



UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIA Y TECNOLOGÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA

**“Estudio y diseño de un robot explorador de ductos mediante la
captura de imágenes en tiempo real utilizando un interface gráfico
remoto“**

**Trabajo de graduación previo a la obtención del título de:
INGENIERO ELECTRÓNICO.**

Autores:

**MIGUEL SANTIAGO DÍAZ ANDRADE
TATIANA ANDREA SARMIENTO ASTUDILLO**

Director:

OMAR SANTIAGO ALVARADO CANDO

CUENCA, ECUADOR

2015

DEDICATORIA

Este trabajo va dedicado a mis padres por su sacrificio y amor incondicional, a mis abuelos y hermanos que han sido mi pilar y ejemplo en el día a día, y a Pedro quien ha sido mi impulso, soporte y consejero en todo momento.

Tatiana Sarmiento A.

DEDICATORIA

Dedicado a mi abuelo quien me ha demostrado que nunca es tarde para aprender.

Miguel Díaz A.

AGRADECIMIENTOS

Quisiéramos agradecer a todos nuestros compañeros y amigos que compartieron las aulas y el conocimiento con nosotros durante tantos años, con cada uno de ustedes aprendimos todo lo que sabemos ahora, a los profesores de la Universidad del Azuay por el apoyo en este trabajo, en especial a los miembros del tribunal y nuestros directores de tesis, que nos han guiado durante este trabajo. También quisiéramos hacer llegar un agradecimiento especial a la empresa INGMATRICOM y al Ing. Fernando Chica por su gran aporte.

Tatiana Sarmiento A.

Miguel Díaz A.

ÍNDICE DE CONTENIDOS

CONTENIDOS	Página
DEDICATORIA	II
AGRADECIMIENTOS	IV
ÍNDICE DE FIGURAS	viii
ÍNDICE DE TABLAS	xii
ÍNDICE DE ANEXOS	xiii
INTRODUCCIÓN	1
CAPÍTULO I MARCO TEÓRICO	4
1.1 INTRODUCCIÓN	4
1.2 DEFINICIÓN	4
1.3 PRINCIPALES CARACTERÍSTICAS DE UN ROBOT	5
1.3.1 Grados de libertad	5
1.3.2 Maniobrabilidad	6
1.3.3 Controlabilidad.....	6
1.3.4 Área de trabajo	7
1.3.5 Precisión y repetitividad:.....	7
1.4 CLASIFICACIÓN GENERAL	8
1.5 CLASIFICACIÓN ROBOTS “IN PIPE”	9
1.5.1 Tipo Pig	9
1.5.2 Tipo Wheel.....	10
1.5.3 Tipo Caterpillar	11
1.5.4 Tipo Wall press	11
1.5.5 Walking type	12
1.5.6 Inchworm type	13
1.5.7 Screw type	13
1.6 ESTADO DEL ARTE	14

CAPÍTULO II DISEÑO MECÁNICO..... 21

2.1 INTRODUCCIÓN	21
2.2 AUTODESK INVENTOR PROFESIONAL	21
2.3 EJE	22
2.4 FUERZA DE TRACCIÓN	24
2.5 RUEDAS	26
2.6 LINKS	29
2.7 ELECTROEROSIÓN DE HILO	29
2.8 PARTES DEL ROBOT	32
2.8.1 Link Simple	32
2.8.2 Link motor	33
2.8.3 Link lateral	35
2.8.4 Link unión triple	36
2.8.5 Acoples servomotores	38
2.8.6 Porta baterías	41
2.8.7 Sujeción PCB	43
2.9 TRATAMIENTO TERMICO (PAVONADO)	45
2.10 ENSAMBLAJE	46

CAPITULO III DISEÑO ELECTRÓNICO..... 48

3.1 INTRODUCCIÓN	48
3.2 ELEMENTOS A UTILIZAR:	48
3.2.1 Cámara IP	48
3.2.2 Conexión inalámbrica	49
3.2.3 Arduino	50
3.2.4 Shield Bluetooth	53
3.2.5 Motores	56
3.2.6 Driver	64
3.2.7 Baterías	68
3.2.8 Step down converter	70
3.2.9 Control DualShock3	78

3.2.10 Control “MOTION JOY”	79
3.3 CONFIGURACIÓN Y PROGRAMACION	80
3.3.1 Configuración cámara IP	80
3.3.2 Desarrollo y programación Arduino	84
3.3.3 Configuración de conexión Control DualShock3 – Computador	89
3.4 DISEÑO DE PCB	93
3.4.1 Altium Designer	93
3.4.2 Elaboración del circuito	95
CAPÍTULO IV DISEÑO DE LA INTERFAZ DE USUARIO	101
4.1 INTRODUCCIÓN	101
4.2 LABVIEW	101
4.2.1 Paneles	102
4.2.2 Controles	102
4.2.3 Datos	103
4.2.4 Estructuras	103
4.2.5 Funciones	104
4.3 COMUNICACIÓN CÁMARA IP	105
4.4 SUBVI ADQUISICIÓN DE DATOS DUALSHOCK 3	108
4.5 PROCESAMIENTO DE DATOS	111
4.5.1 Sentido de avance.....	111
4.5.2 Velocidad de avance	114
4.5.3 Envío de trama de datos a Arduino	117
CAPÍTULO V PRUEBAS DE FUNCIONAMIENTO, DETECCIÓN Y CORRECCIÓN DE ERRORES.	118
5.1 PRUEBA DE INGRESO AL TUBO	118
5.2 PRUEBA DE ÁNGULO MÁXIMO DE OPERACIÓN.....	119
CONCLUSIONES.....	122
BIBLIOGRAFÍA.....	124
ANEXOS	128

ÍNDICE DE FIGURAS

FIGURA 1 GRÁFICA DE GRADOS DE LIBERTAD FUENTE: (IONESCU, 2014)	6
FIGURA 2 ROBOT TIPO PIG. FUENTE: (TUV, 2013)	10
FIGURA 3 ROBOT CON RUEDAS PARA INSPECCIÓN. FUENTE: (SCHRODER, 2013).....	10
FIGURA 4 ROBOT TIPO CATERPILLAR. FUENTE: (INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS)	11
FIGURA 5 ROBOT WALL PRESS. FUENTE: (INSPECTAHIRE, 2014)	12
FIGURA 6 ROBOT DEL TIPO WALKING (HEXÁPODO)	12
FIGURA 7 ROBOT INCHWORM. FUENTE: (MEDICAL AND ELECTROMECHANICAL DESIGN LAB, 2007)	13
FIGURA 8 ROBOT TIPO SCREW. FUENTE: (INSPECTAHIRE, 2014)	14
FIGURA 9 FABRICACIÓN POR AÑO DE ROBOTS (INTERNATIONAL FEDERATION OF ROBOTICS, 2014)	15
FIGURA 10 PROTOTIPO SPRING FUENTE: (T. NISHIHARA, K. OSUKA, & I. TAMURA, 2010)	16
FIGURA 11 CONFIGURACIÓN TRICICLO	19
FIGURA 12 PROGRAMA AUTODESK INVENTOR PARA MODELADO DE PIEZAS	22
FIGURA 13 DIFERENTES VISTAS DE ÁREA DE TRABAJO DEL EJE	23
FIGURA 14 EJE RÍGIDO MODIFICADO (DESARMADO)	23
FIGURA 15 TRANSMISIÓN DE FUERZAS A LAS RUEDAS VERSIÓN 1	24
FIGURA 16 TRASMISIÓN DE FUERZAS A LAS RUEDAS VERSIÓN 1	25
FIGURA 17 TRANSMISIÓN DE FUERZAS A LAS RUEDAS VERSIÓN 2	26
FIGURA 18 PRIMER FRESADO CON BROCA DE 3MM	27
FIGURA 19 ETAPAS DE FRESADO	27
FIGURA 20 RUEDA FINAL (141 GR.)	28
FIGURA 21 RUEDA CON ADAPTADOR	28
FIGURA 22 RUEDA CON ADAPTADOR FUENTE: (INTEREMPRESAS, 2012)	30
FIGURA 23 INICIO DE CORTE DE LINKS	30
FIGURA 24 INTERFAZ GRÁFICA DONDE SE APRECIA EL AVANCE DEL CORTE	31
FIGURA 25 LINKS LATERAL Y UNIÓN TRIPLE EN EL PRIMER CORTE	31
FIGURA 26 RANGO DE MOVIMIENTO LINK SIMPLE	32

FIGURA 27 LINK SIMPLE FINALIZADO 33

FIGURA 28 RANGO DE MOVIMIENTO LINK MOTOR 34

FIGURA 29 LINK MOTOR FINALIZADO 34

FIGURA 30 RANGO DE MOVIMIENTO LINK LATERAL 35

FIGURA 31 LINK LATERAL PRIMERA ETAPA (IZQ.) Y FINALIZADO 36

FIGURA 32 RANGO DE MOVIMIENTO LINK TRIPLE 37

FIGURA 33 LINK TRIPLE PRIMERA ETAPA (IZQ.) Y FINALIZADO 37

FIGURA 34 ELEMENTOS PARA ACOPLAR LA CÁMARA EL ROBOT (SIN DOBLAR) 38

FIGURA 35 ELEMENTOS PARA ACOPLAR LA CÁMARA EL ROBOT (DOBLADOS Y EN
 POSICIÓN) 39

FIGURA 36 ROTACIÓN HORIZONTAL DEL CONJUNTO 40

FIGURA 37 ROTACIÓN VERTICAL DEL CONJUNTO 41

FIGURA 38 PORTA BATERÍAS ANTES Y DESPUÉS DE DOBLAR 42

FIGURA 39 SUJECIÓN DE LA BATERÍA CON BRIDAS PLÁSTICAS 43

FIGURA 40 PORTA PCB 44

FIGURA 41 SUJECIÓN PORTA PCB AL EJE DEL ROBOT 44

FIGURA 42 DIFERENCIA ENTRE TORNILLO EN POSESO DE OXIDACIÓN Y TORNILLO
 PAVONADO FUENTE: (ZONA BOXES, S.F.) 45

FIGURA 43 ROBOT ENSAMBLADO 46

FIGURA 44 CÁMARA IP EASYN FS-613B-M166 WIRELESS. (EASYNUSA, 2014)..... 49

FIGURA 0.45 PUNTO DE ACCESO INALÁMBRICO LINKSYS WRT54G2 WIRELESS-G
 ROUTER FUENTE: (LINKSYS, 2014)..... 49

FIGURA 46 LOGO DE ARDUINO FUENTE: (WIKIMEDIA COMMONS, 2013)..... 50

FIGURA 47 PLACA ARDUINO MEGA 2560 FUENTE: (ARDUINO, S.F.)..... 52

FIGURA 48 SHIELD BLUETOOTH PARA ARDUINO 55

FIGURA 49 ESQUEMA SHIELD BLUETOOTH PARA ARDUINO 55

FIGURA 50 TIPOS DE MOTORES DE PASOS DE IMÁN PERMANENTE..... 59

FIGURA 51 MOTOR POLOLU NEMA 17 64

FIGURA 52. DRIVER DRV8834 PARA MOTORES DE PASOS. FUENTE: (POLOLU
 ROBOTICS & ELECTRONICS, 2014)..... 66

FIGURA 53 ESQUEMA DE CONEXIÓN DE DRIVER..... 68

FIGURA 54 BATERÍA RECARGABLE DE NIMH TENERGY DE 5 000 MAH. 70

FIGURA 55 CIRCUITO DIVISOR DE VOLTAJE 71

FIGURA 56 CONFIGURACIÓN INTERNA LM2576 FUENTE: (TEXAS INSTRUMENTS, 2013)	73
.....	73
FIGURA 57 APLICACIÓN TÍPICA DE LM2576T-ADJ	74
FIGURA 58 CONEXIONES DE PRUEBA PARA LM2576T-ADJ.....	75
FIGURA 59 STEP DOWN CONVERTER FUENTE: (AMAZON, 2014).....	78
FIGURA 60 CONTROL DUAL SHOCK3 FUENTE: (PLAYSTATION)	79
FIGURA 61 SELECCIÓN DE CÁMARA IP PARA CONFIGURACIÓN.....	81
FIGURA 62 CONFIGURACIÓN DE IP PARA CÁMARA.....	82
FIGURA 63 CONFIGURACIÓN DE LAN INALÁMBRICA.....	83
FIGURA 64 CIRCUITO PRINCIPAL DE CÁMARA IP, ANTENA, LENTE.....	84
FIGURA 65 ENTORNO DE DESARROLLO ARDUINO (IDE)	86
FIGURA 66 PÁGINA DE DESCARGAS PARA MOTIONJOY.....	89
FIGURA 67 PANTALLA PRINCIPAL DE CONFIGURACIÓN DE CONTROL DUAL SHOCK3 ..	90
FIGURA 68 SELECCIÓN DE CONTROL A CONFIGURAR	90
FIGURA 69 INSTALACIÓN DE DRIVERS PARA CONTROL MOTIONJOY	91
FIGURA 70 SELECCIÓN DE CONTROL A CONFIGURAR.....	92
FIGURA 71 CONFIGURACIÓN DE CONTROL.....	92
FIGURA 72 PARÁMETROS DE CONFIGURACIÓN DE CONTROL DUALSHOCK3	93
FIGURA 73 INTERFAZ ALTIUM DESIGNER	95
FIGURA 74 MENÚ PARA CREACIÓN DE LIBRERÍA.....	96
FIGURA 75 MENÚ PARA CREACIÓN DE COMPONENTE	96
FIGURA 76 ASISTENTE PARA CREACIÓN DE COMPONENTES	97
FIGURA 77 LIBRERÍA DE DRIVER DRV8834.....	97
FIGURA 78 MENÚ PARA CREACIÓN DE LIBRERÍA ESQUEMÁTICA.....	98
FIGURA 79 CREACIÓN DE COMPONENTE EN LIBRERÍA ESQUEMÁTICA.....	98
FIGURA 80 VINCULACIÓN DE LIBRERÍA ESQUEMÁTICA CON FOOTPRINT.....	99
FIGURA 81 CAPAS PCB.....	100
FIGURA 82 OPCIÓN RUN DE UN VI.....	102
FIGURA 83 OPCIÓN RUN CONTINUOUSLY DE UN VI.....	103
FIGURA 84 OPCIÓN STOP DE UN VI	103
FIGURA 85 OPCIÓN PAUSE DE UN VI.....	103
FIGURA 86 CONFIGURACIÓN URL DE CÁMARA IP	105
FIGURA 87 DIAGRAMA DE BLOQUES COMUNICACIÓN CÁMARA IP.....	105

