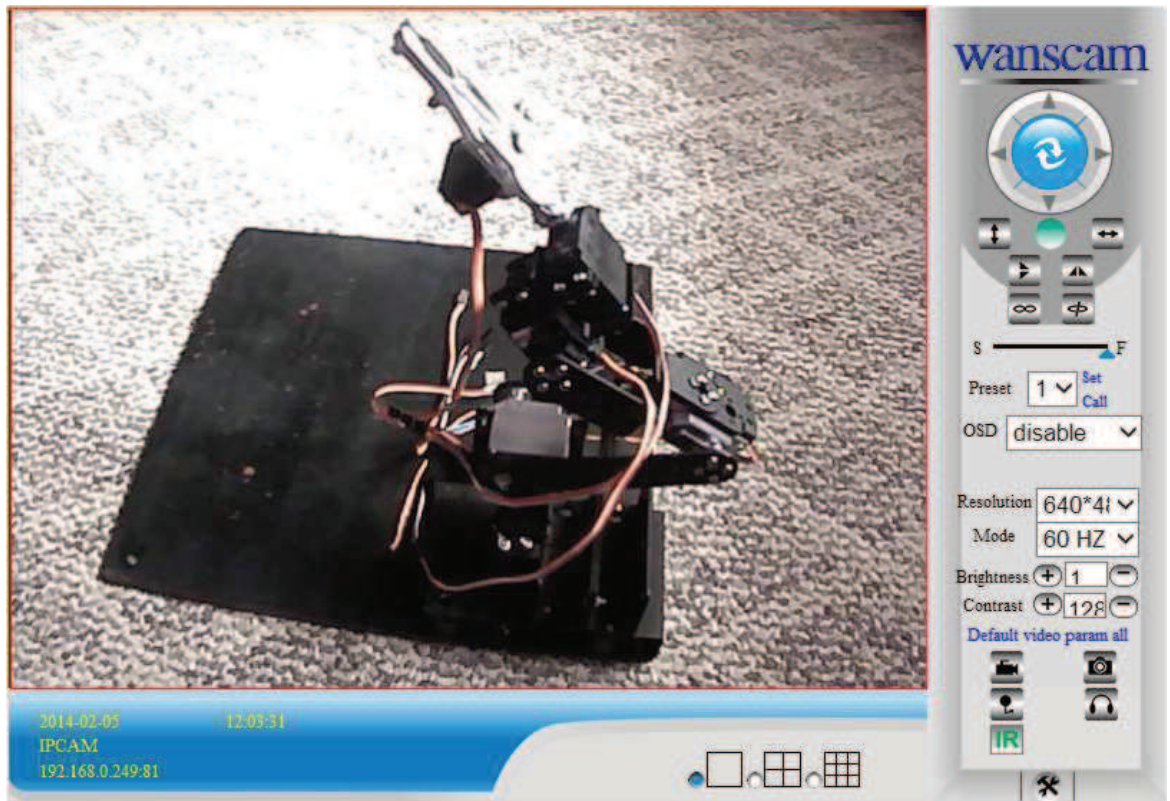


Figura 31 Entorno grafico de la cámara IP

4.2.1.2. Arduino

El Arduino que será ocupado es el mega 2560, este consta de 54 entradas/salidas digitales, 15 de ellas son salidas PWM de las cuales se ocuparon 6 salidas para los movimientos de los servomotores. Se ocupa desde la salida 3 hasta la salida 8. En la figura 32 se puede observar la placa Arduino.

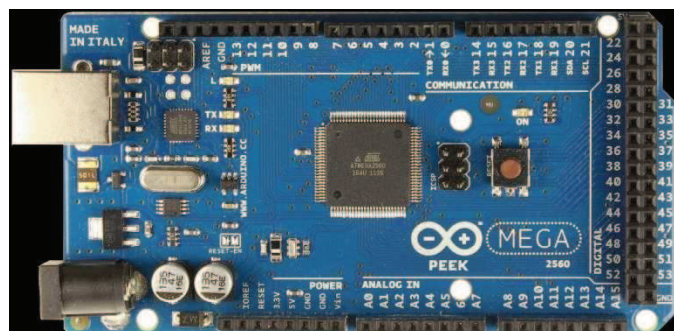


Figura 32 Arduino Mega 2560

4.2.1.3. Arduino Ethernet shield

Este shield hace que el Arduino pueda tener un puerto Ethernet y así comunicarse con la computadora mediante el protocolo TCP/IP, la dirección IP configurada es 192.168.0.177 y el puerto http es 80. Se puede observar en la figura 34 el shield ethernet

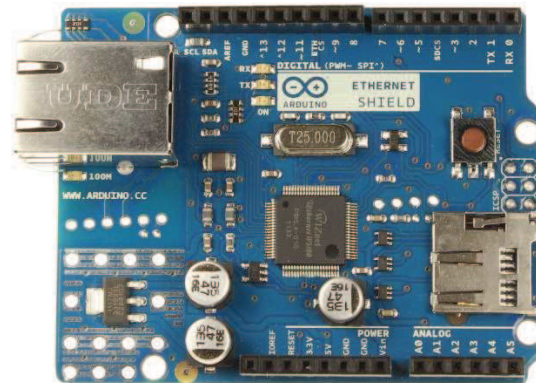


Figura 33 Shield Ethernet Arduino

4.2.1.4. Servomotor

El diseño constará de 2 diferentes tipos de servomotores, el servomotor de la segunda y tercera articulación será de 30 [Kg.cm], el resto de servomotores van a ser de 11 [Kg.cm].

El brazo tiene 360.5 [mm] y el peso que va a manipular el servo es de 0.470 ± 0.005 [kg]. En la figura 34 se puede observar las cargas que van a estar colocadas a lo largo del brazo.

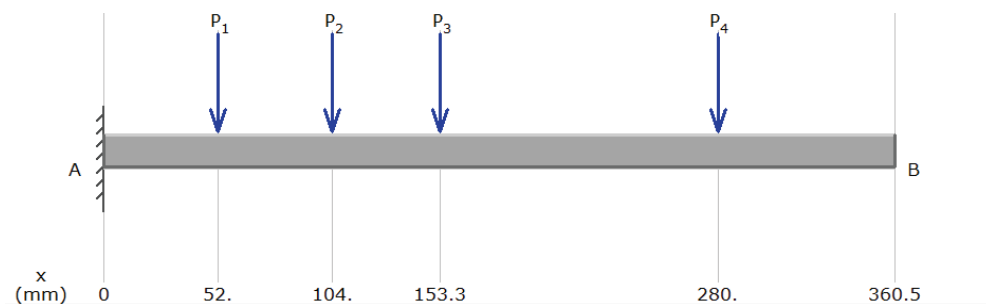


Figura 34. Distribución de las fuerzas a lo largo del brazo

Las fuerzas que están siendo ejercidas son

$$P_1 = 0.4802 [N]$$

$$P_2 = 1.029 [N]$$

$$P_3 = 0.3332 [N]$$

$$P_4 = 2.7538 [N]$$

Para calcular el momento en el punto A se debe hacer la sumatoria de momentos mediante la ecuación [4-5]:

$$\sum M = 0 \quad [4-5]$$

$$M_A - M_{P_1} - M_{P_2} - M_{P_3} - M_{P_4} = 0$$

Donde

M_A = Momento en A

$$M_A = 52[mm] * 0.4802[N] + 104[mm] * 1.029[N] + 153.3[mm] * 0.3332[N] \\ + 280[mm] * 2.7538[N]$$

$$M_A = 954.13 [N * mm]$$

Al dar ese resultado se debe comprobar si el momento del servomotor es igual o mayor que el momento calculado, el momento del servomotor es de 30 [Kg.cm].

Transformando a [N*mm]

$$30[Kg * cm] \cong 2940 [N * mm]$$

Después de hacer el cálculo del servomotor da como resultado que sí logra soportar, para saber cuánto más puede cargar se hace la resta:

$$M_M = M_{Servo} - M_A$$

Donde

M_M = Momento máximo

M_{Servo} = Momento total del servo

M_A = Momento en el punto A

$$M_M = 2940[N * mm] - 954.13[N * mm]$$

$$M_M = 1985.87[N * mm]$$

Para calcular la carga máxima se debe dividir el momento máximo para la longitud total del brazo, como se ve en la siguiente ecuación:

$$P_M = \frac{M_M}{360.5[mm]}$$

$$P_M = 5.508 [N] \cong 0.56204[Kg]$$

El resultado será la carga máxima que moverá el brazo robótico.

4.2.1.5. Análisis de fuente de alimentación y regulador de voltaje

Para la fuente de alimentación hacia los servomotores se requiere 6 voltios para que de esta manera sea posible ocupar su máximo torque y tener un mayor rendimiento. En relación a la fuente de alimentación, se ocupará una que se ocupa en las computadoras de escritorio, esta consta con un voltaje de: ± 3.3 , ± 5 , ± 12 , y soporta una corriente alta, pues se va a ocupar para alimentar los 6 servomotores.

Al no existir una salida de 6 [V], se optó por utilizar un regulador de voltaje y ocupar la salida de 12 voltios. El código de los reguladores de voltaje es LM2596s Board, este está diseñado para soportar 3 [A] es por eso que en la placa se va a colocar 3, para distribuir 2 servomotores a cada uno de estos.

En la figura 35 se indica el circuito:

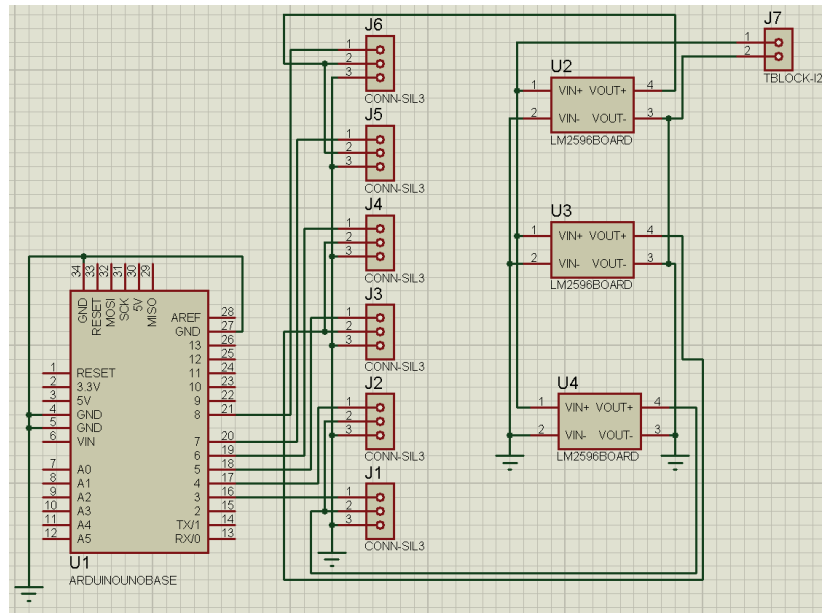


Figura 35. Diseño de la placa de regulación de voltaje en el programa Proteus

En el diseño de la placa, los elementos J1, J2, J3, J4, J5 y J6 representan cada salida de cada uno de los 6 servomotores, esta placa está diseñada para colocar en los pines del Arduino, es por ello que se le coloca el Arduino dentro del esquemático.

En la figura 36, se observa cómo va a ser el diseño de la placa, los pines están distribuidos de tal manera para que esta entre en el Arduino.

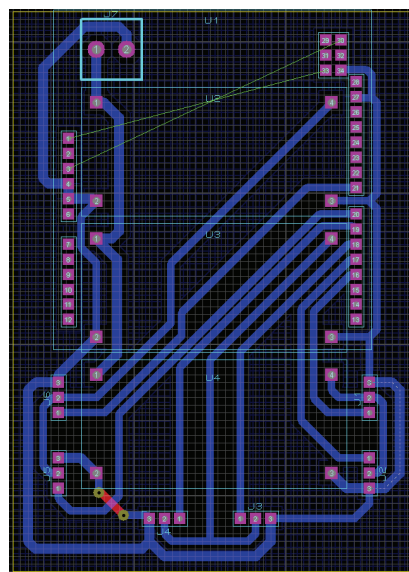


Figura 36 Distribución de la placa a ser construida

4.3. SISTEMA DE CONTROL

4.3.1. PLATAFORMA DE INTERFAZ Y PARÁMETROS DE PROGRAMACIÓN

La interfaz que va a manejar el operario se la realizo en Visual Studio 2012, en el lenguaje de programación Visual Basic.