FIGURA 88 RECONOCIMIENTO DE BOTONES	106
FIGURA 89 CASO 0, DEFAULT	107
FIGURA 90 CASO 1, RELOAD	107
FIGURA 91 CASO 2, START	108
FIGURA 92 SUBVI CONEXIÓN CONTROL	109
FIGURA 93 MENÚ DE CONECTIVIDAD	109
FIGURA 94 OBTENCIÓN DE DATOS DE CONTROL DUALSHOCK	110
FIGURA 95 SUBVI DE ADQUISICIÓN DE DATOS	110
FIGURA 96 DIAGRAMA DE SENTIDO AVANCE.....	111
FIGURA 97 CASO 0 SENTIDO DE AVANCE.....	112
FIGURA 98 CASO 1 SENTIDO DE AVANCE.....	113
FIGURA 99 CASO 2 SENTIDO DE AVANCE.....	113
FIGURA 100 CASO 3 SENTIDO DE AVANCE.....	114
FIGURA 101 CONTROL DE VELOCIDAD	115
FIGURA 102 FÓRMULA DE CORRECCIÓN DE RANGO SERVOMOTOR	116
FIGURA 103 LIMITACIÓN DE MOVIMIENTO DE LA CÁMARA	116
FIGURA 104 FORMACIÓN TRAMA DE DATOS	117
FIGURA 105 ROBOT DENTRO DE LA TUBERÍA	118
FIGURA 106 MEDICIÓN DEL ÁNGULO DE ELEVACIÓN	119

ÍNDICE DE TABLAS

TABLA 1 COMPARACIÓN DE TIPOS DE ROBOTS IN-PIPE, VENTAJAS Y DESVENTAJAS ...	17
TABLA 2 PUERTOS SERIAL EN ARDUINO MEGA	52
TABLA 3 COMPARACIÓN DE CLASES DE BLUETOOTH	54
TABLA 4 PUERTOS DE SHIELD BLUETOOTH PARA ARDUINO	56
TABLA 5 TABLA COMPARATIVA DE MOTORES COMERCIALES	58
TABLA 6 SECUENCIA PARA CONTROL DE GIRO DE MOTOR BIPOLAR FUENTE: (ISLAS, 2007)	60
TABLA 7 SECUENCIA PARA CONTROL DE GIRO DE MOTOR UNIPOLAR DE 2 BOBINAS. FUENTE: (ISLAS, 2007)	61
TABLA 8 SECUENCIA PARA CONTROL DE GIRO DE MOTOR UNIPOLAR TIPO WAVE DRIVE. FUENTE: (ISLAS, 2007)	62
TABLA 9 SECUENCIA PARA CONTROL DE MOTOR DE PASOS TIPO "MEDIO PASO" FUENTE: (ISLAS, 2007)	63
TABLA 10 COMPARACIÓN DE DRIVERS PARA MOTOR DE PASOS.....	65
TABLA 11 CONFIGURACIÓN DE PASO DE UN MOTOR DE PASOS POR ESTADOS	67
TABLA 12 CONSUMO DE ENERGÍA DE LOS DISPOSITIVOS	69
TABLA 13 COMPARATIVO ENTRE ARDUINO Y PROCESSING.....	85
TABLA 14 POSIBLES COMBINACIONES DE CONTROL DE AVANCE	111
TABLA 15 ÁNGULOS MÁXIMOS OPERACIÓN	120
TABLA 16 COMPARACIÓN ÁNGULOS MÁXIMOS DE OPERACIÓN	121

ÍNDICE DE ANEXOS

ANEXO 1 - LINK SIMPLE	129
ANEXO 2 - LINK MOTOR	130
ANEXO 3 - LINK UNIÓN TRIPLE.....	131
ANEXO 4 - LINK LATERAL	132
ANEXO 5 - EJE EXTREMO.....	133
ANEXO 6 - UNIÓN EJES	134
ANEXO 7 - EJE EXTREMO V2ANEXO 8 - UNIÓN EJES RECTO.....	135
ANEXO 9 - RUEDA	137
ANEXO 10 - MOTOR NEMA 17.....	138
ANEXO 11 - ADAPTADOR RUEDA - MOTOR NEMA 17.....	139
ANEXO 12 - PORTA SERVOMOTOR 1	140
ANEXO 13 - PORTA SERVOMOTOR 2	141
ANEXO 14 - UNIÓN SERVOMOTOR 1	142
ANEXO 15 - UNIÓN SERVOMOTOR 2	143
ANEXO 16 - SERVO MOTOR HS-645MG 1	144
ANEXO 17 - SERVO MOTOR HS-645MG 2	145
ANEXO 18 - ENSAMBLE RUEDAS	146
ANEXO 19 - ENSAMBLE RUEDAS MOTOR	147
ANEXO 20 - ENSAMBLE EJE	148
ANEXO 21 - ENSAMBLE EJE V2	149
ANEXO 22 - ENSAMBLE CUERPO	150
ANEXO 23 - ENSAMBLE PORTA CÁMARA	151
ANEXO 24 - DRIVER MOTOR.....	152
ANEXO 25 - PROGRAMACIÓN ARDUINO	153

**“ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE
LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN
INTERFACE GRÁFICO REMOTO”**

RESUMEN

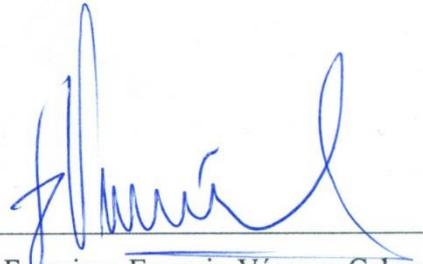
Este trabajo de graduación tiene como objetivo el desarrollo de un robot capaz de inspeccionar el interior de tuberías de diámetro entre 9 y 11 pulgadas, por medio de la captura de imágenes en tiempo real. El intercambio de información entre el computador y el robot se realiza por medio de comunicación Bluetooth, la captura de imágenes se realiza utilizando una cámara IP, la cual transmite en tiempo real el entorno captado dentro del ducto, permitiendo así un generar un plan de reparación y mejora de forma preventiva o urgente, sin necesidad de arriesgar la integridad de trabajadores o incurrir en gastos innecesarios. Para el control y adquisición de datos se utiliza una tarjeta integrada con un microcontrolador, capaz de obtener mediciones de las baterías, y manipular motores. La alimentación del sistema se realiza por medio de baterías de alta capacidad para lograr que el robot funcione de manera inalámbrica.

Palabras Clave: Robot, Arduino, LabVIEW, ductos, explorador, *stepdown*, *in-pipe*.



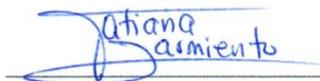
Ing. Omar Santiago Alvarado Cando

Director de Tesis



Ing. Francisco Eugenio Vásquez Calero

Director de Escuela



Tatiana Andrea Sarmiento Astudillo



Miguel Santiago Díaz Andrade

Autores

"STUDY AND DESIGN OF A PIPELINE EXPLORER ROBOT BY REAL TIME
IMAGE CAPTURE USING A REMOTE GRAPHIC INTERFACE "

ABSTRACT

This graduation work aims to develop a robot capable of inspecting the inside of 9 and 11 inches diameter pipes, through real time image capturing. The exchange of information between the computer and the robot is done through Bluetooth communication. The image capture is performed using an IP camera which transmits in real-time the environment capture within the duct; making it possible to generate a preventive or urgent repair and improvement plan without risking the integrity of workers or unnecessary expenses. For control and data acquisition, we used a microcontroller integrated card, able to obtain measurements of the batteries, and manipulate engines. The system power is performed by high-capacity batteries to make the robot work wirelessly.

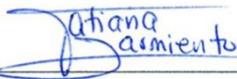
Keywords: Robot, Arduino, LabVIEW, pipelines, explorer, stepdown, in-pipe.



Ing. Omar Santiago Alvarado Cando
Thesis Director



Ing. Francisco Eugenio Vásquez Calero
School Director



Tatiana Andrea Sarmiento Astudillo
Author



Miguel Santiago Díaz Andrade
Author



Translated by,
Lic. Lourdes Crespo

Miguel Santiago Díaz Andrade

Tatiana Andrea Sarmiento Astudillo

Trabajo de Graduación

Ing. Omar Santiago Alvarado Cando.

Mayo, 2015

**ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS
MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL
UTILIZANDO UN INTERFACE GRÁFICO REMOTO**

INTRODUCCIÓN

Es verdad que en la actualidad hay algunos trabajos que los humanos prefieren cederlo a los robots, ya sea porque son complicados, peligrosos o físicamente inaccesibles. Estos dos últimos son algunos de los motivos por los cuales se ha decidido realizar este trabajo de grado, ya que la labor de inspección de tuberías es un trabajo peligroso y en ciertos casos inaccesible para una persona.

Se sabe que existen gran variedad de robots, capaces de realizar todo tipo de actividades que puedan facilitar el trabajo a las personas. Este es uno de esos casos, ya que para realizar inspecciones, limpieza, etc. en cualquier tubería, se utilizan los robots comúnmente denominados “*in-pipe*”, término que utilizaremos a lo largo del documento. Los robots “tele operados o tele robots” son aquellos que pueden ser controlados desde cierta distancia, ya sea por seguridad o inaccesibilidad del operario. En este caso es muy importante esta característica para poder garantizar la seguridad del usuario y evitar accidentes.

Para entender de una mejor manera los riesgos a los que está expuesto un trabajador al realizar este tipo de inspecciones se realizará una breve explicación de los diferentes peligros, limitaciones y factores que se pueden encontrar al realizar esta acción.

En la mayoría de las plantas industriales la utilización de tuberías muy importante, incluso en algunas de ellas se considera imprescindible en la producción y desempeño de la misma, debido a que en estas circulan diferentes tipos de materiales como: aceites de diferentes densidades, fluidos químicos, gases, agua, entre otros, que serán utilizados en los diversos procesos de producción.

Estas tuberías pueden sufrir diferentes tipos de daños que comprende desde corrosión por el paso del tiempo o desgaste natural, fracturas ocasionadas por cambios bruscos de temperatura hasta daños provocados por terceros (golpes, accidentes, mala manipulación, instalación incorrecta, etc.), por lo que es de suma importancia realizar inspecciones y mantenimientos a manera preventiva, evitando detener los procesos de producción por extensos intervalos de tiempo o en el peor de los casos indefinidamente. Por otro lado se debe tener en cuenta que al realizar mantenimiento periódico previniendo futuras averías se obtiene un porcentaje de eficiencia mucho más alto, ya que no se producen pérdidas de fluido, presión, contaminación del material transportado etc.; además se debe considerar que cualquiera de estos daños puede representar un peligro potencial tanto para los trabajadores de la planta como para la maquinaria y edificación.

Uno de los factores primordiales en una planta son los tiempos de producción siendo lo óptimo mantener las actividades de manera constante sin interrupciones, por lo que una falla que produzca pausas en el proceso provocaría muchas pérdidas, por lo que todos los componentes deben estar en perfecto estado. Considerando este factor, las inspecciones deben ser lo más exactas y rápidas posibles.

En caso de no contar con robots para inspección, este trabajo sería prácticamente imposible de realizar en ciertos casos, debido a los siguientes factores:

- Tuberías ubicadas en lugares expuestos a temperaturas extremas,
- Transporte de líquidos nocivos para la salud,
- Redes de tuberías extensas,
- Ubicaciones en áreas de difícil acceso y
- Visibilidad limitada o nula;

Además de los puntos expuestos se debe considerar que un trabajador podría cometer errores y obviar detalles debido al cansancio por temperaturas intensas, fatiga por la duración de la labor o por realizar movimientos repetitivos.