Para realizar esto se requiere de las librerías que están el Visual Studio, una de las que se requiere es “Imports System.Net”, esta librería hace posible la comunicación mediante Ethernet hacia el Arduino.

Al tener 6 servomotores se requiere enviar 6 datos, para ello se va a hacer el envío de cada servomotor con 2 letras diferentes, el primero es hacia el primer servomotor y así sucesivamente.

Para los 6 servomotores se va a enviar primero de la letra A hasta la F para el sexto, así el Arduino identificará cual servo se va a mover. Dado que el Visual Studio envía dato por dato y no un numero en conjunto, se tiene que mandar una letra para saber cuándo el número llega, es por eso que para el primer servomotor se manda la letra G y para el sexto, la letra L. A continuación se presenta la programación para el primer servomotor.

```
Private Sub Button2_Click_1(sender As Object, e As EventArgs) Handles
Button2.Click
    h1 = Val(TextBox1.Text)
    HScrollBar1.Value = h1
    If h1 > b1 Then
        For p As Integer = b1 + 1 To h1 Step 1
            b1 = b1 + 1
            If Timer2.Enabled = True Then
                Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("A" + b1 + "G")
                TcpClient1.Client.Send(sendBytes)
            End If
        Next
    End If
End Sub
```

```

Else
    For p As Integer = b1 - 1 To h1 Step -1
        b1 = b1 - 1
        If Timer2.Enabled = True Then
            Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("A" + b1 + "G")
            TcpClient1.Client.Send(sendBytes)
        End If
    Next
End If
b1 = h1
p1 = (360 - h1) * 3.141592 / 180
x1 = Math.Cos(p1) * 100 + 198
y1 = Math.Sin(p1) * 100 + 802
End Sub

```

Donde:

b1: es el dato que va a ser enviado hacia el Arduino

h1: es el valor ingresado en el entorno

p: contador

x1, y1: son datos de coordenadas de las líneas del entorno

Lo que están haciendo los bucles es enviar los datos en números uno por uno, por ejemplo, el servomotor al estar en la posición 90 y se le coloca 180, va a ir de uno en uno y no va a saltar directamente hacia el otro dato, eso se realizó para que no se mueva bruscamente el brazo robótico.

En la figura 37 se puede observar el entorno gráfico, la sección en blanco es donde estará el control de la cámara, abajo están los 6 servomotores, y en la parte de la izquierda está el botón "Conectar" para que exista comunicación hacia el Arduino.

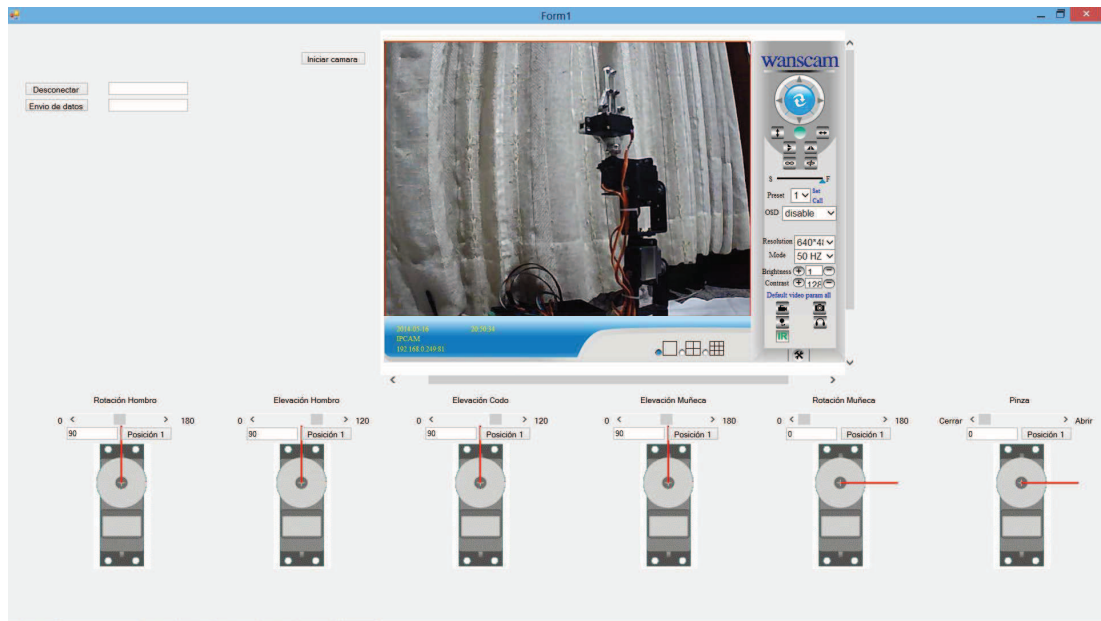


Figura 37. Interfaz Grafica

4.3.2. PROGRAMACIÓN DE PLACA DE CONTROL

Aquí el Arduino recibe los datos que fueron enviados, recibe la primera letra, y recibe de dígito en dígito en código ASCII, es por ello que al dato se le resta 48 para que quede en números enteros.

Cada vez que entre un dato el contador "a" va a sumar 1, empezando en 0, al entrar la primera letra el contador se resetea, y empieza a leer los números, como se muestra a continuación.

```

if (a==1){
  sA1=c-48;
  sB1=c-48;
  sC1=c-48;
  sD1=c-48;
  sE1=c-48;
  sF1=c-48;
}
if (a==2){
  sA2=c-48;

```



```
sB2=c-48;  
sC2=c-48;  
sD2=c-48;  
sE2=c-48;  
sF2=c-48;  
}
```

```
if (a==3){  
sA3=c-48;  
sB3=c-48;  
sC3=c-48;  
sD3=c-48;  
sE3=c-48;  
sF3=c-48;  
}
```

Al saber que llegaron los datos de los servomotores, se puede observar de la siguiente manera

```
if (c=='G'){  
a=0;  
if (b=='4'){  
sA1=sA1*100;  
sA2=sA2*10;  
sA3=sA3+sA2+sA1;  
Serial.println(sA3);  
servo1.write(sA3);  
}  
if (b=='3'){  
sA1=sA1*10;  
sA2=sA2+sA1;  
Serial.println(sA2);  
servo1.write(sA2);  
}
```

```
if (b=='2'){  
  Serial.println(sA1);  
  servo1.write(sA1);  
}  
delay(100);  
}
```

Si es un solo dato, el contador va a ser 2, ya que al leer la primera letra se resta uno del contador y va a leer desde el primer número. Al ser 3, será un número de dos dígitos y la letra al final, y al ser 4 será un número de 3 dígitos y la letra al final.

4.4. CONSTRUCCIÓN

Para empezar a realizar la construcción primero se necesita conocer las piezas que se van a ocupar, como se muestra en la figura 38 y 39:

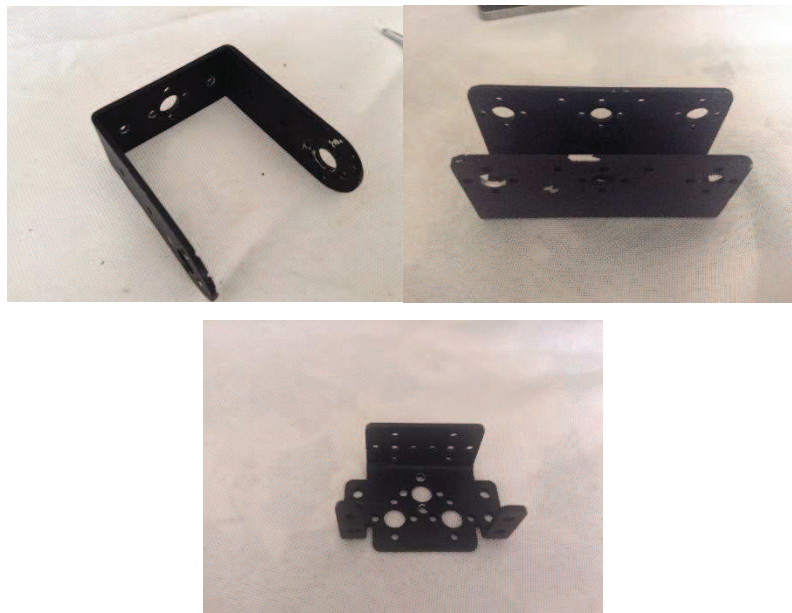


Figura 38. Articulación en U, base y soporte para servomotor



Figura 39. Unión en L, rodamiento y eslabón del eje del servomotor

Se inicia armando el bastidor, el cual estará anclado al piso. Se unen las 5 piezas como se muestra en la figura 40.



Figura 40. Bastidor

Se coloca el soporte del servomotor como se muestra en la figura 41

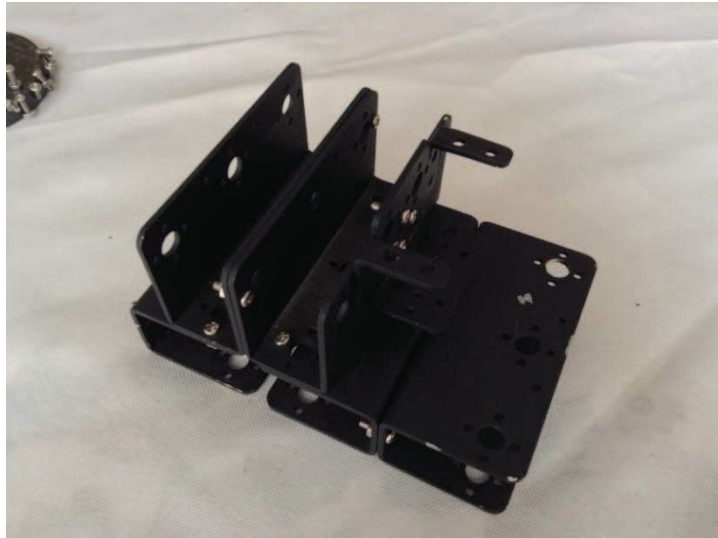


Figura 41. Bastidor con soporte de servomotor

Luego se coloca el servomotor y a la misma altura que esta el eje del servomotor, se coloca un tornillo para que ese sea el eje de movilidad que va a estar con él rodamiento, como se muestra en la figura 42.

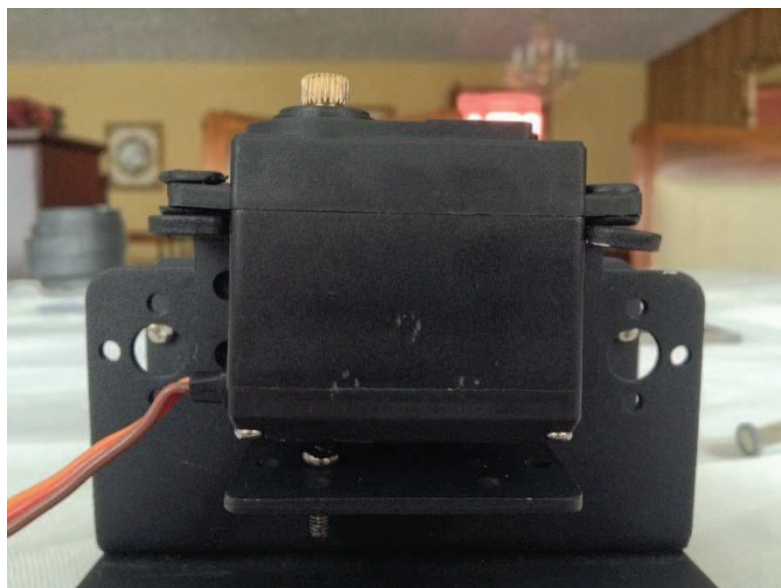


Figura 42. Colocación de servomotor

En la figura 43 se puede observar cómo se encuentra el rodamiento conjuntamente con el tornillo y la tuerca.