En conclusión es necesaria una solución que pueda reducir el tiempo de inspección, aumentando la producción y mejorando la calidad; al igual que garantizar la seguridad de todos los recursos, ya sean materiales o humanos que se encuentren en torno a estas tuberías. Por lo que un robot es la opción más viable y práctica, además de ser mucho más rápida, eficiente y de menor riesgo.

El diseño y construcción de un robot capaz de inspeccionar las tuberías de una planta de producción, venciendo los problemas de temperatura (altas o bajas), visibilidad (en la parte interna de las tuberías), desenvolvimiento en ambientes hostiles (gases o líquidos tóxicos) y rápido tiempo de inspección tienen gran apertura y acogida en la actualidad, por lo que se está investigando y desarrollando mejoras en ellos. Al tener como ventaja la transmisión de video en tiempo real, se puede estimar el daño y su magnitud de manera inmediata, para posteriormente realizar la reparación, limpieza o cambio de la parte afectada.

CAPÍTULO I

MARCO TEÓRICO

1.1 INTRODUCCIÓN

En este capítulo se efectuará un análisis teórico de las necesidades por las cuales este robot debe ser desarrollado y los problemas que es capaz de resolver. De igual manera se realizará un estudio de las características, elementos a utilizar y principio de funcionamiento de los robots denominados “in-pipe”, determinando cuales son las características que se están utilizando actualmente a nivel mundial y cuáles son los diseños más eficientes.

Por medio de este análisis se podrá obtener un conocimiento general de este género de autómatas y poder cumplir con los requerimientos de la industria que los utiliza.

1.2 DEFINICIÓN

“El término robot fue introducido por primera vez por el checo Karel Capek en 1920. La palabra *robota* en checo significa trabajo. Desde entonces este término ha sido aplicado a una gran variedad de dispositivos mecánicos, tales como tele operadores, vehículos submarinos y terrestres autónomos. Prácticamente cualquier cosa que sea operada con un grado de autonomía, usualmente bajo el control de una computadora, hasta cierto punto ha sido llamado robot.” (Mark, 2006)

Una de las definiciones más reconocidas es la adoptada por la Organización Internacional de Estándares ISO, que define al robot industrial como:

“Manipulador multifuncional reprogramable con varios grados de libertad, capaz de mover materias, piezas, herramientas o dispositivos especiales según trayectorias variables, programadas para realizar tareas diversas.” (Mark, 2006)

Ahora que se ha definido un robot es necesario realizar un análisis de las principales características que lo definen, se puede decir que son: Grados de libertad, área de trabajo, repetitividad, maniobrabilidad, controlabilidad y precisión.

A continuación describiremos cada una de ellas y como afecta en el desarrollo y diseño de un robot.

1.3 PRINCIPALES CARACTERÍSTICAS DE UN ROBOT

1.3.1 Grados de libertad

Los grados de libertad o también conocidos como DOF (Degrees of Freedom) es el número independiente de movimientos que puede llegar a tener un actuador, en otras palabras, indican el estado físico de un sistema y es fundamental porque nos permite determinar el alcance que puede tener un robot dentro de su área de trabajo (Mark, 2006). Hacen referencia a los posibles movimientos en el eje X, Y, y Z al igual que la capacidad de rotación.

Existen 6 tipos de grados de libertad separados en 2 grupos:

Traslacional:

- Movimiento arriba-abajo
- Movimiento izquierda-derecha
- Movimiento adelante-atrás

Rotacional:

- Inclinación adelante-atrás (pitching)
- Giro izquierda-derecha (yawing)
- Inclinación lado-lado (rolling)

En la figura 8 se puede ver una referencia de cada uno de los grados de libertad que se mencionan anteriormente.

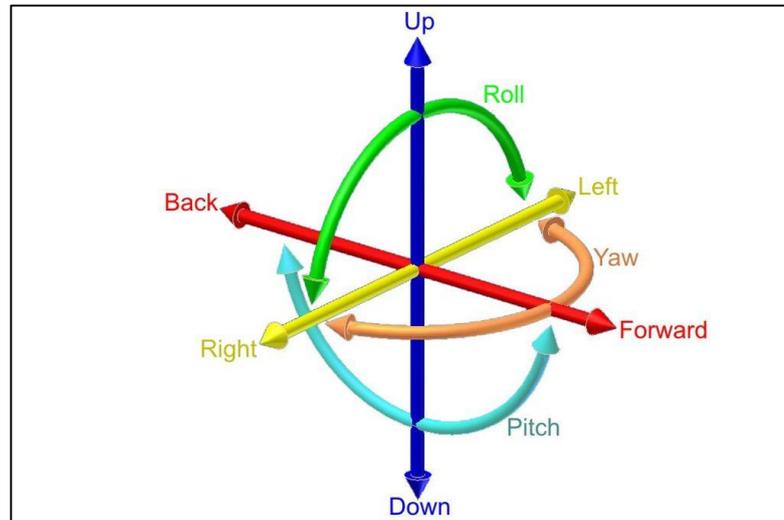


Figura 1 Gráfica de grados de libertad

Fuente: (Ionescu, 2014)

1.3.2 Maniobrabilidad

Es la capacidad de un robot para movilizarse en un plano sin importar la orientación con el eje vertical, esta depende básicamente de orientación de las ruedas. Este parámetro está relacionado con los grados de libertad

1.3.3 Controlabilidad

La controlabilidad tiene una relación inversa con la maniobrabilidad, denota la capacidad de moverse alrededor de un sistema en todo su espacio de configuración utilizando sólo ciertas manipulaciones admisibles. La definición exacta varía ligeramente en el marco o el tipo de modelos aplicada.

1.3.4 Área de trabajo

El área de trabajo está definida por dos parámetros importantes, los elementos del manipulador y los grados de libertad. Esta característica es fundamental para definir el área de acción o incidencia que va a tener el modelo que se decida implementar para realizar una acción determinada. Hay que realizar un análisis adecuado, ya que se debe considerar que ciertas áreas no pueden ser utilizadas, tanto por limitaciones de giro como desplazamientos en las articulaciones.

1.3.5 Precisión y repetitividad:

Otras características que se deben tener en cuenta son la precisión y repetitividad, las cuales se miden de la siguiente manera:

- Precisión.- Que tan exacto puede llegar el manipulador a un punto dado.
- Repetitividad.- Capacidad de alcanzar una misma posición por varias ocasiones.

Para que un robot pueda realizar los movimientos para los cuales fue diseñado, es necesario proveerle de los comúnmente conocidos manipuladores, capaces de proporcionar los DOF.

Los manipuladores están compuestos por:

- Links: Cuerpo rígido, conectado a otro, por medio de joints con el propósito de transmitir la fuerza o movimiento (ZHANG, FINGER, & BEHREN).
- Joints: uniones o articulaciones, encargadas de unir dos links para generar un movimiento.

Estos dos tipos de elementos son los encargados de formar la cadena cinemática, que puede generar movimientos tanto de revolución como prismáticos, a continuación se detalla el principio de cada una de ellas:

- Revolución: Como su nombre lo indica, genera un movimiento de rotación
- Prismática: Genera un desplazamiento lineal.

1.4 CLASIFICACIÓN GENERAL

En la actualidad existen una gran cantidad de clasificaciones para los muchos tipos de autómatas que se han creado, pero la más grande y general, es la que los divide en dos diferentes grupos:

- Robots Fijos
- Robots Móviles

Dentro del primer grupo se pueden considerar a todos los robots que están sujetos a un plano fijo, con respecto al cual se mueven el resto de las partes, para realizar un trabajo determinado.

En el grupo de los robots móviles, se tienen a todos los autómatas capaces de movilizarse libremente dentro de un entorno específico para el cual fueron diseñados, generalmente este es el tipo de robot utilizado para exploración.

Obviamente estas dos grandes categorías se pueden subdividir en varias categorías, según diferentes parámetros, los cuales pueden ser:

El medio en el que se desenvuelven:

- Acuáticos
- Aéreos
- Terrestres
- Espaciales, etc.

El grado de control:

- Autónomo
- Tele operado

Los elementos de movilidad:

- Extremidades
- Ruedas
- Orugas
- Hélices, etc.

De esta forma se pueden extender muchas otras clasificaciones, tomando en cuenta varios parámetros y según las necesidades. Pero por ahora continuaremos con la clasificación específica de los robots in-pipe.

1.5 CLASIFICACIÓN ROBOTS “IN PIPE”

A continuación se detallará la clasificación más común que se puede encontrar en varios papers sobre este tema, utilizaremos una basada en el estudio de Hyouk Ryeol Choi y Se-gon Rohla de la Universidad de Ingeniería mecánica de Sungkyunkwan Korea con el tema “Differential-Drive In-Pipe Robot for Moving Inside Urban Gas Pipelines”, que se realiza de acuerdo al tipo de mecanismo de movimiento dentro de la tubería:

1.5.1 Tipo Pig: El movimiento es realizado utilizando de la presión del fluido dentro de la red de tuberías. Éste se coloca antes de enviar el gas o fluido, y es conducido pasivamente a lo largo de la red. Generalmente es usado para tuberías de diámetros grandes



Figura 2 Robot tipo Pig.

Fuente: (TUV, 2013)

1.5.2 Tipo Wheel: El tipo rueda es utilizado generalmente para realizar inspecciones de tuberías horizontales y con poca inclinación, ya que su característica fundamental es el uso de cuatro o más ruedas a manera de un vehículo, en el cual se monta la cámara y el resto de elementos a utilizar y se controla de una manera similar a un vehículo común. Una de las mayores desventajas es, como se mencionó anteriormente, sólo puede circular por lugares que no tengan inclinación muy pronunciada.



Figura 3 Robot con ruedas para inspección.

Fuente: (Schroder, 2013)

1.5.3 Tipo Caterpillar: También conocido en español como tipo “oruga”, este nombre se debe a que las ruedas son de este tipo, es decir varios eslabones que giran sin fin y provocan el movimiento, brindando más tracción, facilidad de moverse en terrenos desiguales y sobrepasando obstáculos de una mejor manera. Al poseer una mayor área de contacto el peso del vehículo es distribuido a lo largo de ella, aumentando la capacidad en cuanto al peso que puede soportar y las ruedas no se verían afectadas. Una de las mayores desventajas es la complejidad mecánica que posee, y mayor desgaste.

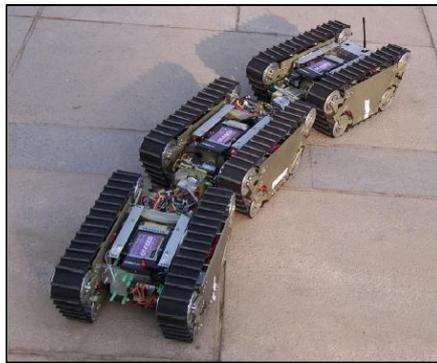


Figura 4 Robot tipo Caterpillar.

Fuente: (International Journal of Advanced Robotic Systems)

1.5.4 Tipo Wall press: Este robot es, como lo indica su nombre, aquel que genera una presión en la pared para movilizarse dentro de la red de tuberías. Puede combinarse con cualquiera de los tipos b, e, c e inclusive con el tipo f. siendo uno de los más eficientes, ya que a más de poder movilizarse por tuberías horizontales, también puede escalar las tuberías con una gran inclinación y aún mejor las totalmente verticales. Por este motivo este es el tipo de robot a desarrollar en esta tesis, además de tener una capacidad adaptativa en relación al diámetro interno del ducto a ser inspeccionado. Más adelante se analizará los rangos de diámetros en los que puede movilizarse de manera adecuada.

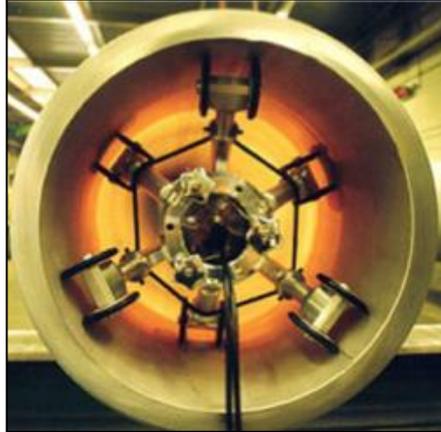


Figura 5 Robot Wall press.

Fuente: (Inspectahire, 2014)

1.5.5 Walking type: En esta clasificación se encuentran los robot cuadrúpedos y hexápodos, aunque los más utilizado son los primeros, los hexápodos presentan muchas ventajas sobre el anterior como son, estabilidad, velocidad y robustez. Pero tienen como desventaja la complicidad en el control y el costo. Este grupo de robots con extremidades, son muy útiles cuando se tienen muchos obstáculos como piedras, protuberancias, o grietas que las ruedas no pueden evitar. Pero al igual como se indicó en clasificaciones anteriores su mayor desventaja, tanto como para cuadrúpedos y hexápodos, es la inclinación en las tuberías, ya que este no tiene la capacidad de escalar tubos empinados o verticales.



Figura 6 Robot del tipo walking (hexápodo)

Fuente: Autor

1.5.6 Inchworm type: Este es un tipo de robot muy peculiar y útil dentro de esta clasificación, una definición que puede ser citada es “Un robot tipo inchworm es aquel que imita el patrón de movimiento de un gusano. Generalmente consiste en módulos interconectados que se deforman en el sentido de movimiento (extensión) o producen fricción contra el medio (ancla).” (CHENG, YEO, & GAO, 2001) Con esto se quiere decir que el movimiento que realiza es de alargamiento y contracción como un gusano que se impulsa para avanzar.

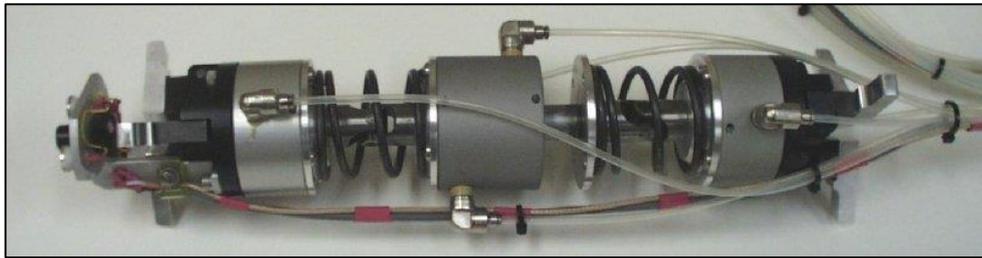


Figura 7 Robot Inchworm.

Fuente: (Medical and electromechanical design lab, 2007)

1.5.7 Screw type: La principal ventaja de este tipo de robots es que pueden ser manipulados con un solo actuador, por lo tanto el mecanismo es mucho más sencillo comparado con otros modelos que utilizan una mayor cantidad de actuadores para realizar los movimientos reduciendo a la vez el consumo de energía. El movimiento se basa en el giro del actuador que va a producir el avance del robot. La principal desventaja es su limitación en la movilidad ya que no puede realizar giros de manera estable, o subir por las tuberías verticales.



Figura 8 Robot tipo Screw.

Fuente: (Inspectahire, 2014)

1.6 ESTADO DEL ARTE

Ahora que se tiene un mayor conocimiento de los diferentes tipos de robots para inspeccionar tuberías que existen en el mercado, se realizará un breve análisis de uno de cada tipo, de los que se han publicado en documentos de investigación durante los últimos 20 años, para poder identificar los pros y contras y poder comprender el diseño que se va a realizar en este trabajo de grado

Lo que se puede percibir de todo este análisis es que el desarrollo de la robótica ha crecido exponencialmente, una de las motivaciones es evitar la intervención humana en procesos que generen riesgos o peligros potenciales.

En los últimos años se ha registrado un gran crecimiento en la fabricación y venta de robots para la industria según indica la IFR (International Federation of Robotics), esto es debido a la gran utilidad y diversas áreas de aplicación, siendo. Esto garantiza un entorno seguro para los trabajadores y reduce costos de producción.

Según la IFR en el último año se generó un 12% de incremento lo que significa 178.132 unidades, siendo las principales áreas de aplicación las industrias automotriz, química, plásticos-cauchos y así como la alimenticia las de mayor crecimiento. Según estos datos, el porcentaje de crecimiento en los últimos 5 años es el más alto registrado desde 1995.

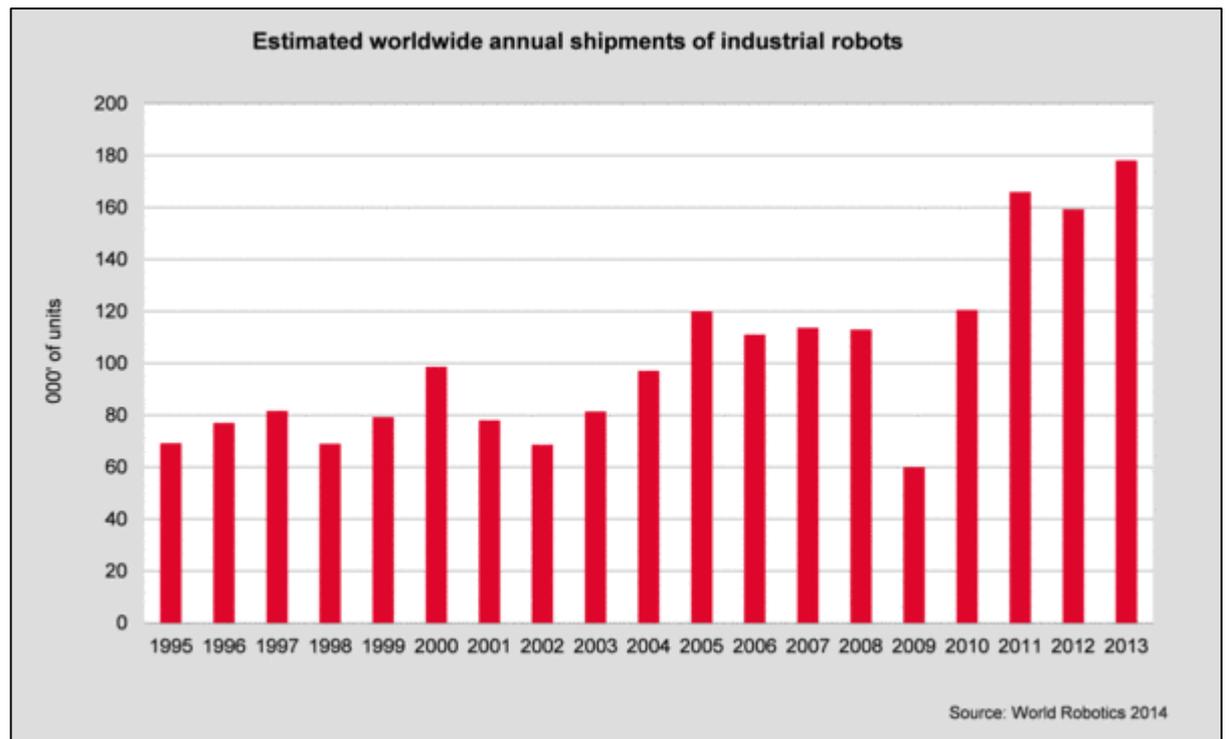


Figura 9 Fabricación por año de robots.

Fuente: (*International Federation of Robotics, 2014*)

Como se ha mencionado en algunas ocasiones el proceso de inspección de tuberías para los humanos resulta muy riesgoso, es por esto que se han desarrollado estos ayudantes para quitar los peligros y mejorar la productividad.

A lo largo del tiempo se han desarrollado muchos prototipos de robots capaces de inspeccionar tuberías, es por esto que se los ha clasificado según su aplicación y movilidad, según lo descrito anteriormente, estos pueden ser tipo: PIG (Pipe Inspection Gauge), Caterpillar, screw, inchworm, de ruedas, Wall-press y walking (CHOI & ROH, 2007) (RYEW, RYU, & ROH, 2000).

Ahora haremos un breve análisis y descripción de los mecanismos, método de inspección, control y manejo, de los modelos existentes. Tomaremos como ejemplo uno de cada tipo para explicar los puntos antes expuestos, entonces comenzaremos por los robots tipo:

- **Screw:** o su traducción al español “tipo tornillo” toman este nombre debido al movimiento que utilizan para moverse dentro de la tubería. Un claro ejemplo de este es el conocido como SPRING (T, K, & TAMURA, 2010), usado para la inspección dentro de tuberías de gas.



Figura 10 Prototipo SPRING

Fuente: (T, K, & TAMURA, 2010)

Este prototipo presentó varios inconvenientes inicialmente, uno de ellos y el más importante resultó ser la movilidad dentro de la tubería, el robot se atascaba al llegar a una válvula o las llantas se resbalaban al llegar a un codo, todo esto debido a la forma del robot. Para solucionar estos inconvenientes se propone rediseñar el cuerpo de SPRING.

- **Walking:** Unos de los robots más reconocidos dentro de este tipo, es el desarrollado por la Universidad de Munich conocido como “MORITZ” (ZAGLER & PFEIFLER, 2003). Este diseño permite escalar tuberías de cualquier inclinación y uniones de cualquier tipo. Este robot cuenta con 8 patas, estas obtienen su movimiento con un motor DC. El movimiento está basado en la fricción y presión que generan las patas contra el tubo para sostenerse en el

lugar, 4 de ellas permanecen estables mientras las otras 4 adelantan o retroceden, se repiten los movimientos y es así como avanza o retrocede.

- **Wall-press:** Esta categoría es una de las más amplias y desarrolladas, ya que combinan muchas de las cualidades de las demás categorías, también abarcan tuberías desde 10 hasta 50 cm. Tienen varios tipos de mecanismos, en los que se utilizan muelles, motores, válvulas, etc. para generar la presión contra las paredes de la tubería.
- **Wheeled:** Estos tipos de robots están aún en proceso de desarrollo ya que son usados generalmente en tuberías de diámetros pequeños. Hay prototipos de micro robots como el de K. Suzumori que permiten inspeccionar ductos de aproximadamente 1 pulgada.

Luego de haber revisado varios prototipos en papers podemos hacer la siguiente tabla comparativa:

Tabla 1 Comparación de tipos de robots in-pipe, Ventajas y Desventajas

Tipo	Principio de movimiento	Ventajas	Desventajas
Inchworm	El movimiento es inspirado en el de las lombrices, primero avanza la parte delantera, recoge el cuerpo y avanza la parte posterior	<ul style="list-style-type: none"> • Permite inspeccionar tuberías verticales con mucha facilidad. • Adaptable a varios diámetros 	<ul style="list-style-type: none"> • Movimientos limitados, no puede circular por codos o cualquier ángulo en tuberías
Wall Press	Generan una fuerza hacia afuera para presionar los actuadores contra las paredes de las tuberías y poder	<ul style="list-style-type: none"> • Gran adaptabilidad a varios diámetros • Capacidad de inspeccionar tanto tuberías horizontales 	<ul style="list-style-type: none"> • La fricción generada puede dañar el interior de la tubería • Propensión a perder tracción

	moverse dentro de ellas	<p>como con gran inclinación</p> <ul style="list-style-type: none"> • Mecanismo sencillo • Evita atascos debido a su estructura • Consumo eficiente de energía 	
Pig	Utilizan la presión del gas o líquido generado dentro de la tubería para poder movilizarse	<ul style="list-style-type: none"> • Utiliza los mismos recursos que circulan por la tubería para su movimiento 	<ul style="list-style-type: none"> • Diseño de estructura muy compleja • Poca maniobrabilidad
Wheel	Utilizan el movimiento de las ruedas para moverse dentro de la tubería	<ul style="list-style-type: none"> • Movimientos rápidos • Mecanismo muy simple • Bajo costo 	<ul style="list-style-type: none"> • Propensión a perder tracción
Walking	Tiene extremidades que se apoyan en las paredes de la tubería, se intercalan los movimientos de las patas para evitar desequilibrios	<ul style="list-style-type: none"> • No tiene pérdida de tracción • No causa daños al interior de la tubería • Capaz de inspeccionar tuberías verticales 	<ul style="list-style-type: none"> • Puede atascarse si coincide con un agujero en la tubería • Mecanismo y estructuras complejas
Screw	Gira por medio de motores igual que el movimiento de un tornillo, generando el avance del cuerpo	<ul style="list-style-type: none"> • No daña las paredes internas del tubo debido a su movimiento • Puede cambiar de diámetro de inspección fácilmente 	<ul style="list-style-type: none"> • Propensión a atascos en la tubería debido a la estructura • Mecanismo muy complejo

Fuente: Autor

También es importante analizar los desarrollos existentes en nuestro país en el tema de inspección de ductos. Luego de una búsqueda con temas relacionados a la inspección de ductos y tuberías por medio de robots, hemos encontrado solamente el desarrollo en investigación denominado “DISEÑO Y CONSTRUCCIÓN DE UN ROBOT MÓVIL TELEOPERADO PARA LA LOCALIZACIÓN DE OBSTÁCULOS EN TUBERÍAS” (ESPINOZA VELASQUEZ, [200-]), desarrollado por Magali Elizabeth Espinoza Velásquez previo a la obtención del título de Ingeniera Electrónica en la Universidad Politécnica del Ejército.

Este prototipo consiste en un robot tipo Wheeled, con capacidad de transmitir imágenes en tiempo real. El principal objetivo es inspeccionar tuberías que transportan productos que intervienen en la industrialización del petróleo, reduciendo costos y peligros para los trabajadores que intervienen en los diferentes procesos productivos. Este robot ha sido diseñado para tuberías de 10 pulgadas, con una configuración triciclo (como se observa en la figura a continuación) y con tracción por medio de motores DC.

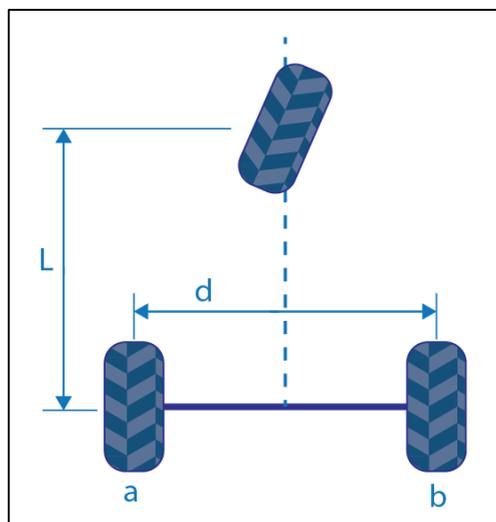


Figura 11 Configuración Triciclo

Fuente: Autor

Es operado por medio de radio frecuencia y controlado con un PIC 16F877A. Utiliza una cámara Xcam2 Cámara Color Rf Exterior S130331.