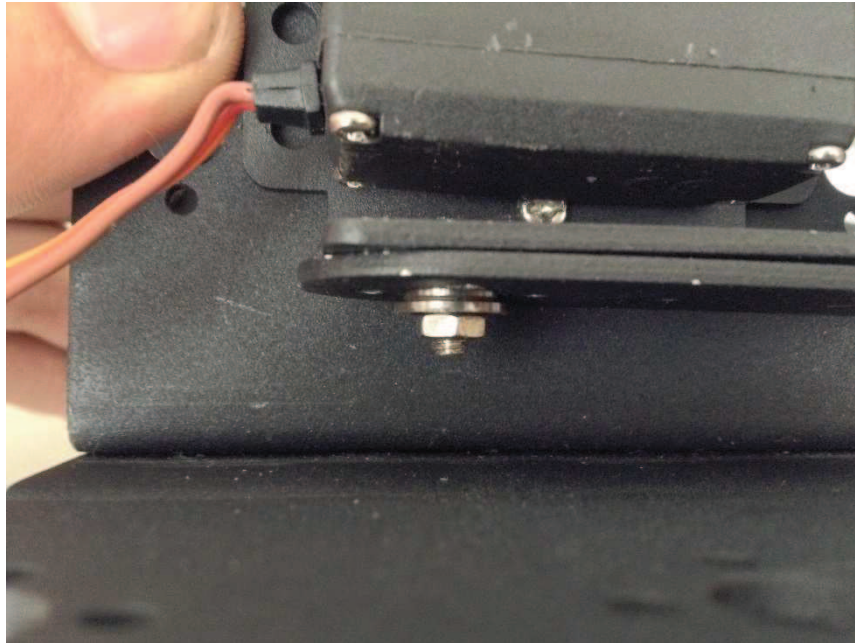


Figura 43. Rodamiento, tuerca y tornillo

Una vez colocado el rodamiento y la pieza en U, se coloca otra pieza de soporte de servomotor, conjuntamente sujetado hacia el eslabón del eje del servomotor, como se observa en la figura 44.

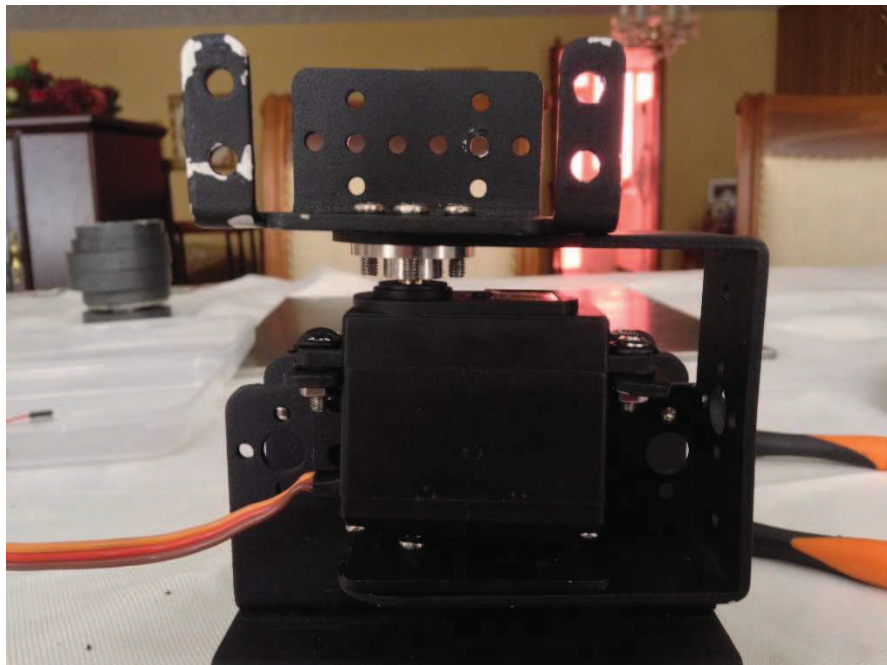


Figura 44. Unión soporte de servomotor con servomotor de rotación

Para hacer considerar segundo grado de libertad, se requiere unir 2 piezas en U como se muestra en la figura 45.

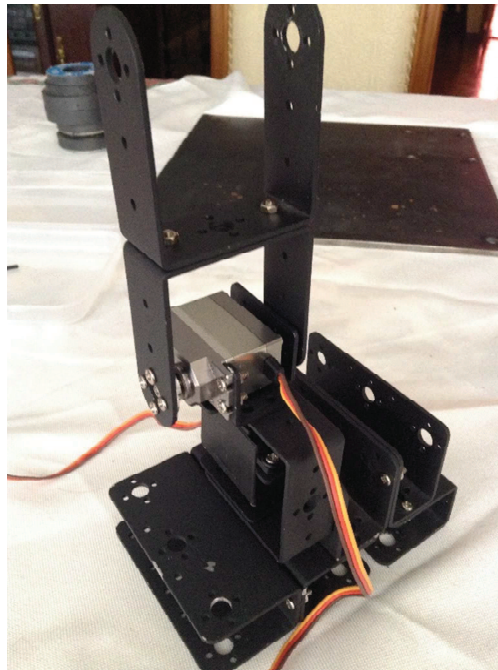


Figura 45. Conexión segundo grado de libertad

Una vez realizado esto, se seguirá ocupando los soportes de servomotores para generar los demás grados de libertad.

Primero se unen los eslabones de la manera que se observa en la figura 46.

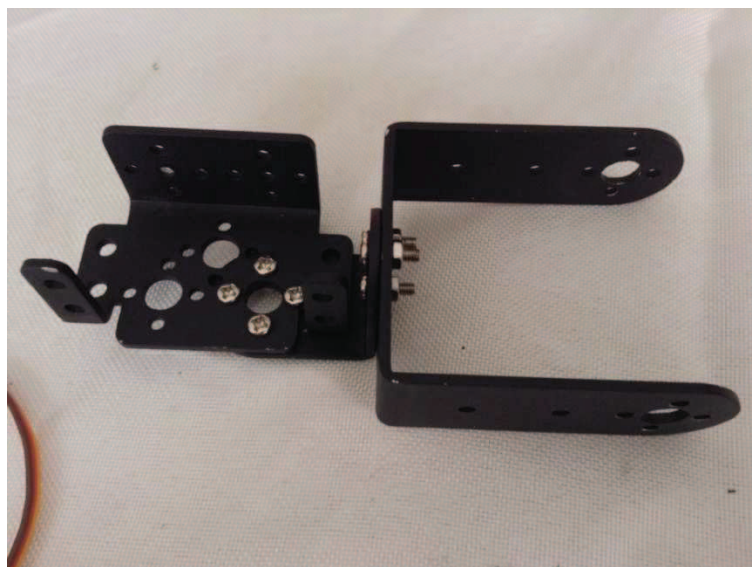


Figura 46. Unión de eslabones para formar la parte superior del brazo

Una vez realizada la anterior unión se coloca con el resto del brazo, como se ve en la figura 47.

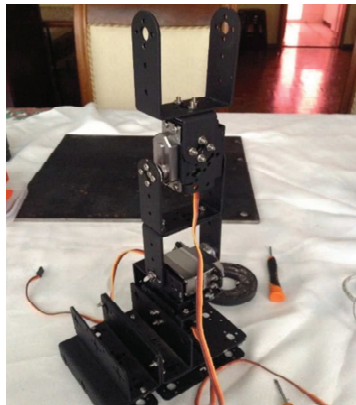


Figura 47. Brazo 3 grados de libertad

Para hacer el movimiento de la muñeca se requiere unir 2 soportes de servomotor, el uno hará que suba y baje, y el otro que rote la pinza, como se muestra en la figura 48.



Figura 48. Unión de eslabones para movimiento de la muñeca

Colocando los servomotores como muestra la figura 49.



Figura 49. Servomotor en eslabones de movimiento de muñeca

Una vez unido los eslabones de la muñeca, el equipo irá quedando como se muestra en la figura 50.

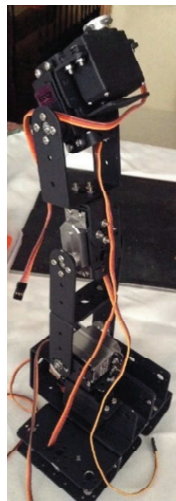


Figura 50. Brazo robot con 5 grados de libertad

La pinza es el último grado de libertad, esta permitirá sujetar los objetos que se desee mover. La pinza se muestra en la figura 51.



Figura 51. Pinza o gripper

Colocando la pinza se termina la construcción del brazo.

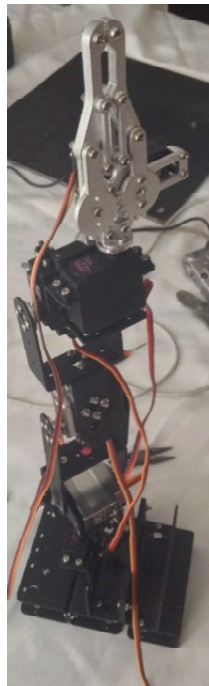


Figura 52. Brazo robot con 6 grados de libertad

Para fijar el brazo robot se tiene una plataforma de 35x35 [cm], la cual se colocará en una esquina, como se muestra en la figura 53.

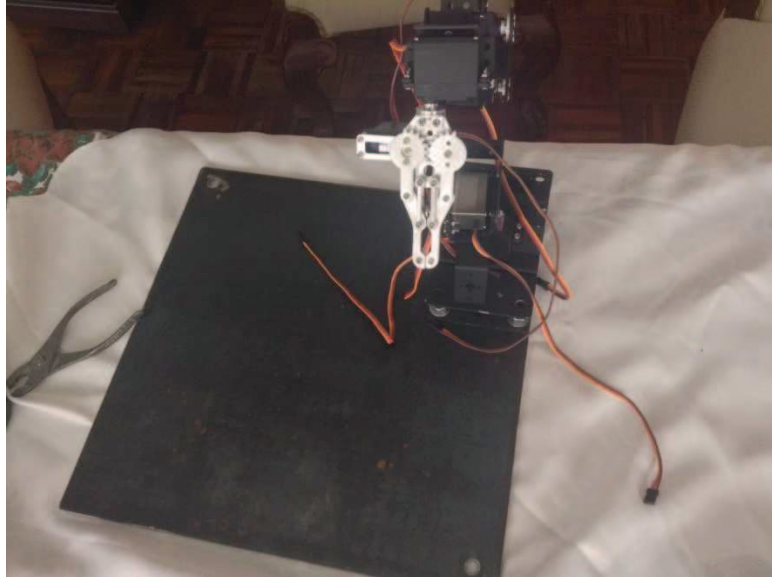


Figura 53. Brazo robot en plataforma

Una vez finalizado el ensamble incluyendo la cámara IP, la placa de control con la placa de potencia y la fuente de alimentación se observara como esta en la figura 54.



Figura 54. Brazo robótico finalizado

5. ANÁLISIS DE RESULTADOS

El objetivo del presente trabajo realizado ha sido lograr que un brazo robótico concebido para simular la carga, transporte y colocación de piezas en un espacio de operación, para ello se llevará a cabo la manipulación de este sistema, mediante un entorno grafico controlado desde un computador.

El brazo robótico va a estar controlado remotamente mediante una placa Arduino, en el entorno grafico se observará éste mediante una cámara IP, ambos (Arduino y cámara IP) estarán conectados a un Wi-Fi para darle un mayor alcance de conectividad.

5.1. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

Una de las pruebas requeridas ha sido el tiempo que se demora la movilidad de una sección de la plataforma hacia otra, este tipo de prueba se ha realizado cuando el brazo robótico esta sin carga, con carga mínima y con carga máxima.

A continuación se presenta 2 figuras donde se podrá observar la posición inicial y la posición final del brazo robótico, la figura 55 muestra la posición inicial y la figura 56 muestra la posición final.