Con todo este análisis se puede decir que:

- Si se requiere fabricar un robot para tuberías de diámetros pequeños a medianos, es decir de 4 a 9 in es recomendable utilizar robots tipo inchworm, wheeled o screw
- Para tuberías de gran tamaño, de 9 in en adelante, es recomendable utilizar las estructuras de robots tipo Walking o Wall press.

Por todas las ventajas y desventajas antes mencionadas, en esta investigación y trabajo de grado, el tipo de robot a fabricar será “Wall press” con tracción por medio de ruedas. La principal ventaja del uso de ruedas, es el equilibrio y adaptabilidad que entregan.

Con todo esto en mente se debe desarrollar un prototipo que cumpla con las características antes mencionadas en cuanto a la estructura, control, adquisición de imágenes en tiempo real, eficiencia en el consumo de energía y capacidad para subir tuberías con la mayor inclinación posible.

CAPÍTULO II

DISEÑO MECÁNICO

2.1 INTRODUCCIÓN

En este capítulo se desarrollará el diseño mecánico del robot. Para esto se utilizarán varias herramientas tanto virtuales como físicas para el modelado y maquinado de las piezas. Se analizará los diferentes segmentos que formarán el robot, los materiales, pesos, forma y alcances de la estructura. De igual forma se simulará y analizará si el robot cumple con lo planteado.

2.2 AUTODESK INVENTOR PROFESIONAL

Es una herramienta muy útil para el modelado 3D, permite realizar particulares para posteriormente ensamblarlos. Una de las características que más se destacan es la simulación de los objetos, en donde se puede denominar el material del que está hecho el objeto y obtener cálculos muy precisos de fuerza, peso, volumen etc. al igual que se pueden realizar ensambles para determinar si los movimientos de las piezas podrían llegar a tener algún tipo de interferencia, de esta manera se pueden corregir errores inclusive mucho antes de fabricar los elementos. Otra de sus características importantes está una interfaz amigable con el usuario, que como se mencionó anteriormente permite realizar simulaciones de partes móviles, renders realistas con texturas, cálculos de pesos y volumen de los componentes en total, realización de planos con normas técnicas y vistas según la necesidad.

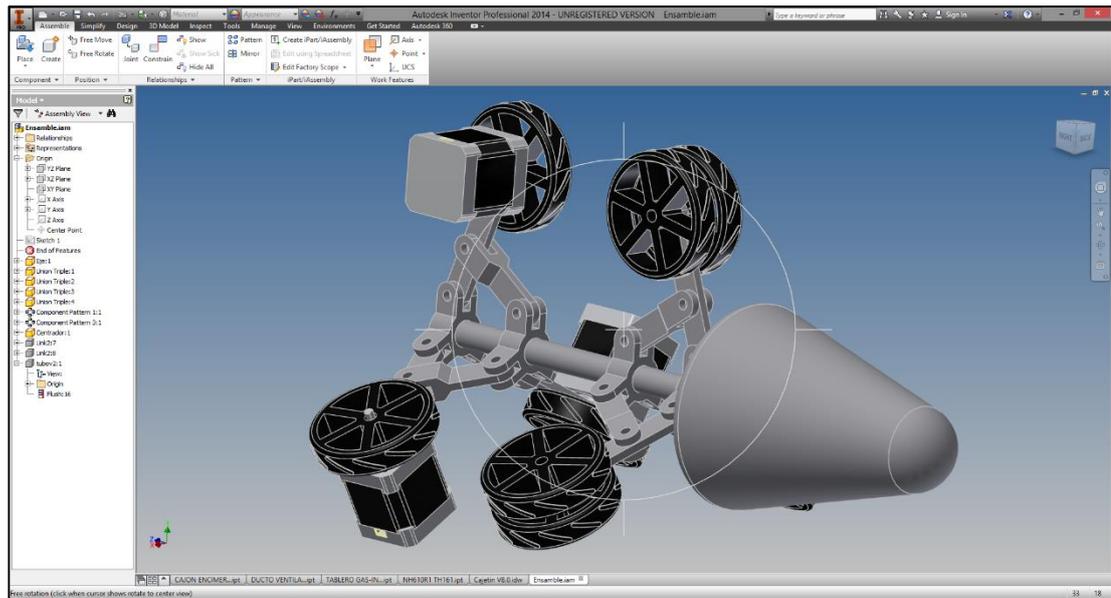


Figura 12 Programa Autodesk Inventor para modelado de piezas

Fuente: Autor

2.3 EJE

Para poder empezar con el diseño del robot, se debía generar la estructura que lo guiará, para esto se utilizaron herramienta Autodesk Inventor que se mencionó en el punto anterior.

Inicialmente la “columna” se diseñó y construyó como un eje dividido en tres partes, unidos en dos puntos con un grado de libertad rotacional, uno con respecto a X y el otro a Y tomando al eje como eje z. Esto le brindaba al robot la libertad de poder rotar su parte superior con respecto a la inferior en dos sentidos abarcando un área similar a una media esfera.

Las secciones están unidas por tornillos M6 recortados el exceso para que no sobresalgan. Los extremos del eje tiene un diámetro de 15mm mientras el resto tiene un diámetro de 14.9mm, esto es para que las uniones triples se deslicen con una décima de milímetro de juego.

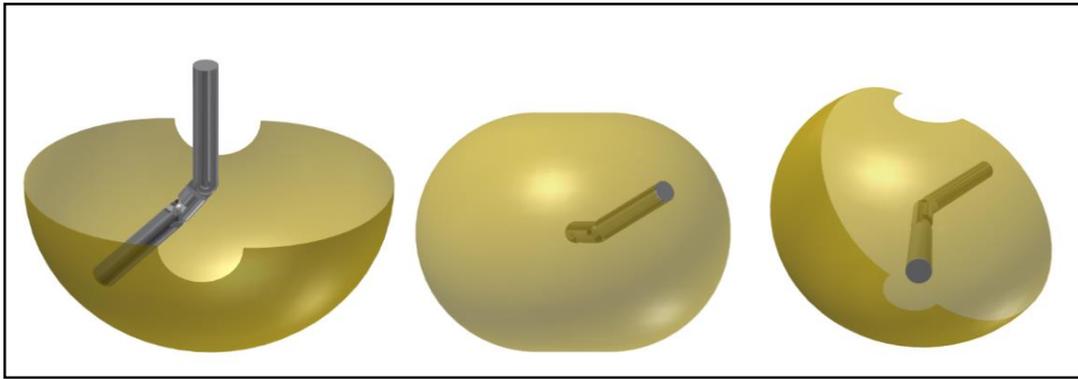


Figura 13 Diferentes vistas de área de trabajo del eje

Fuente: Autor

Cuando se realizaron las primeras pruebas de ingreso a la tubería fue muy difícil que las dos secciones ingresaran ya que la flexibilidad en la parte central impedía la alineación de las ruedas. Requería mucha fuerza y manipulación del robot lo que resultaría imposible de hacer una vez que los componentes electrónicos estén ensamblados por su fragilidad.

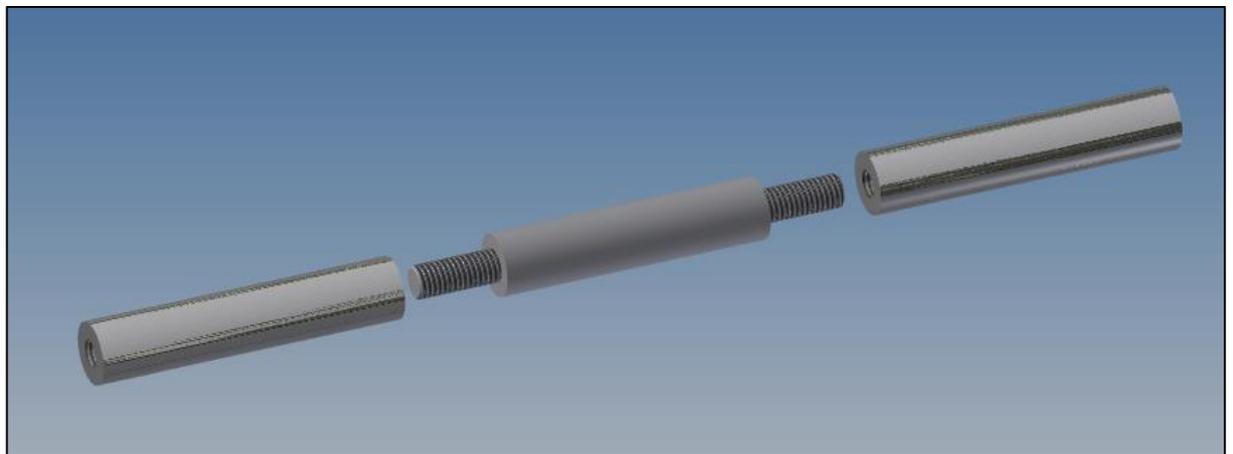


Figura 14 Eje rígido modificado (desarmado)

Fuente: Autor

Esto se solucionó eliminando el link central que permitía la rotación y cortando los extremos donde se acoplaban los tornillos, se fresó un agujero y se rosco para tornillo M8 en ambos ejes, finalmente se agregó un eje de 70mm con dos roscas M8 a cada

extremo y se atornilló a los ejes anteriores. Esto le dio la rigidez necesaria para poder entrar en la tubería sin mayor esfuerzo.

2.4 FUERZA DE TRACCIÓN

Para dar la tracción las ruedas del robot y que éste se presione contra la pared se pensó en varias alternativas como extensiones periscópicas o un motor que empuje las extremidades del robot, pero finalmente se optó por un resorte ya que tiene la ventaja de ser un elemento pasivo y de poco costo.

Inicialmente se construyó un diseño simétrico, donde la parte frontal era igual a la posterior. Un resorte de compresión se colocó en la mitad lo que empujaba dos link deslizantes en el eje al estirarse. Esto hacía rotar las extremidades hacia afuera con respecto a un link estático en cada extremo. Se calculó la longitud de las patas para que cuando estas hagan contacto con las paredes del tubo el resorte aún ejerciera fuerza dando la tracción requerida.

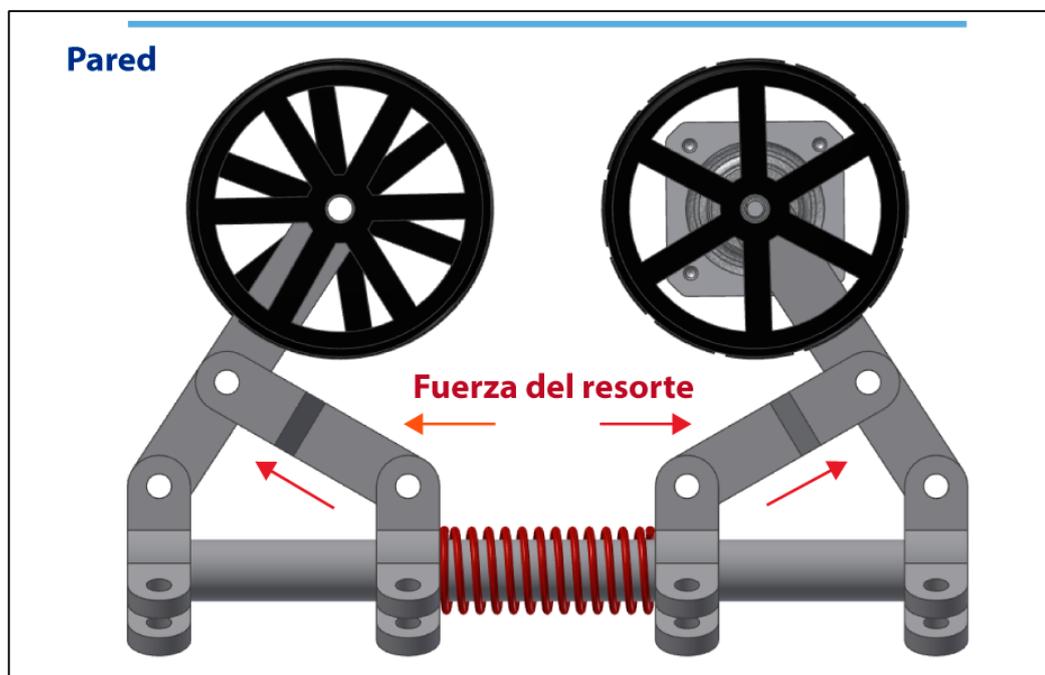


Figura 15 Transmisión de fuerzas a las ruedas versión 1

Fuente: Autor

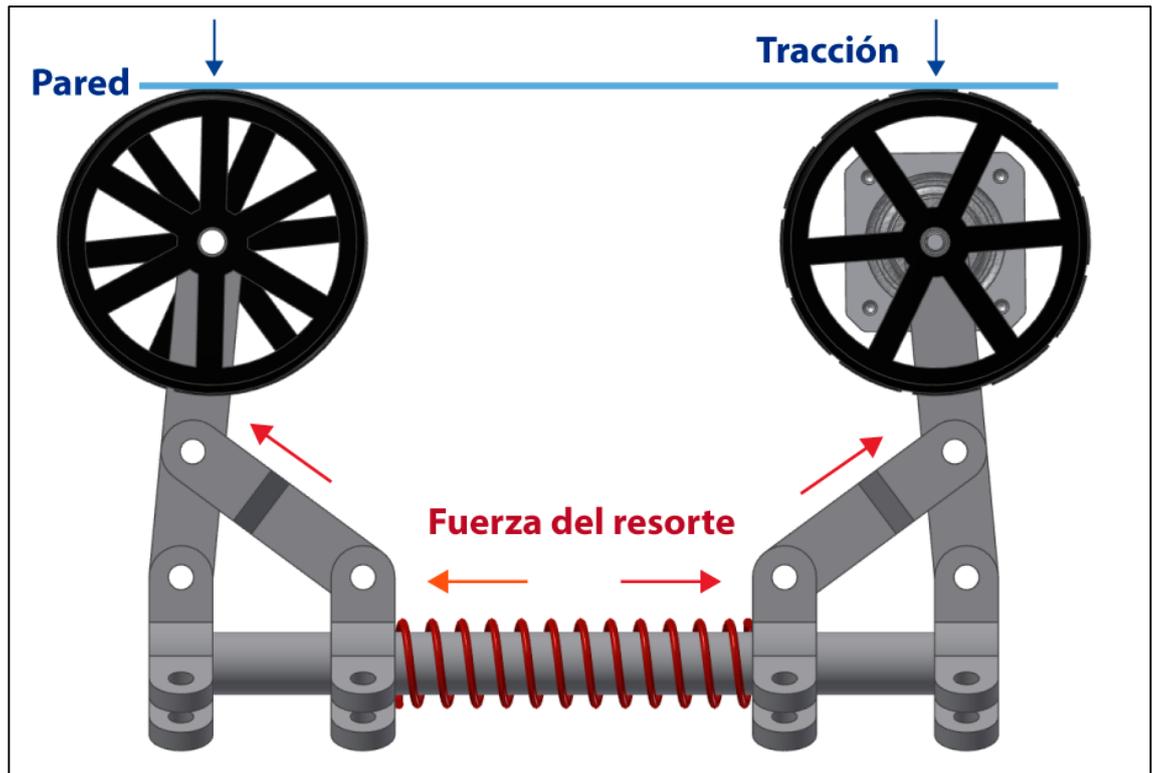


Figura 16 Trasmisión de fuerzas a las ruedas versión 1

Fuente: Autor

Esto funcionó inicialmente al probar la compresión de las extremidades, pero el momento de meter el robot en la tubería ocurrió un gran problema. El sistema de tracción al ser simétrico permitía el fácil ingreso de la parte frontal ya que la ruedas eran empujadas para atrás, pero cuando la sección posterior intentaba entrar resultaba imposible empujar las ruedas hacia adelante porque no había como sujetar al robot para comprimir el resorte permitir el ingreso de estas.

Teniendo este problema como prioridad se modificó el diseño pero intentando utilizar el menor número de partes nuevas y aprovechando las anteriores.

Se solucionó intercambiando el links estático y móvil de la parte posterior y agregando tres resortes de tracción que realizaban el trabajo opuesto al el resorte central. Esto nos dio una copia del funcionamiento de la parte frontal y ahora las ruedas podían comprimirse en la misma dirección lo que permitió el ingreso al tubo sin dificultad y con la tracción requerida.

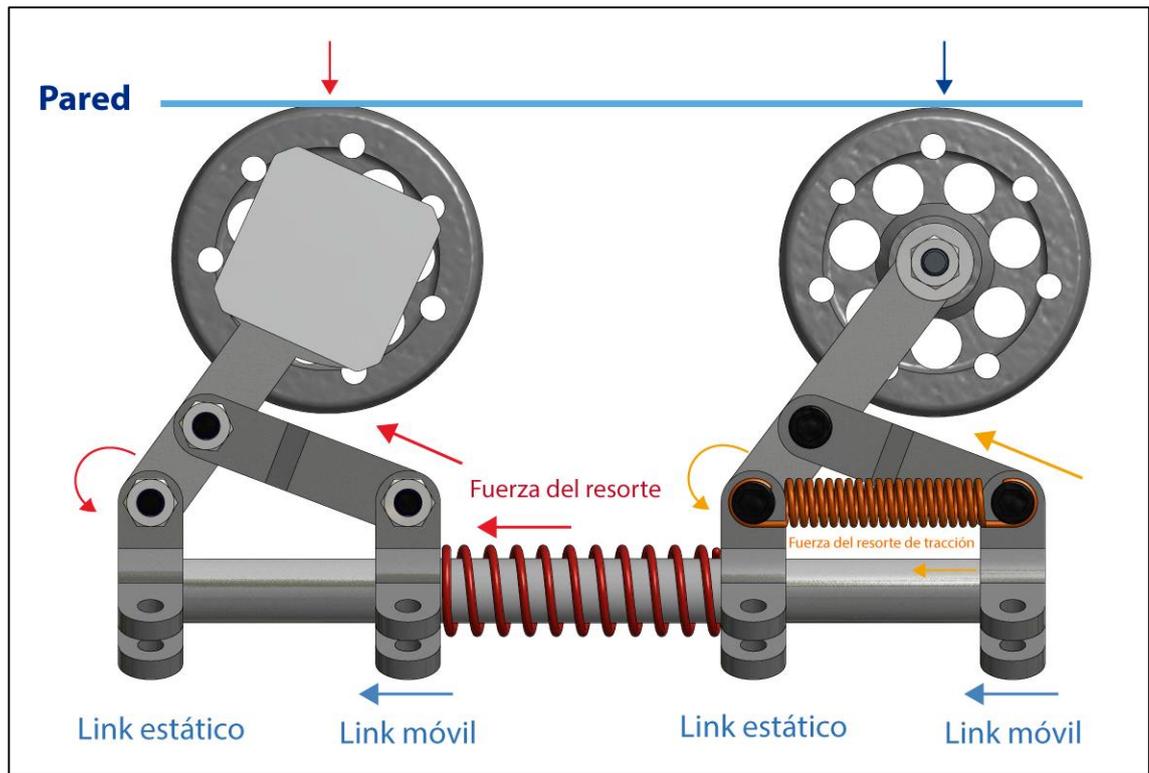


Figura 17 Transmisión de fuerzas a las ruedas versión 2

Fuente: Autor

2.5 RUEDAS

Cuando se tuvo lista la simulación del robot, se buscó las ruedas disponibles en el mercado para acoplarlas al diseño, entonces se decidió las ruedas disponibles en el mercado de 72mm de diámetro, de caucho macizo y pesan 160gr.

Uno de los puntos importantes ha sido mantener el peso lo más bajo posible para evitar un consumo alto en las baterías y esfuerzo excesivo en los motores, para esto se redujo el peso extrayendo material innecesario con una fresadora manual mediante el siguiente procedimiento:

- Dibujo de la ubicación de los agujeros (Anexo 9)
- Marcado de centros con punteador basándose en los planos.
- Primer fresado con broca de 3mm

- Segundo fresado con broca de 8mm
- Tercer fresado con broca de 12mm
- Fresado adicional con broca de 6mm

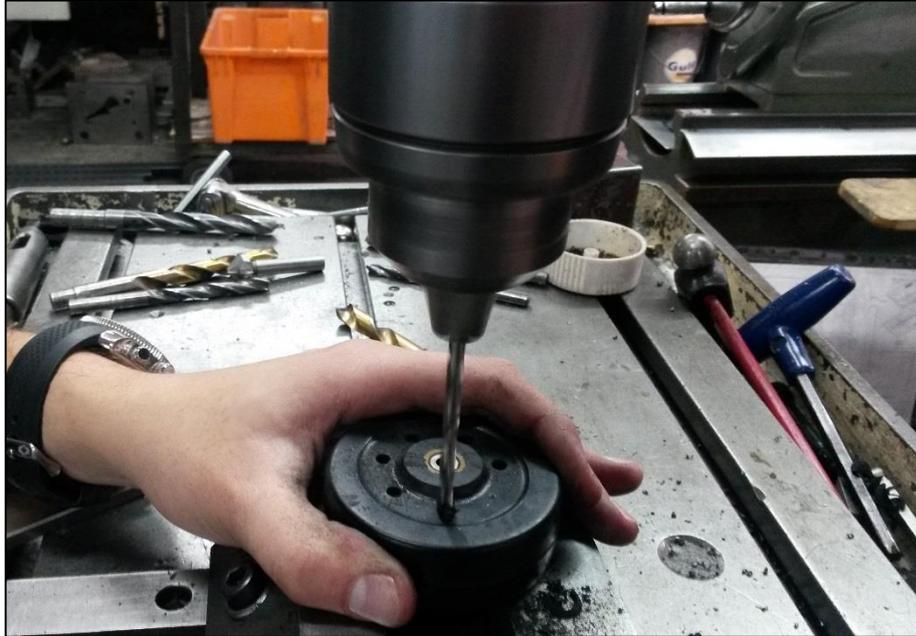


Figura 18 Primer fresado con broca de 3mm

Fuente: Autor



Figura 19 Etapas de fresado

Fuente: Autor

El proceso de fresar las 9 ruedas tomó un tiempo aproximado de 2 horas y redujo 39 gr. a cada rueda, en total 351gr, teniendo como resultado la rueda de la Figura 11.



Figura 20 Rueda final (141 gr.)

Fuente: Autor

Para las 3 ruedas que se acoplan a los motores se adaptó un buje con la forma del eje del motor (Anexo 11) y se insertaron a las ruedas con una prensa neumática.



Figura 21 Rueda con adaptador

Fuente: Autor

Como proceso final, con una lima de diamante se redondeó los bordes de las ruedas que tienen contacto con las paredes del tubo para aumentar el área de contacto y mejorar la tracción

2.6 LINKS

Dentro del campo de la cinemática de robots, los links, o uniones en español, son cuerpos rígidos donde sus uniones proveen movimientos ya sea de traslación o rotación.

A continuación se describirá el proceso de construcción de los links, para esto se usó el proceso de corte por electroerosión de hilo, ya que se encuentra disponible en la fábrica donde se elaboraron las piezas y es de mucho mayor beneficio en razón de costo/tiempo en comparación con una fresadora.

2.7 ELECTROEROSIÓN DE HILO

Cuando se terminó el modelado, era necesario realizar el corte y maquinado de las piezas, por lo que mediante varias visitas a la empresa INGMATRICOM, se determinó que el proceso más conveniente por la relación de tiempo, costo, producto final era utilizar la máquina de electroerosión de hilo.

Se pudo observar el funcionamiento de la máquina que consiste en un hilo conductor de 0.2mm de diámetro por el que circula corriente y genera un arco eléctrico al contacto con una pieza metálica. Esto calienta la pieza y pulveriza macropartículas de metal. Se utiliza un flujo continuo de líquido dieléctrico, que por lo general es agua des ionizada, para arrastrar las partículas de metal erosionado. La mesa de trabajo se mueve en los ejes X & Y, el hilo es rebobinado continuamente en el eje Z. Esta máquina nos permite realizar cortes en planchas metálicas con excelente precisión y dejando ángulos que serían muy complicados o imposibles con una fresadora. La desventaja es cuando requerimos cortar en el interior de una pieza porque se debe cortar el hilo, pasarlo por un agujero previo y amarrar el hilo, esto toma tiempo y material que no puede ser usado nuevamente.

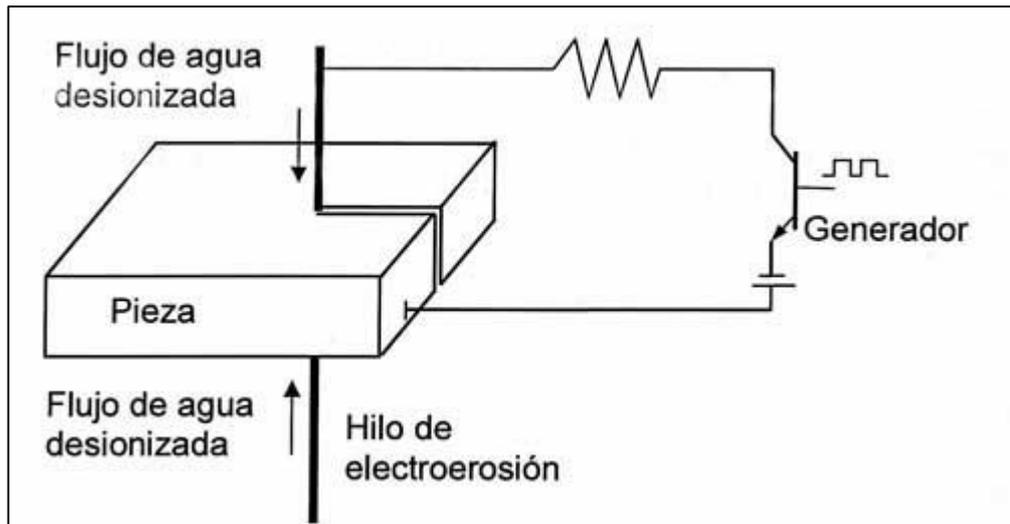


Figura 22 Rueda con adaptador

Fuente: (Interempresas, 2012)

Para la construcción de los link se usó ésta máquina por su rapidez de corte y costo reducido. Para evitar el problema de los agujeros internos y tener que cortar el hilo para cada uno, se cortó pequeñas partes entre los extremos del link y los agujeros, esto permite ahorrar tiempo de máquina y dinero.