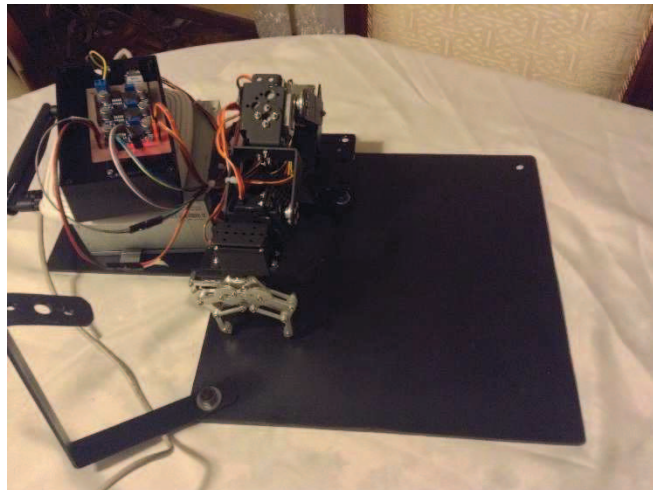


Figura 55. Posición inicial

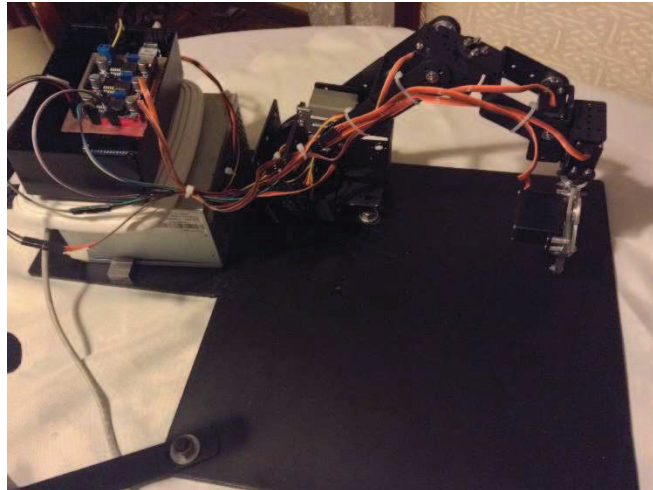


Figura 56. Posición Final

Para realizar las pruebas se generó un código dentro de la programación para que realice una secuencia, esta secuencia empieza con el primer servomotor (rotación) en 0 [grados] y con el segundo en 45 [grados], lo que realizará la secuencia es primero levantar a 90 [grados]. El segundo servomotor, hará rotar 90 [grados] y volverá a bajar hacia 45 grados, las interrupciones colocadas en el programa son de 100 [milisegundos], esta secuencia tomará aproximadamente 18 [s] en terminar. Cabe indicar que se colocó interrupciones para evitar los movimientos bruscos que realizan los servomotores.

La segunda prueba importante que se ha realizado es la máxima distancia de conectividad disponible, para ello se debe colocar el Wi-Fi a una distancia y evitar que la señal pase por paredes o cualquier tipo de obstáculos.

La tercera prueba que se realizó es la adaptación del usuario tanto del entorno gráfico creado como de la movilidad del brazo robótico.

5.1.1. MONTAJE

Al momento de realizar el montaje del brazo se debe fijar la posición en la que van a estar los servomotores y cuáles van a ser su rango de movilidad (figura 57), como se puede ver en la siguiente figura, el segundo servomotor está a 90 [grados], así se puede desplazar hasta 0 [grados] hacia delante y hasta 120 [grados] hacia atrás.

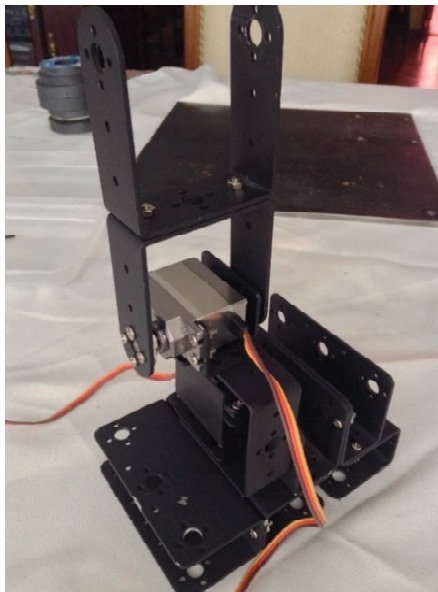


Figura 57. Posición de montaje

De igual manera con las siguientes articulaciones, lo recomendable es ensamblar todas estando en 90 [grados] cada servomotor, para que al momento de programar sea más fácil, como se indica en la figura 58.

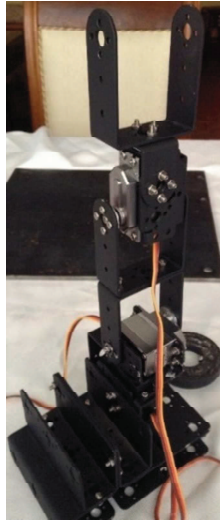


Figura 58. Tercera articulación

5.1.2. PRUEBAS DE MOVIMIENTOS DE CADA ARTICULACIÓN

Las articulaciones del brazo robótico al contener diferentes servomotores tienen distintas características, los servomotores TowerPro MG996r rotan 180 [grados], en cambio los servomotores Turnigy MG959 rotan 120 [grados], esta es una de las razones por las que se va a manejar el brazo dentro de la plataforma establecida.

5.1.3. PRUEBAS DE MOVIMIENTO DEL BRAZO COMPLETO SIN CARGA

El brazo robótico empezará en la posición que se muestra en la figura 59:

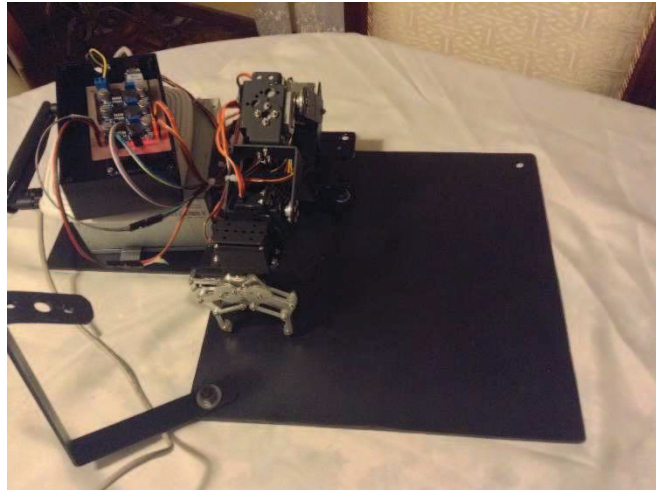


Figura 59. Posición inicial sin carga

La secuencia que se va a ejecutar es primero el levantamiento del segundo servomotor, este servomotor va a rotar de 45 a 90 [grados], luego el servomotor de la base rotará 90 [grados] y luego nuevamente volverá a descender 45 [grados] el segundo servomotor, para quedar en la posición de la figura 60:

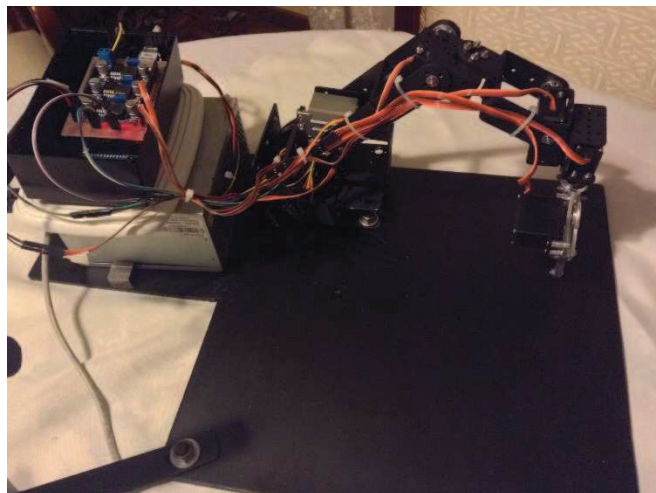


Figura 60. Posición final sin carga

Con las interrupciones anteriormente explicadas el tiempo que se debe demorar en completar toda la secuencia es de 18 [s]. Se realizó 3 repeticiones y se tomó 3 mediciones que son:

Tabla 8. Tiempo de carga

	Dato 1	Dato 2	Dato 3
Sin carga	18.63 [s]	18.44 [s]	18.56 [s]

Al calcular la media aritmética se tiene 18.54 [s]. Se puede observar que este dato es cercano al calculado teóricamente

5.1.4. PRUEBAS DE MOVIMIENTO DEL BRAZO COMPLETO CON CARGA **MÍNIMA**

La carga con la que se va a realizar las pruebas, ha sido una caja pequeña, como se muestra en la figura 61, de 0.022 ± 0.003 [Kg].



Figura 61. Objeto de prueba para carga mínima

Se realizará las pruebas desde los mismos puntos de salida con la misma secuencia realizada anteriormente (figura 62 y 63).

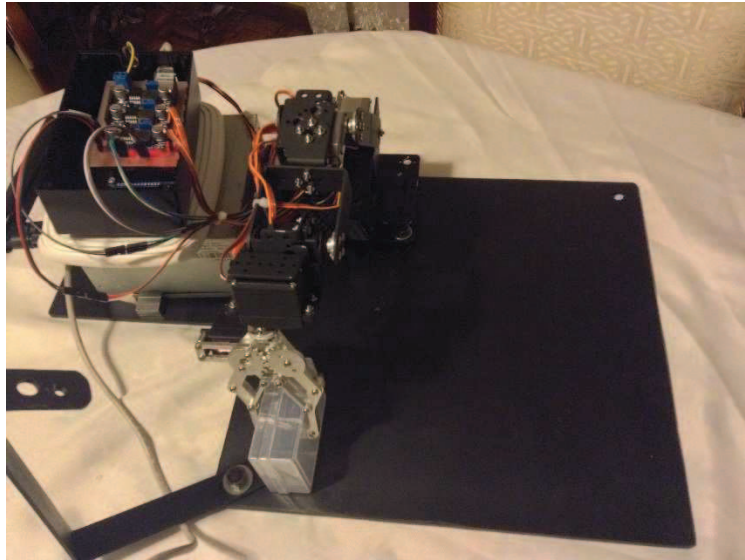


Figura 62. Posición inicial con carga mínima

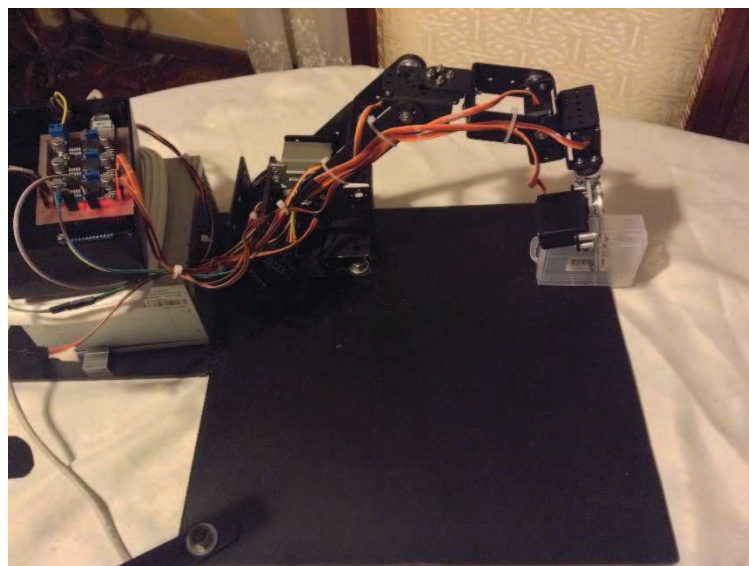


Figura 63. Posición final con carga mínima

Tabla 9. Tiempo de Carga mínima

	Dato 1	Dato 2	Dato 3
Carga Mínima	18.93 [s]	18.37 [s]	18.57 [s]

Al calcular la media aritmética se tiene 18.62 [s]. Se puede observar que este dato es cercano al calculado teóricamente y es cercano al que se obtuvo en la prueba de movimiento sin carga

5.1.5. PRUEBAS DE MOVIMIENTO DEL BRAZO COMPLETO CON CARGA MÁXIMA

La carga con la que se va a realizar esta prueba es un engrane, como se muestra en la figura 64, que pesa 0.350 ± 0.002 [Kg].



Figura 64. Objeto de prueba para carga máxima

Se realizará las pruebas desde los mismos puntos de salida con la misma secuencia realizada anteriormente (figura 65 y 66).

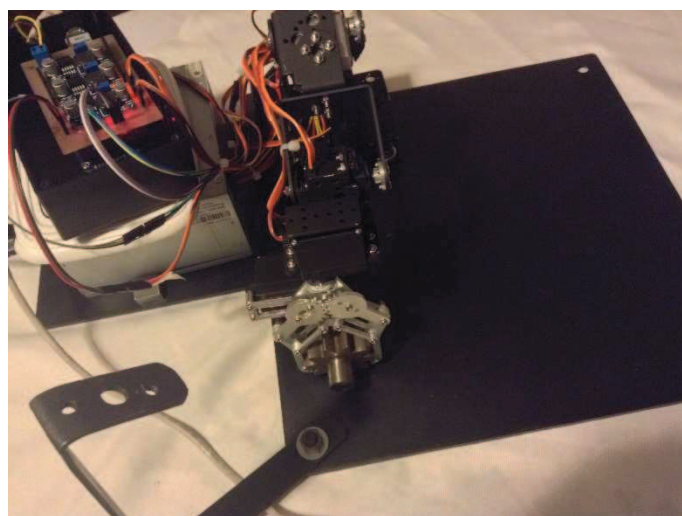


Figura 65. Posición inicial con carga máxima

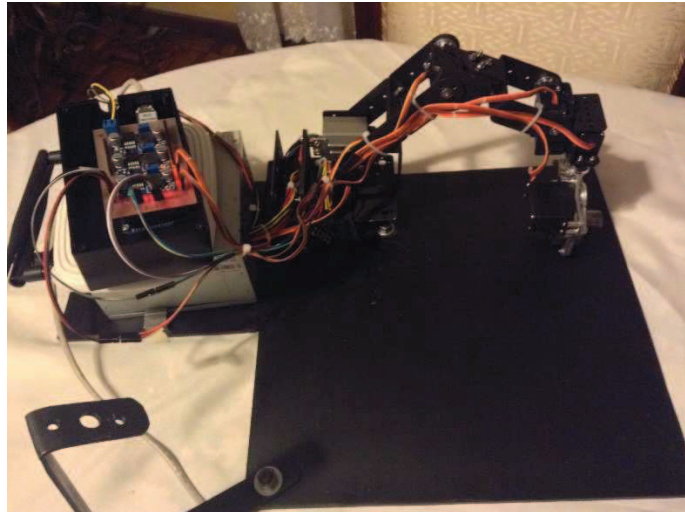


Figura 66. Posición final con carga máxima

Tabla 10. Tiempo de carga máxima

	Dato 1	Dato 2	Dato 3
Carga Máxima	17.94 [s]	18.47 [s]	18.58 [s]

Al calcular la media aritmética se tiene 18.33 [s]. Se puede observar que este dato es cercano al calculado teóricamente, es cercano al que se obtuvo en la prueba de movimiento sin carga y con carga mínima

5.1.6. DISTANCIA MÁXIMA DE CONECTIVIDAD SIN OBSTÁCULOS

Al realizar la prueba en campo, sin muchos obstáculos, solamente con unas pocas paredes, la máxima distancia es de 35-45[m].

5.1.7. ADAPTACIÓN AL ENTORNO GRÁFICO Y A LA MOVILIDAD DEL BRAZO ROBÓTICO

Para la realización de esta prueba se utilizó 3 personas de distintas edades. Cada una debía realizar la secuencia 2 veces. Como se puede ver en la figura 67, el usuario empieza desde esa posición y tiene que desplazarse hacia la

derecha para sujetar el objeto a moverse, como se muestra en la figura 68. Una vez realizado esto, el brazo rota 90 hacia la zona de descarga, como se observa en la figura 69.



Figura 67. Posición inicial del usuario



Figura 68. Posición de sujeción del objeto

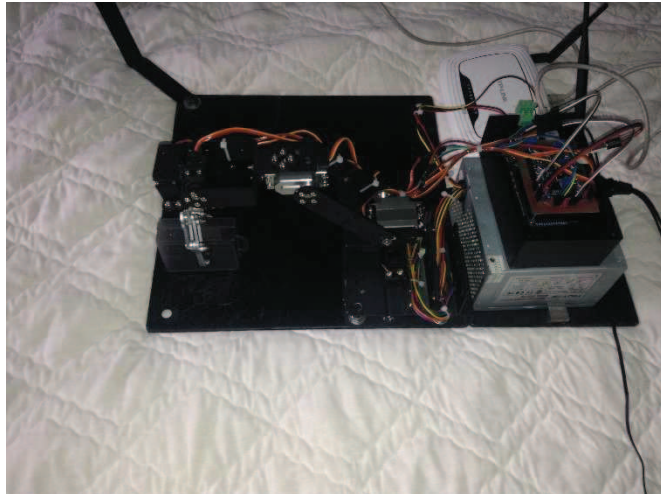


Figura 69. Posición final

Los tres sujetos de prueba constan de, un niño de 7 años, un adulto de 28 años, y un adulto de 55 años. Antes de empezar con la prueba se les dio a conocer la funcionalidad de los controles, tanto de la cámara como del brazo. En la tabla 11.

Tabla 11 Resultados de sujetos de prueba

Edad del sujeto de pruebas	Primera secuencia		Segunda secuencia	
	Tiempo en sujetar el objeto	Tiempo en completar la secuencia	Tiempo en sujetar el objeto	Tiempo en completar la secuencia
7 años	5:30	7:50	4:35	6:58
28 años	3:30	4:35	2:53	3:46
55 años	3:52	5:25	3:00	4:18

En la primera secuencia los tres sujetos, se sentían nerviosos y realizaban movimientos lentos, en cambio en la segunda secuencia, ya sabían lo que debían hacer y realizaron los movimientos de manera rápida y más exacta

5.1.8. MOVILIDAD DEL BRAZO ROBÓTICO DENTRO DE LA INTERFAZ GRÁFICA

En la figura 70 se puede observar la interfaz gráfica, conjuntamente con la visibilidad que proporciona la cámara IP



Figura 70. Visibilidad de brazo robótico realizando movimientos

En la figura 71 se puede observar al brazo robótico al momento de sujetar un objeto

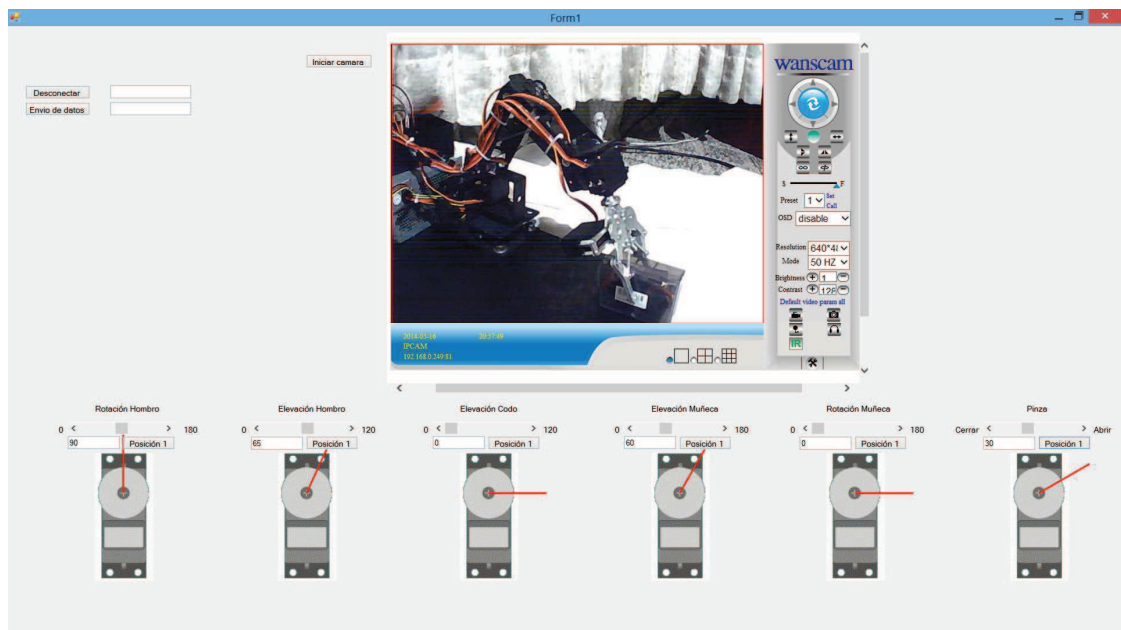


Figura 71. Visibilidad del brazo robótico al momento de la sujeción de un objeto

En la figura 72 se puede observar el brazo robótico al momento de realizar movimientos para la colocación del objeto que está siendo sujetado

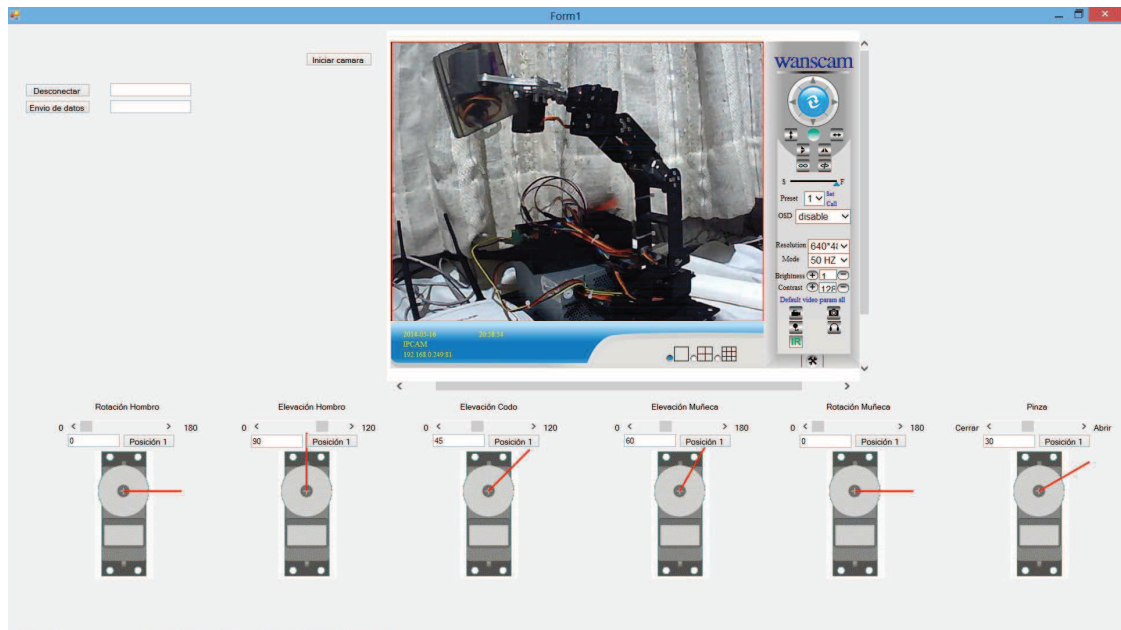


Figura 72. Movimiento del brazo robótico cuando está sujetando un objeto

5.2. COMPROBACIÓN DE HIPÓTESIS

Al finalizar las pruebas se demuestra que el brazo robótico construido es funcional y se ha podido cumplir tanto los objetivos como el alcance propuesto. La mayor carga posible será la mostrada anteriormente, para evitar el sobrecalentamiento, tanto de la placa de control como de los servomotores. El brazo funciona correctamente ya que realiza las instrucciones que se le envía y por tanto se verifica un diseño e implementación funcional y efectiva.

6. CONCLUSIONES Y RECOMENDACIONES

Luego del proceso de diseño y construcción de un prototipo de un brazo robótico tele-operado para ejecutar instrucciones de colocación de objetos, fue posible llegar a las siguientes conclusiones y recomendaciones.

CONCLUSIONES

- En la programación de la placa de control, al momento de configurar cada tipo de servomotor, se debió ajustar distintos tiempos en nivel lógico alto. El TowerPro MG996R se maneja en el rango de 0.6 [ms] cuando está en 0° ; hasta 2.2 [ms] cuando está en 180° ; en el otro Turnigy MG959, de 0.8 [ms] cuando está en 0° hasta 2.9 [ms] cuando está en 180° .
- La manera en que los usuarios se adaptan es rápida, personas no muy acostumbradas al uso de la computadora, se familiarizó rápidamente al entorno gráfico y a los movimientos que realiza el brazo robótico, mientras más usan este entorno y el brazo robótico van a reducir el tiempo que se demoran en completar la secuencia, el usuario que ya está adaptado realizó la secuencia de la sección 5.1.7 aproximadamente en 2:30 minutos.
- Los usuarios nuevos en este entorno, al momento de ocupar la cámara IP, pierden la ubicación espacial, no pueden distinguir bien la profundidad en la que se encuentra el objeto a ser sujetado, además no logran realizar movimientos muy precisos, ya que no están familiarizados con este sistema
- La interfaz gráfica, por lo que consta de una opción de ingreso de grados hacia cada servomotor, fue diseño para gente que tenga algún tipo de análisis geométrico, es por esto que para realizar un proceso más rápido y exacto es necesario ocupar esa opción que da la interfaz gráfica.