Figura 23 Inicio de corte de links

Fuente: Autor

La máquina utiliza una interfaz tipo CAD (diseño asistido por computadora) para seguir los vectores dibujados previamente en AUTOCAD.



Figura 24 Interfaz gráfica donde se aprecia el avance del corte

Fuente: Autor



Figura 25 Links lateral y unión triple en el primer corte

Fuente: Autor

2.8 PARTES DEL ROBOT

A continuación se detalla las piezas que fueron cortadas con electroerosión por hilo:

2.8.1 Link Simple

Este link, el cual es el más sencillo, sirve de sujeción a las ruedas posteriores. Su movimiento es solo de rotación (pitch) con respecto al eje del robot. La fuerza de los resortes de tracción es transmitida hacia los tres links simples empujando las ruedas hacia el exterior.

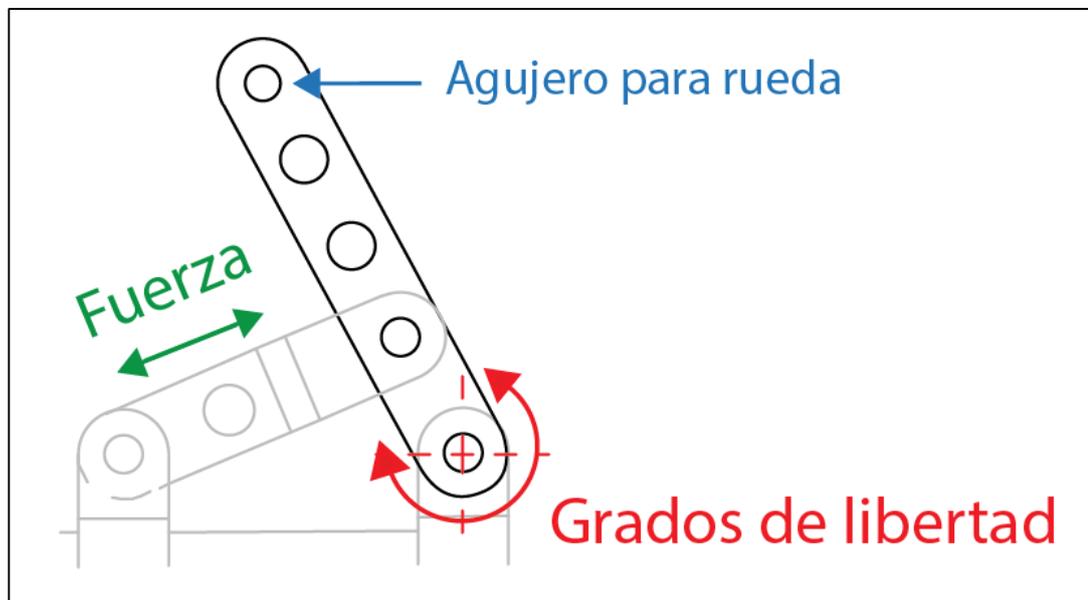


Figura 26 Rango de movimiento link simple

Fuente: Autor

Fue fabricado a partir de una plancha de hierro en un solo proceso con electro erosión de hilo. Al igual que los demás links terminan en forma redonda para evitar cortes por filos vivos y mejorar la estética.

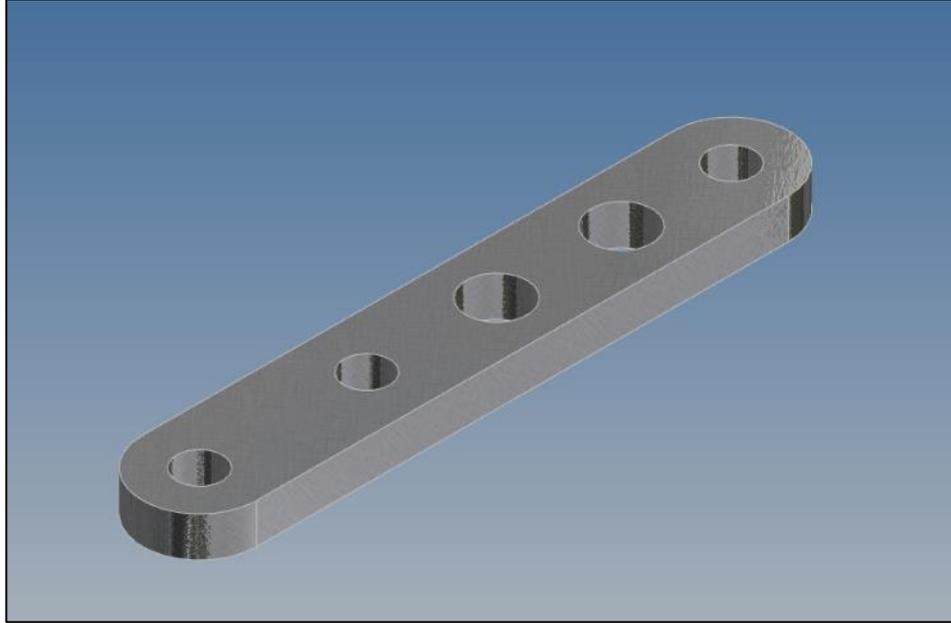


Figura 27 Link simple finalizado

Fuente: Autor

2.8.2 Link motor

Tiene un funcionamiento similar al link simple con la única diferencia que la fuerza es aplicada por el resorte de compresión central. Para la sujeción de los motores se diseñó un pequeño brazo en forma de arco con dos agujeros avellanados para tornillos M3.

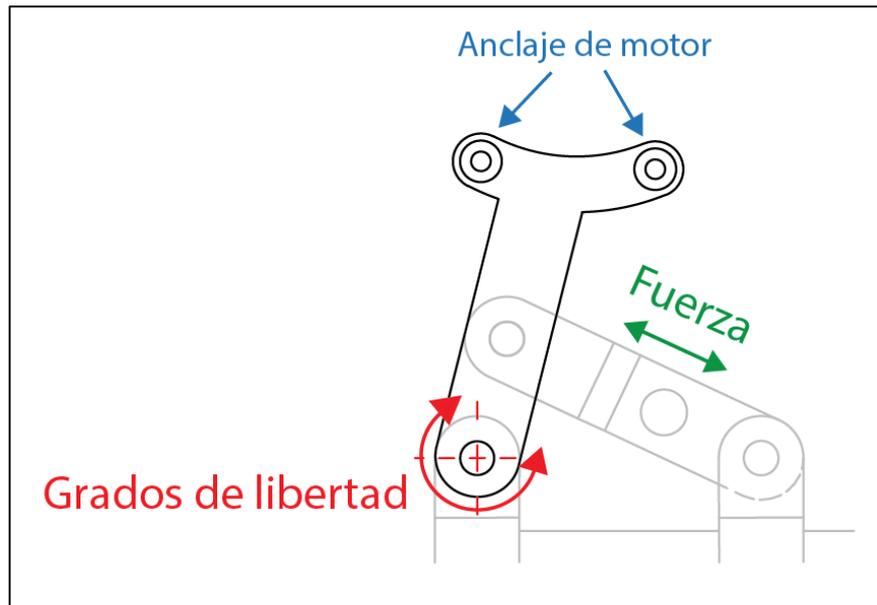


Figura 28 Rango de movimiento link motor

Fuente: Autor

Al igual que el link simple se cortaron los 3 links motor en la electroerosión de hilo y posteriormente se realizó el avellanado con una fresadora para que las cabezas de los tornillos no afecten el movimiento del link.

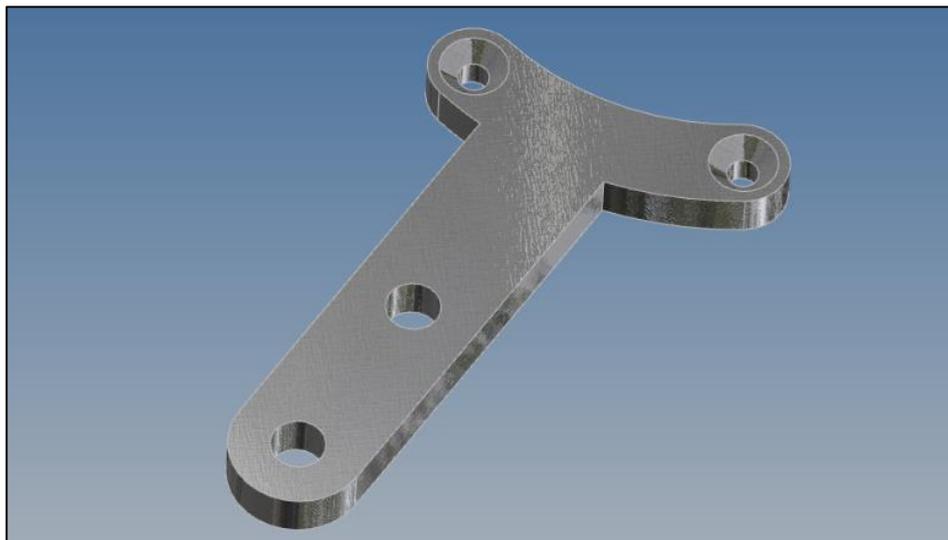


Figura 29 Link motor finalizado

Fuente: Autor

2.8.3 Link lateral

Estos links con forma de horquilla o de “Y” son los responsables de transmitir la fuerza de los resortes horizontal y convertirla en vertical hacia los links simple y links motor. Fueron diseñados de manera que colisionen y se detengan cuando el robot está completamente extendido asegurando que las ruedas y motores nunca sobrepasaran los 90 grados con respecto al eje.

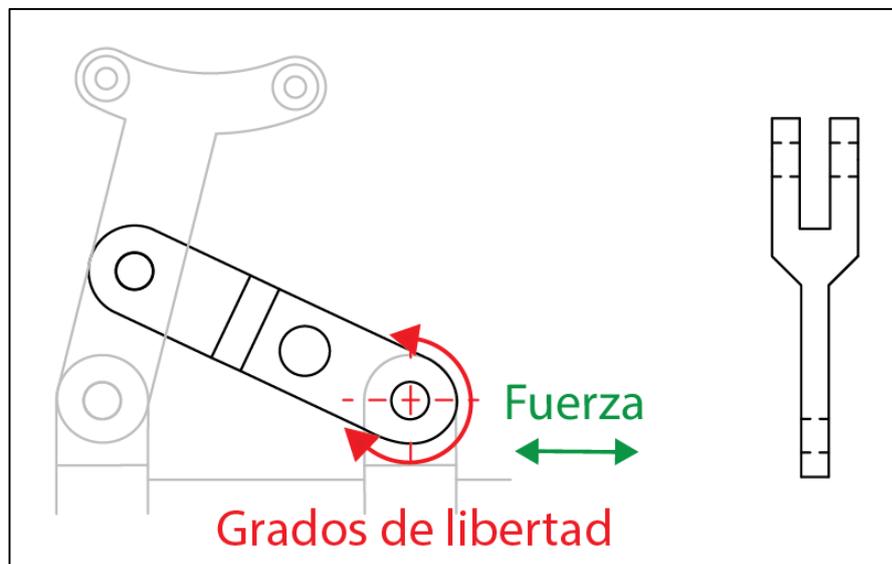


Figura 30 Rango de movimiento link lateral

Fuente: Autor

La fabricación fue realizada en dos procesos como me aprecia en la siguiente imagen, primero se cortó la forma básica y segundo se giró la pieza para cortar los agujeros y radios. En total construyeron 6 de estos links.