- Al momento de codificar los datos que envía el programa Visual Studio hacia el Arduino, se debe tomar en cuenta que servomotor se está controlando y los grados que está recibiendo, para esto se colocó letras al inicio y al final de cada uno de los datos que van a ser enviados hacia la placa de control, el programa dentro de la placa de control, al recibir la letra inicial entiende a que servomotor va dirigido codifica cada dígito después de esa letra hasta la última letra que significa la finalización. Esto ayuda para que el programa realice los comandos correctamente y evitar confusiones y conflictos por parte de la placa de control.
- Al momento de realizar las esperas en las que se desplaza cada grado en cada uno de los servomotores, se lo colocó 100 [milisegundos] con el objetivo primordial de hacerlo lento para que esto facilite el uso a personas que recién van a empezar la manipulación en este entorno gráfico y el uso del brazo robótico.
- Al hacer las pruebas con un peso máximo de 350 ± 5 [g], los reguladores de voltaje se empezaron a calentar, es por ello que para evitar daños en la placa de potencia, la carga máxima ha sido la establecida anteriormente en la sección 5. en el capítulo de análisis y resultados.
- Se empleó la placa de potencia para que todos los servomotores ocupen 6 [V] y así ocupar su máximo torque posible, además se dividió 2 servomotores a cada regulador para disminuir la tensión que generan estos. Además se hizo esto ya que se necesita que entre en la placa de control
- La comunicación Wi-fi que se realizó desde la computadora hacia la placa de control, facilitó el desarrollo del entorno gráfico. De esta manera en el entorno gráfico se puede observar el brazo robótico con la ayuda de la cámara IP y es posible controlar la dirección que se necesite en ese momento.

RECOMENDACIONES

- Se debe tener cuidado al conectar cada uno de los cables de cada servomotor, ya que si se conecta de modo incorrecto se puede dañar o quemar el servomotor o los componentes de la placa.
- Siempre que se pierda la conexión con la tarjeta de control es recomendable reconectar la placa de control y reiniciarlo a la interfaz para evitar problemas tanto en el control del brazo robótico como en la interfaz misma.
- Es recomendable al momento de encender el brazo robótico no estar demasiado cerca de este ya que al momento de desplazarse a su posición inicial se mueve de manera brusca.
- Si en el transcurso del uso llegará a fallar la placa de potencia, sería recomendable colocar cada regulador de voltaje para cada servomotor de forma independiente.
- Si se llegará a desmontar el brazo robótico es recomendable calibrar los servomotores y proseguir con el ensamble.
- En el uso de servomotores siempre se requiere enviar un tren de pulsos para que se mantenga el servomotor en la posición dada y no pierda torque.

BIBLIOGRAFÍA

- Eric Chaves Vega. (2012). *Administración de materiales*. Editorial Universidad Estatal a Distancia. Costa Rica
- Mott, Robert. (2006). *Diseño de elementos de máquina*. Editorial Pearson. Costa Rica
- Barrientos A Peñin L. (2007). *Fundamentos de Robótica*. Editorial McGraw Hill. España
- Budynas, Richard, & Nisbett, Keith. (1999). *Diseño en Ingeniería Mecánica de Shigley*. Editorial McGraw Hill. México
- Tecnológico de Monterrey. (2013). *Robótica*
- Beer, F., & Johnston. (2013). *Mecánica de materiales*. Editorial McGraw Hill. México
- Moya, P. (2013). *Desarrollo de un sistema de control automático en el proceso de corte de tela para la confección de camisetas deportivas para niños*. Quito: Universidad Tecnológica Equinoccial
- Muñoz, G. (2013). *Diseño y construcción de una plataforma móvil a control remoto para un brazo robótico en aplicaciones de seguridad*. Quito: Universidad Tecnológica Equinoccial

ANEXOS

Anexo 1. Programación Arduino

```
#include <Servo.h>
#include <SPI.h>
#include <Ethernet.h>
const int analogOutPin = 2;
const int LedConexion = 53;
char d;
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
Servo servo6;
int sA1=0;
int sA2=0;
int sA3=0;
int sB1=0;
int sB2=0;
int sB3=0;
int sC1=0;
int sC2=0;
int sC3=0;
int sD1=0;
int sD2=0;
int sD3=0;
int sE1=0;
int sE2=0;
```



```
int sE3=0;

int sF1=0;

int sF2=0;

int sF3=0;

int a=0;

int b=0;

byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};

IPAddress ip(192, 168, 0, 177);

EthernetServer server(80);

void setup() {
  Serial.begin(9600);

  servo1.attach(3,800,2700); //mg996r
  servo2.attach(4,800,2900); //mg959
  servo3.attach(5,800,2900); //mg959
  servo4.attach(6,600,2200); //mg996r
  servo5.attach(7,600,2200);
  servo6.attach(8,600,2300); //pinza

  servo1.write(90);
  servo2.write(90);
  servo3.write(90);
  servo4.write(90);
  servo5.write(0);
  servo6.write(180);

  pinMode(LedConexion,OUTPUT);
```

```

while (!Serial) {
}

Ethernet.begin(mac, ip);
server.begin();
Serial.print("server is at ");
Serial.println(Ethernet.localIP());
}

void loop() {
  digitalWrite(LedConexion,LOW);

  EthernetClient client = server.available();
  if (client) {
    digitalWrite(LedConexion,HIGH);
    Serial.println("new client");
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();

        a=a+1;

        b=a+48;

        if (c=='G'){
          a=0;

          if (b=='4'){
            sA1=sA1*100;

            sA2=sA2*10;

            sA3=sA3+sA2+sA1;

            Serial.println(sA3);
          }
        }
      }
    }
  }
}

```

```

servo1.write(sA3);
}
if (b=='3'){
    sA1=sA1*10;
    sA2=sA2+sA1;
    Serial.println(sA2);
    servo1.write(sA2);
}
if (b=='2'){
    Serial.println(sA1);
    servo1.write(sA1);
}
delay(100);
}
if (c=='H'){
a=0;
if (b=='4'){
    sB1=sB1*100;
    sB2=sB2*10;
    sB3=sB3+sB2+sB1;
    Serial.println(sB3);
    Serial.write("B");
    servo2.write(sB3);
}
if (b=='3'){
    sB1=sB1*10;

```

```

sB2=sB2+sB1;
Serial.println(sB2);
servo2.write(sB2);
Serial.write("B");
}
if (b=='2'){
Serial.println(sB1);
servo2.write(sB1);
Serial.write("B");
}
delay(100);
}
if (c=='1'){
a=0;
if (b=='4'){
sC1=sC1*100;
sC2=sC2*10;
sC3=sC3+sC2+sC1;
Serial.write("C");
Serial.println(sC3);
servo3.write(sC3);
}
if (b=='3'){
sC1=sC1*10;
sC2=sC2+sC1;
Serial.write("C");
}

```

```
Serial.println(sC2);
servo3.write(sC2);
}
if (b=='2'){
  Serial.write("C");
  Serial.println(sC1);
  servo3.write(sC1);
}
delay(100);
}
if (c=='J'){
a=0;
if (b=='4'){
  sD1=sD1*100;
  sD2=sD2*10;
  sD3=sD3+sD2+sD1;
  Serial.write("D");
  Serial.println(sD3);
  servo4.write(sD3);
}
if (b=='3'){
  sD1=sD1*10;
  sD2=sD2+sD1;
  Serial.write("D");
  Serial.println(sD2);
  servo4.write(sD2);
```

```
}

if (b=='2'){
  Serial.write("D");
  Serial.println(sD1);
  servo4.write(sD1);
}
delay(100);
}
if (c=='K'){
a=0;
if (b=='4'){
  sE1=sE1*100;
  sE2=sE2*10;
  sE3=sE3+sE2+sE1;
  Serial.write("E");
  Serial.println(sE3);
  servo5.write(sE3);
}
if (b=='3'){
  sE1=sE1*10;
  sE2=sE2+sE1;
  Serial.write("E");
  Serial.println(sE2);
  servo5.write(sE2);
}
}
```

```

if (b=='2'){
    Serial.write("E");
    Serial.println(sE1);
    servo5.write(sE1);
}
delay(100);
}
if (c=='L'){
    a=0;
    if (b=='4'){
        sF1=sF1*100;
        sF2=sF2*10;
        sF3=180-(sF3+sF2+sF1);
        Serial.write("F");
        Serial.println(sF3);
        servo6.write(sF3);
    }
    if (b=='3'){
        sF1=sF1*10;
        sF2=180-(sF2+sF1);
        Serial.write("F");
        Serial.println(sF2);
        servo6.write(sF2);
    }
    if (b=='2'){
        sF1=180-sF1;

```

```
Serial.write("F");
Serial.println(sF1);
servo6.write(sF1);
}
delay(100);
}
if (c=='A'){
  a=a-1;
}
if (c=='B'){
  a=a-1;
}
if (c=='C'){
  a=a-1;
}
if (c=='D'){
  a=a-1;
}
if (c=='E'){
  a=a-1;
}
if (c=='F'){
  a=a-1;
}
if (a==1){
  sA1=c-48;
```



```
sB1=c-48;
sC1=c-48;
sD1=c-48;
sE1=c-48;
sF1=c-48;
}
if (a==2){
sA2=c-48;
sB2=c-48;
sC2=c-48;
sD2=c-48;
sE2=c-48;
sF2=c-48;
}

if (a==3){
sA3=c-48;
sB3=c-48;
sC3=c-48;
sD3=c-48;
sE3=c-48;
sF3=c-48;
}
}
}
delay(1);
```

```
client.stop();  
digitalWrite(LedConexion,LOW);  
Serial.println(" ");  
Serial.println("client disconnected");  
}  
}
```

Anexo 2. Programación Visual Studio

```

Imports System.Net

Public Class Form1
    Dim s1 As Byte
    Dim s2 As Byte
    Dim s3 As Byte
    Dim s4 As Byte
    Dim s5 As Byte
    Dim s6 As Byte
    Dim b1 As String
    Dim b2 As String
    Dim b3 As String
    Dim b4 As String
    Dim b5 As String
    Dim b6 As String
    Dim x1 As Integer
    Dim y1 As Integer
    Dim h1 As String
    Dim p1 As Double
    Dim f1 As Double
    Dim x2 As Integer
    Dim y2 As Integer
    Dim h2 As String
    Dim p2 As Double
    Dim f2 As Double
    Dim x3 As Integer
    Dim y3 As Integer
    Dim h3 As Double
    Dim p3 As Double
    Dim f3 As Double
    Dim x4 As Integer
    Dim y4 As Integer
    Dim h4 As Double
    Dim p4 As Double
    Dim f4 As Double
    Dim x5 As Integer
    Dim y5 As Integer
    Dim h5 As Double
    Dim p5 As Double
    Dim f5 As Double
    Dim x6 As Integer
    Dim y6 As Integer
    Dim h6 As Double
    Dim p6 As Double
    Dim f6 As Double
    Dim cont As Integer
    Dim TcpClient1 As Sockets.TcpClient
    Dim TcpClientStream As Sockets.NetworkStream

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        WebBrowser1.Navigate("http://192.168.0.249:81")
    End Sub

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        x1 = 298
        y1 = 802
        x2 = 613
        y2 = 802

```

```

x3 = 926
y3 = 802
x4 = 1255
y4 = 802
x5 = 1557
y5 = 802
x6 = 1873
y6 = 802
s1 = 0
s2 = 0
s3 = 0
s4 = 0
s5 = 0
s6 = 0
h1 = 90
h2 = 90
h3 = 90
h4 = 90
h5 = 0
h6 = 0
TextBox1.Text = 90
TextBox2.Text = 90
TextBox3.Text = 90
TextBox4.Text = 90
TextBox5.Text = 0
TextBox6.Text = 0
HScrollBar1.Value = 90
HScrollBar2.Value = 90
HScrollBar3.Value = 90
HScrollBar4.Value = 90
HScrollBar5.Value = 0
HScrollBar6.Value = 0
b1 = h1
p1 = (360 - h1) * 3.141592 / 180
x1 = Math.Cos(p1) * 100 + 198
y1 = Math.Sin(p1) * 100 + 802
b2 = h2
p2 = (360 - h2) * 3.141592 / 180
x2 = Math.Cos(p2) * 100 + 513
y2 = Math.Sin(p2) * 100 + 802
b3 = h3
p3 = (360 - h3) * 3.141592 / 180
x3 = Math.Cos(p3) * 100 + 826
y3 = Math.Sin(p3) * 100 + 802
b4 = h4
p4 = (360 - h4) * 3.141592 / 180
x4 = Math.Cos(p4) * 100 + 1155
y4 = Math.Sin(p4) * 100 + 802
b5 = h5
p5 = (360 - h5) * 3.141592 / 180
x5 = Math.Cos(p5) * 100 + 1457
y5 = Math.Sin(p5) * 100 + 802
b6 = h6
p6 = (360 - h6) * 3.141592 / 180
x6 = Math.Cos(p6) * 100 + 1773
y6 = Math.Sin(p6) * 100 + 802
PictureBox1.SendToBack()
PictureBox2.SendToBack()
PictureBox3.SendToBack()
PictureBox4.SendToBack()

```

```

PictureBox5.SendToBack()
PictureBox6.SendToBack()
PictureBox1.BackColor = Color.Transparent
PictureBox2.BackColor = Color.Transparent
PictureBox3.BackColor = Color.Transparent
PictureBox4.BackColor = Color.Transparent
PictureBox5.BackColor = Color.Transparent
PictureBox6.BackColor = Color.Transparent

```

End Sub

```

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick

```

```

    LineShape1.X1 = 198
    LineShape1.Y1 = 802
    LineShape1.X2 = x1
    LineShape1.Y2 = y1
    LineShape2.X1 = 513
    LineShape2.Y1 = 802
    LineShape2.X2 = x2
    LineShape2.Y2 = y2
    LineShape3.X1 = 826
    LineShape3.Y1 = 802
    LineShape3.X2 = x3
    LineShape3.Y2 = y3
    LineShape4.X1 = 1155
    LineShape4.Y1 = 802
    LineShape4.X2 = x4
    LineShape4.Y2 = y4
    LineShape5.X1 = 1457
    LineShape5.Y1 = 802
    LineShape5.X2 = x5
    LineShape5.Y2 = y5
    LineShape6.X1 = 1773
    LineShape6.Y1 = 802
    LineShape6.X2 = x6
    LineShape6.Y2 = y6

```

End Sub

```

Private Sub WebBrowser1_DocumentCompleted(sender As Object, e As
WebBrowserDocumentCompletedEventArgs) Handles WebBrowser1.DocumentCompleted

```

End Sub

```

Private Sub Button2_Click_1(sender As Object, e As EventArgs) Handles
Button2.Click

```

```

    h1 = Val(TextBox1.Text)
    HScrollBar1.Value = h1
    If Val(h1) > Val(b1) Then
        For p As Integer = b1 + 1 To h1 Step 1
            b1 = b1 + 1
            If Timer2.Enabled = True Then
                Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("A" + b1 + "G")
                TcpClient1.Client.Send(sendBytes)
            End If
        Next

```

```

Else
    For p As Integer = b1 - 1 To h1 Step -1
        b1 = b1 - 1
        If Timer2.Enabled = True Then
            Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("A" + b1 + "G")
            TcpClient1.Client.Send(sendBytes)
        End If
    Next
End If
If Timer2.Enabled = True Then
    Dim sendBytes() As Byte = System.Text.Encoding.ASCII.GetBytes("A"
+ TextBox1.Text + "G")
    TcpClient1.Client.Send(sendBytes)
End If
b1 = h1
p1 = (360 - h1) * 3.141592 / 180
x1 = Math.Cos(p1) * 100 + 198
y1 = Math.Sin(p1) * 100 + 802
End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
    h2 = Val(TextBox2.Text)
    HScrollBar2.Value = h2
    If Val(h2) > Val(b2) Then
        For p As Integer = b2 + 1 To h2 Step 1
            b2 = b2 + 1
            If Timer2.Enabled = True Then
                Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("B" + b2 + "H")
                TcpClient1.Client.Send(sendBytes)
            End If
        Next
    Else
        For p As Integer = b2 - 1 To h2 Step -1
            b2 = b2 - 1
            If Timer2.Enabled = True Then
                Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("B" + b2 + "H")
                TcpClient1.Client.Send(sendBytes)
            End If
        Next
    End If
    If Timer2.Enabled = True Then
        Dim sendBytes() As Byte = System.Text.Encoding.ASCII.GetBytes("B"
+ TextBox2.Text + "H")
        TcpClient1.Client.Send(sendBytes)
    End If
    b2 = h2
    p2 = (360 - h2) * 3.141592 / 180
    x2 = Math.Cos(p2) * 100 + 513
    y2 = Math.Sin(p2) * 100 + 802
End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles
Button4.Click
    h3 = Val(TextBox3.Text)
    HScrollBar3.Value = h3

```

```

    If h3 > b3 Then
        For p As Integer = b3 + 1 To h3 Step 1
            b3 = b3 + 1
            If Timer2.Enabled = True Then
                Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("C" + b3 + "I")
                TcpClient1.Client.Send(sendBytes)
            End If
        Next
    Else
        For p As Integer = b3 - 1 To h3 Step -1
            b3 = b3 - 1
            If Timer2.Enabled = True Then
                Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("C" + b3 + "I")
                TcpClient1.Client.Send(sendBytes)
            End If
        Next
    End If
    If Timer2.Enabled = True Then
        Dim sendBytes() As Byte = System.Text.Encoding.ASCII.GetBytes("C"
+ TextBox3.Text + "I")
        TcpClient1.Client.Send(sendBytes)
    End If
    b3 = h3
    p3 = (360 - h3) * 3.141592 / 180
    x3 = Math.Cos(p3) * 100 + 826
    y3 = Math.Sin(p3) * 100 + 802
End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles
Button5.Click
    h4 = Val(TextBox4.Text)
    HScrollBar4.Value = h4
    If h4 > b4 Then
        For p As Integer = b4 + 1 To h4 Step 1
            b4 = b4 + 1
            If Timer2.Enabled = True Then
                Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("D" + b4 + "J")
                TcpClient1.Client.Send(sendBytes)
            End If
        Next
    Else
        For p As Integer = b4 - 1 To h4 Step -1
            b4 = b4 - 1
            If Timer2.Enabled = True Then
                Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("D" + b4 + "J")
                TcpClient1.Client.Send(sendBytes)
            End If
        Next
    End If
    If Timer2.Enabled = True Then
        Dim sendBytes() As Byte = System.Text.Encoding.ASCII.GetBytes("D"
+ TextBox4.Text + "J")
        TcpClient1.Client.Send(sendBytes)
    End If
    b4 = h4
    p4 = (360 - h4) * 3.141592 / 180

```



```

        x4 = Math.Cos(p4) * 100 + 1155
        y4 = Math.Sin(p4) * 100 + 802
    End Sub

    Private Sub Button6_Click(sender As Object, e As EventArgs) Handles
Button6.Click
        h5 = Val(TextBox5.Text)
        HScrollBar5.Value = h5
        If h5 > b5 Then
            For p As Integer = b5 + 1 To h5 Step 1
                b5 = b5 + 1
                If Timer2.Enabled = True Then
                    Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("E" + b5 + "K")
                    TcpClient1.Client.Send(sendBytes)
                End If
            Next
        Else
            For p As Integer = b5 - 1 To h5 Step -1
                b5 = b5 - 1
                If Timer2.Enabled = True Then
                    Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("E" + b5 + "K")
                    TcpClient1.Client.Send(sendBytes)
                End If
            Next
        End If
        If Timer2.Enabled = True Then
            Dim sendBytes() As Byte = System.Text.Encoding.ASCII.GetBytes("E"
+ TextBox5.Text + "K")
            TcpClient1.Client.Send(sendBytes)
        End If
        b5 = h5
        p5 = (360 - h5) * 3.141592 / 180
        x5 = Math.Cos(p5) * 100 + 1457
        y5 = Math.Sin(p5) * 100 + 802
    End Sub

    Private Sub Button7_Click(sender As Object, e As EventArgs) Handles
Button7.Click
        h6 = Val(TextBox6.Text)
        HScrollBar6.Value = h6
        If h6 > b6 Then
            For p As Integer = b6 + 1 To h6 Step 1
                b6 = b6 + 1
                If Timer2.Enabled = True Then
                    Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("F" + b6 + "L")
                    TcpClient1.Client.Send(sendBytes)
                End If
            Next
        Else
            For p As Integer = b6 - 1 To h6 Step -1
                b6 = b6 - 1
                If Timer2.Enabled = True Then
                    Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("F" + b6 + "L")
                    TcpClient1.Client.Send(sendBytes)
                End If
            Next
        End If
    Next

```

```

        End If
        If Timer2.Enabled = True Then
            Dim sendBytes() As Byte = System.Text.Encoding.ASCII.GetBytes("F"
+ TextBox6.Text + "L")
            TcpClient1.Client.Send(sendBytes)
        End If
        b6 = h6
        p6 = (360 - h6) * 3.141592 / 180
        x6 = Math.Cos(p6) * 100 + 1773
        y6 = Math.Sin(p6) * 100 + 802
    End Sub

    Private Sub HScrollBar1_Scroll(sender As Object, e As ScrollEventArgs)
Handles HScrollBar1.Scroll

        h1 = HScrollBar1.Value
        If h1 <> s1 Then
            s1 = h1
            b1 = h1
            p1 = (360 - h1) * 3.141592 / 180
            x1 = Math.Cos(p1) * 100 + 198
            y1 = Math.Sin(p1) * 100 + 802
            TextBox1.Text = h1
            If Timer2.Enabled = True Then
                Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("A" + TextBox1.Text + "G")
                TcpClient1.Client.Send(sendBytes)
            End If
        End If
    End Sub

    Private Sub HScrollBar2_Scroll(sender As Object, e As ScrollEventArgs)
Handles HScrollBar2.Scroll

        h2 = HScrollBar2.Value
        If h2 <> s2 Then
            s2 = h2
            b2 = h2
            p2 = (360 - h2) * 3.141592 / 180
            x2 = Math.Cos(p2) * 100 + 513
            y2 = Math.Sin(p2) * 100 + 802
            TextBox2.Text = h2
            If Timer2.Enabled = True Then
                Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("B" + TextBox2.Text + "H")
                TcpClient1.Client.Send(sendBytes)
            End If
        End If
    End Sub

    Private Sub HScrollBar3_Scroll(sender As Object, e As ScrollEventArgs)
Handles HScrollBar3.Scroll

        h3 = HScrollBar3.Value
        If h3 <> s3 Then
            s3 = h3
            b3 = h3
            p3 = (360 - h3) * 3.141592 / 180
            x3 = Math.Cos(p3) * 100 + 826
            y3 = Math.Sin(p3) * 100 + 802
            TextBox3.Text = h3
            If Timer2.Enabled = True Then

```

```

        Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("C" + TextBox3.Text + "I")
        TcpClient1.Client.Send(sendBytes)
    End If
End If
End Sub

Private Sub HScrollBar4_Scroll(sender As Object, e As ScrollEventArgs)
Handles HScrollBar4.Scroll
    h4 = HScrollBar4.Value
    If h4 <> s4 Then
        s4 = h4
        b4 = h4
        h4 = HScrollBar4.Value
        p4 = (360 - h4) * 3.141592 / 180
        x4 = Math.Cos(p4) * 100 + 1155
        y4 = Math.Sin(p4) * 100 + 802
        TextBox4.Text = h4
        If Timer2.Enabled = True Then
            Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("D" + TextBox4.Text + "J")
            TcpClient1.Client.Send(sendBytes)
        End If
    End If
End Sub

Private Sub HScrollBar5_Scroll(sender As Object, e As ScrollEventArgs)
Handles HScrollBar5.Scroll
    h5 = HScrollBar5.Value
    If h5 <> s5 Then
        s5 = h5
        b5 = h5
        p5 = (360 - h5) * 3.141592 / 180
        x5 = Math.Cos(p5) * 100 + 1457
        y5 = Math.Sin(p5) * 100 + 802
        TextBox5.Text = h5
        If Timer2.Enabled = True Then
            Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("E" + TextBox5.Text + "K")
            TcpClient1.Client.Send(sendBytes)
        End If
    End If
End Sub

Private Sub HScrollBar6_Scroll(sender As Object, e As ScrollEventArgs)
Handles HScrollBar6.Scroll
    h6 = HScrollBar6.Value
    If h6 <> s6 Then
        s6 = h6
        b6 = h6
        p6 = (360 - h6) * 3.141592 / 180
        x6 = Math.Cos(p6) * 100 + 1773
        y6 = Math.Sin(p6) * 100 + 802
        TextBox6.Text = h6
        If Timer2.Enabled = True Then
            Dim sendBytes() As Byte =
System.Text.Encoding.ASCII.GetBytes("F" + TextBox6.Text + "L")
            TcpClient1.Client.Send(sendBytes)
        End If
    End If
End Sub

```

```

End Sub

Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles
TextBox1.TextChanged
    Dim a As Integer
    a = Val(TextBox1.Text)
    If a > 180 Or a < 0 Then
        TextBox1.Text = ""
    End If
End Sub

End Sub

Private Sub TextBox2_TextChanged(sender As Object, e As EventArgs) Handles
TextBox2.TextChanged
    Dim a As Integer
    a = Val(TextBox2.Text)
    If a > 120 Or a < 0 Then
        TextBox2.Text = ""
    End If
End Sub

End Sub

Private Sub TextBox3_TextChanged(sender As Object, e As EventArgs) Handles
TextBox3.TextChanged
    Dim a As Integer
    a = Val(TextBox3.Text)
    If a > 120 Or a < 0 Then
        TextBox3.Text = ""
    End If
End Sub

End Sub

Private Sub TextBox4_TextChanged(sender As Object, e As EventArgs) Handles
TextBox4.TextChanged
    Dim a As Integer
    a = Val(TextBox4.Text)
    If a > 180 Or a < 0 Then
        TextBox4.Text = ""
    End If
End Sub

End Sub

Private Sub TextBox5_TextChanged(sender As Object, e As EventArgs) Handles
TextBox5.TextChanged
    Dim a As Integer
    a = Val(TextBox5.Text)
    If a > 180 Or a < 0 Then
        TextBox5.Text = ""
    End If
End Sub

End Sub

Private Sub TextBox6_TextChanged(sender As Object, e As EventArgs) Handles
TextBox6.TextChanged
    Dim a As Integer
    a = Val(TextBox6.Text)
    If a > 80 Or a < 0 Then
        TextBox6.Text = ""
    End If
End Sub

```

```

End Sub

Private Sub Button8_Click(sender As Object, e As EventArgs) Handles
Button8.Click
    If Button8.Text = "Conectar" Then
        TcpClient1 = New Sockets.TcpClient("192.168.0.177", 80)
        Timer2.Enabled = True
        TcpClientStream = TcpClient1.GetStream()
        Button8.Text = "Desconectar"
        Dim sendBytes() As Byte = System.Text.Encoding.ASCII.GetBytes("A"
+ TextBox1.Text + "G")
        TcpClient1.Client.Send(sendBytes)
    ElseIf Button8.Text = "Desconectar" Then
        Button8.Text = "Conectar"
        Timer2.Enabled = False
        TcpClient1.Close()

    End If

End Sub

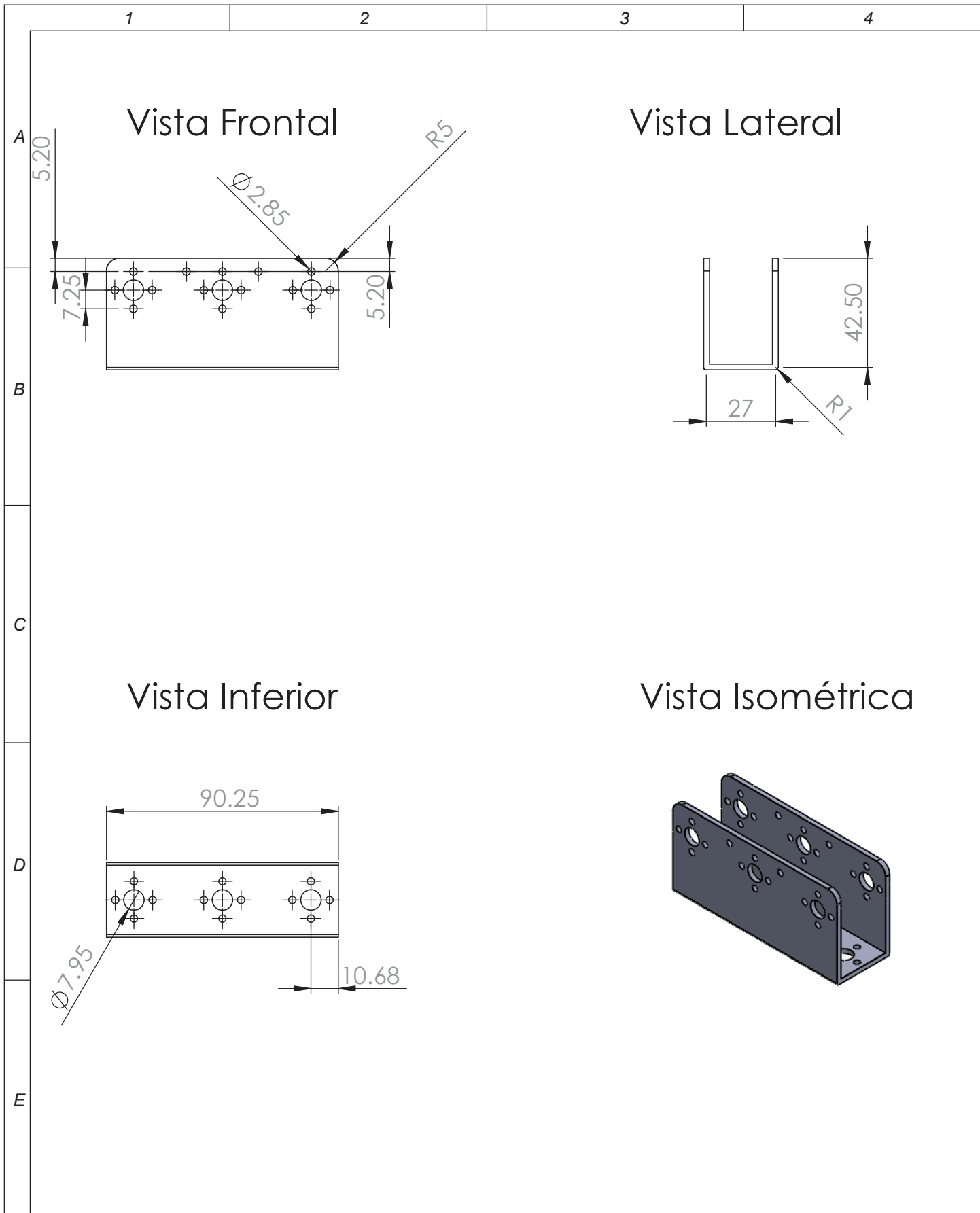
Private Sub Button9_Click(sender As Object, e As EventArgs) Handles
Button9.Click
    Dim sendBytes() As Byte = System.Text.Encoding.ASCII.GetBytes("A" +
TextBox8.Text)
    'sendBytes = TextBox8.Text
    TcpClient1.Client.Send(sendBytes)

End Sub

Private Sub Timer2_Tick(sender As Object, e As EventArgs) Handles
Timer2.Tick
    If TcpClientStream.DataAvailable = True Then
        Dim rcvbytes(TcpClient1.ReceiveBufferSize) As Byte
        TcpClientStream.Read(rcvbytes, 0,
CInt(TcpClient1.ReceiveBufferSize))
        TextBox3.Text = System.Text.Encoding.ASCII.GetString(rcvbytes)
    End If
End Sub
End Class

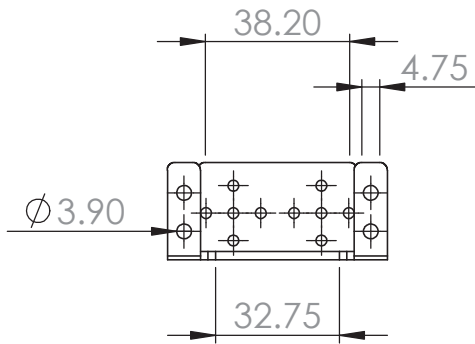
```

Anexo 3. Planos y diseños del brazo robótico

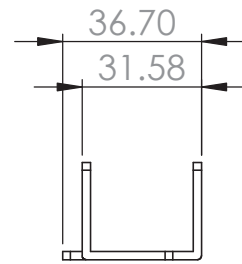


				TOLERANCIA: ±0.02 mm	PESO [Kg]: 0.051	MATERIAL: Aluminio 1050-0 Espesor=2.5 mm	
					<i>Fecha</i> Abril, 2014	<i>Nombre</i> Paul Bustos	ESCALA: 1:2
					<i>Revisado</i>		
					<i>Aprobado</i>		
				FIRMA / EMPRESA:		DIBUJO Nº: Anexo 3.1.	HOJA: 1
<i>Edición</i>	<i>Modificación</i>	<i>Fecha</i>	<i>Nombre</i>				

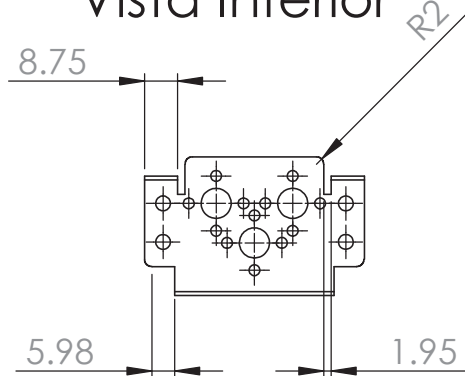
Vista Frontal



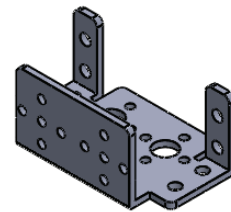
Vista Lateral



Vista Inferior

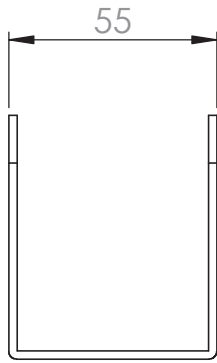


Vista Isométrica

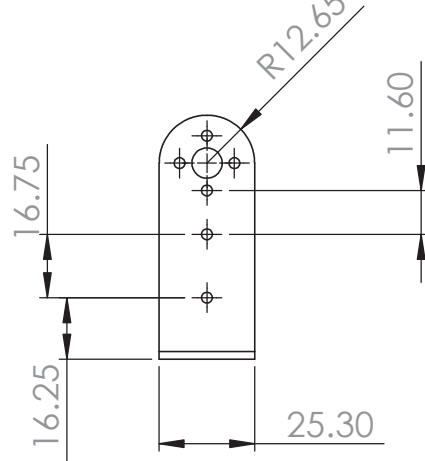


				TOLERANCIA: ±0.02 mm	PESO [Kg]: 0.015	MATERIAL: Aluminio 1050-0 Espesor=2.5 mm	
						Soporte servo motor	ESCALA: 1:2
						Anexo 3.2.	HOJA: 2
Edición	Modificación	Fecha	Nombre	FIRMA / EMPRESA:		DIBUJO Nº:	

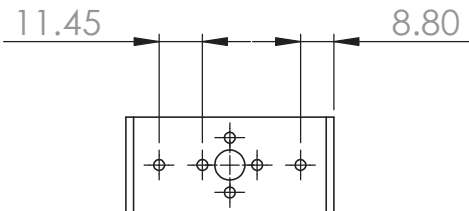
Vista Frontal



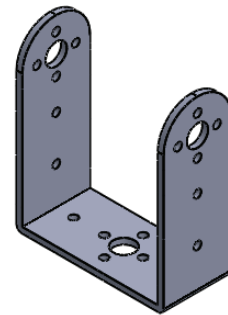
Vista Lateral



Vista Inferior



Vista Isométrica



				TOLERANCIA: ±0.02 mm	PESO [Kg]: 0.022	MATERIAL: Aluminio 1050-0 Espesor=2.5 mm	
					<i>Fecha</i> Abril, 2014	<i>Nombre</i> Paul Bustos	ESCALA: 1:2
					<i>Revisado</i>		HOJA: 3
					<i>Aprobado</i>		
				FIRMA / EMPRESA:		DIBUJO Nº: Anexo 3.3.	
<i>Edición</i>	<i>Modificación</i>	<i>Fecha</i>	<i>Nombre</i>				