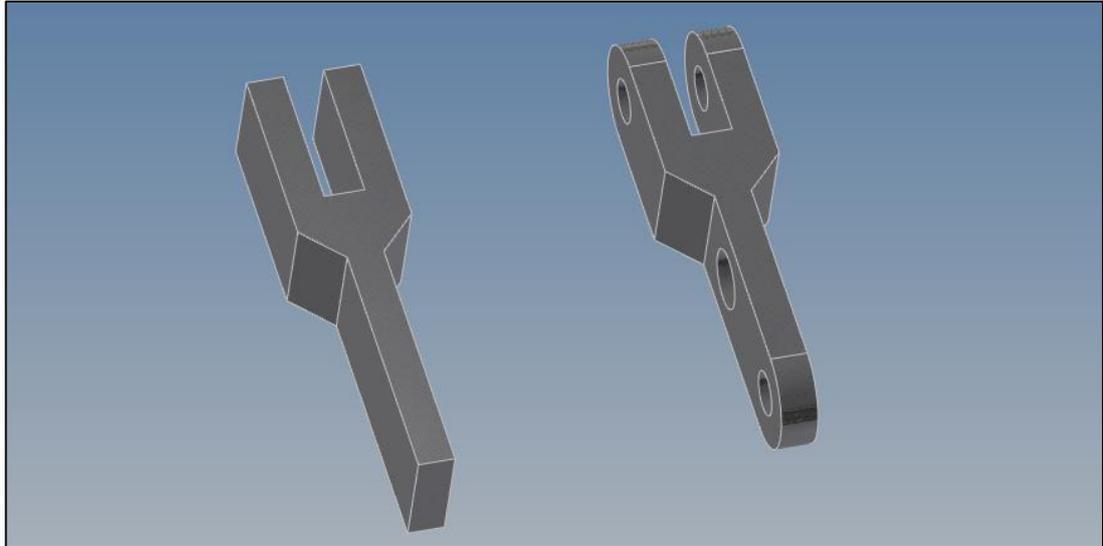


Figura 31 Link lateral primera etapa (izq.) y finalizado

Fuente: Autor

2.8.4 Link unión triple

Estos links triples unen todos los otros links al eje. Trabajan en pares, siendo uno estático donde en base a estos rotan los link y el otro móvil en el cual los resortes aplican la fuerza. Los 4 componentes son iguales, para que uno se deslice y el otro sea fijo se realizó ajuste en el eje de dos décimas de milímetro. Si a estos link se les aplica fuerza suficiente pueden rotar en el sentido roll. Esto es útil para alinear las ruedas frontales con las posteriores

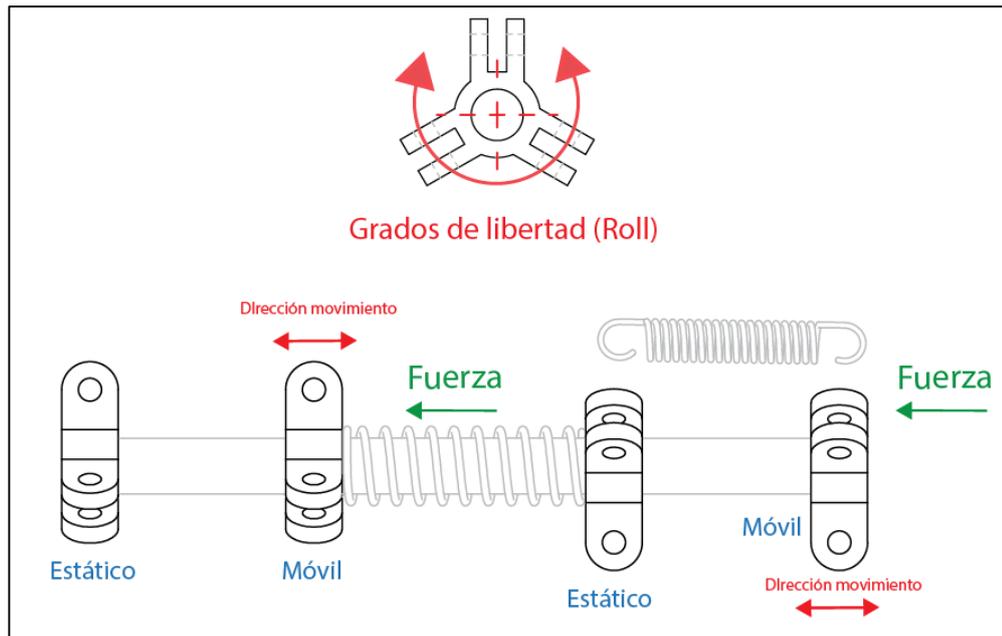


Figura 32 Rango de movimiento link triple

Fuente: Autor

El proceso de fabricación fue el más complejo de todos los links. Primero se cortó la forma base y luego se roto tres veces a 120 grados para cada sujeción y radios. En comparación de tiempo, estos links tomaron el mismo tiempo de fabricación que todos los otros links antes mencionados.

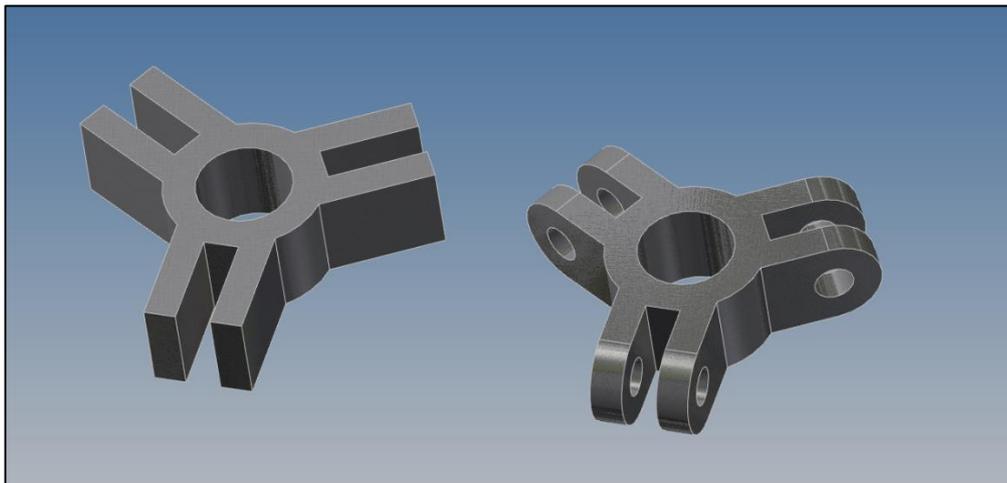


Figura 33 Link triple primera etapa (izq.) y finalizado

Fuente: Autor

2.8.5 Acoples servomotores

Para sujetar los servos motores que mueven la cámara se elaboraron 4 partes a partir de plancha de hierro de 3mm. Se optó por esta medida ya que se contaba con material disponible y es lo suficiente maleable para doblarlo sin mucho esfuerzo.

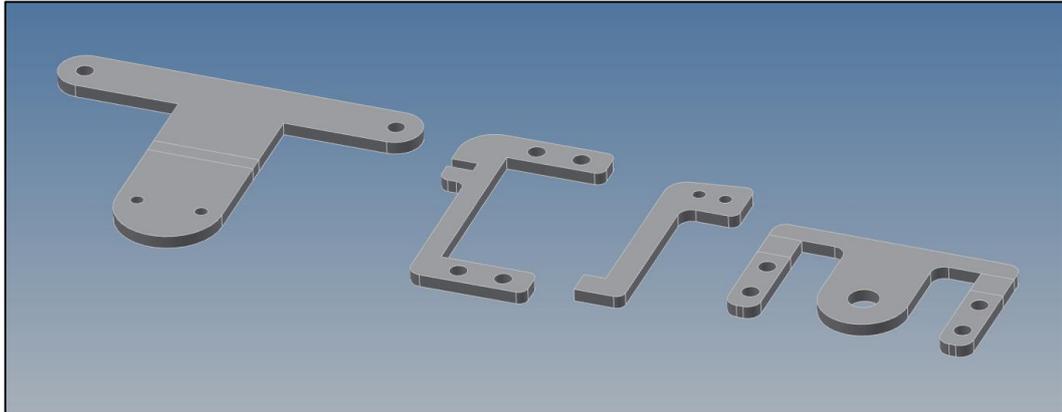


Figura 34 Elementos para acoplar la cámara el robot (sin doblar)

Fuente: Autor

Elementos de acople de servomotores:

- Porta servomotor 1 (amarillo)
- Porta servomotor 2 (naranja)
- Unión servomotor 1 (morado)
- Unión servomotor 2 (azul)

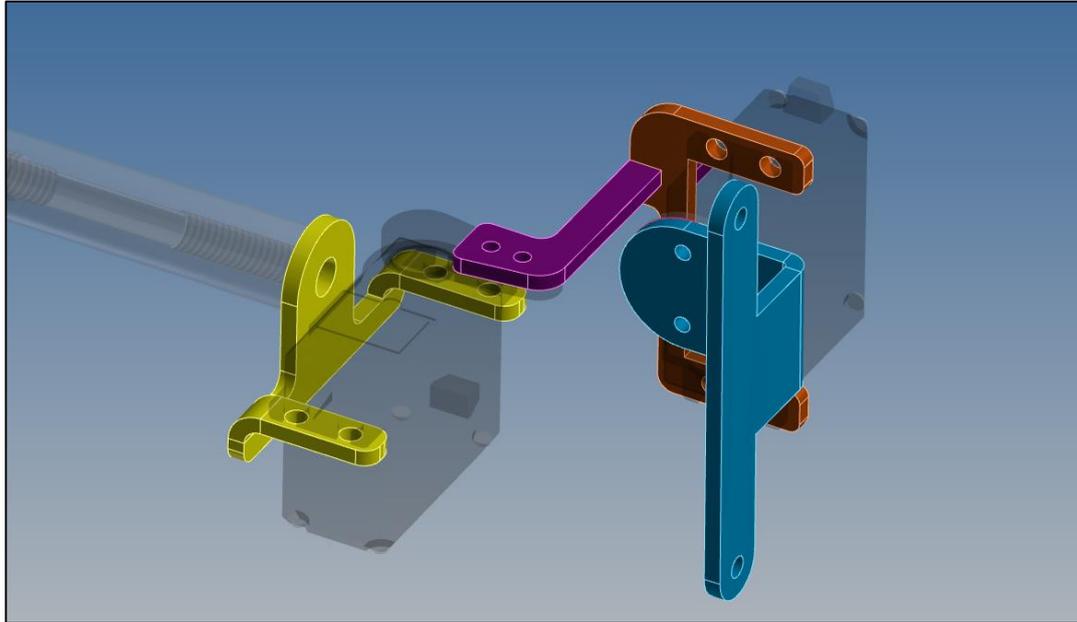


Figura 35 Elementos para acoplar la cámara el robot (doblados y en posición)

Fuente: Autor

Dos de estas partes (porta servomotor 1 y unión servomotor 2) fueron dobladas posteriormente con prensa hidráulica porque resultaba muy costoso realizarlas a partir de bloques macizos. Al doblarlas a partir de una plancha se obtiene una estructura resistente, liviana rápida de construir.

El porta servomotor 1 se unió al eje del robot con un perno M8, a este se atornilla el primer servo que realiza el movimiento horizontal de la cámara. De un adaptador en forma de gota que está incluido con los servomotores parte la unión servomotor 1 que está soldada al porta servomotor 2. Este único punto de soldadura de todo el robot fue necesario al ser la única manera de unir los dos servomotores sin construir una parte extra de mucha complejidad.

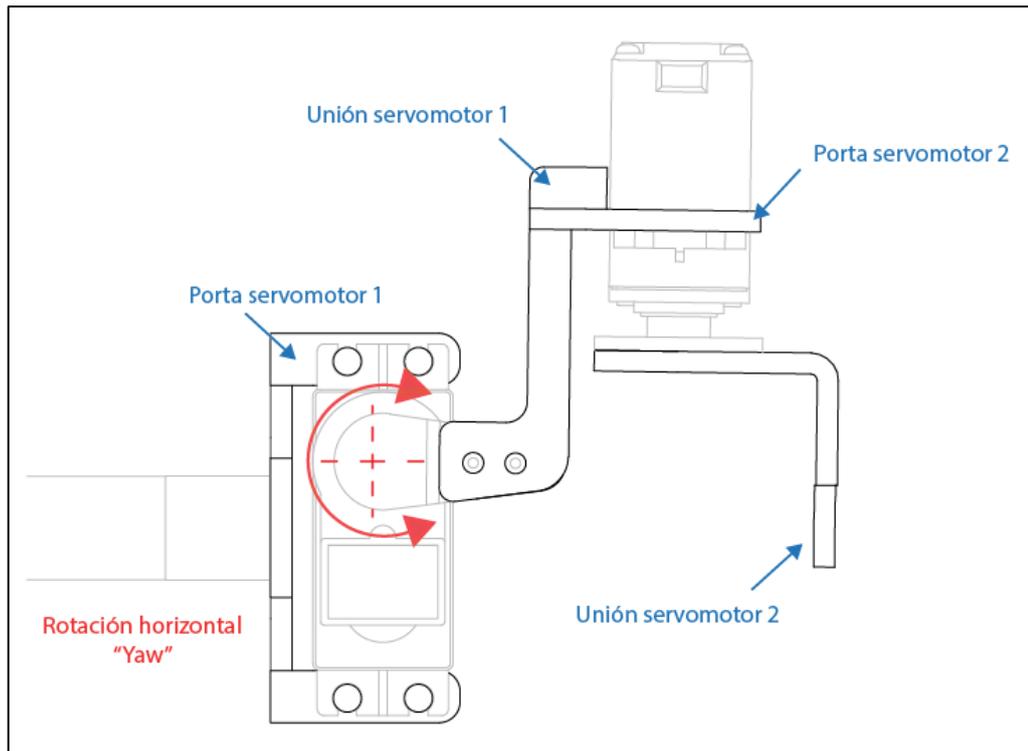


Figura 36 Rotación horizontal del conjunto

Fuente: Autor

Finalmente la unión servomotor 2 rota con el segundo servo dando el giro vertical a la cámara. Esta unión cuenta con 2 agujeros para atornillar la placa de la cámara IP con tornillos M3 largos y contratueras de separación evitando el contacto de los pines con partes metálicas.

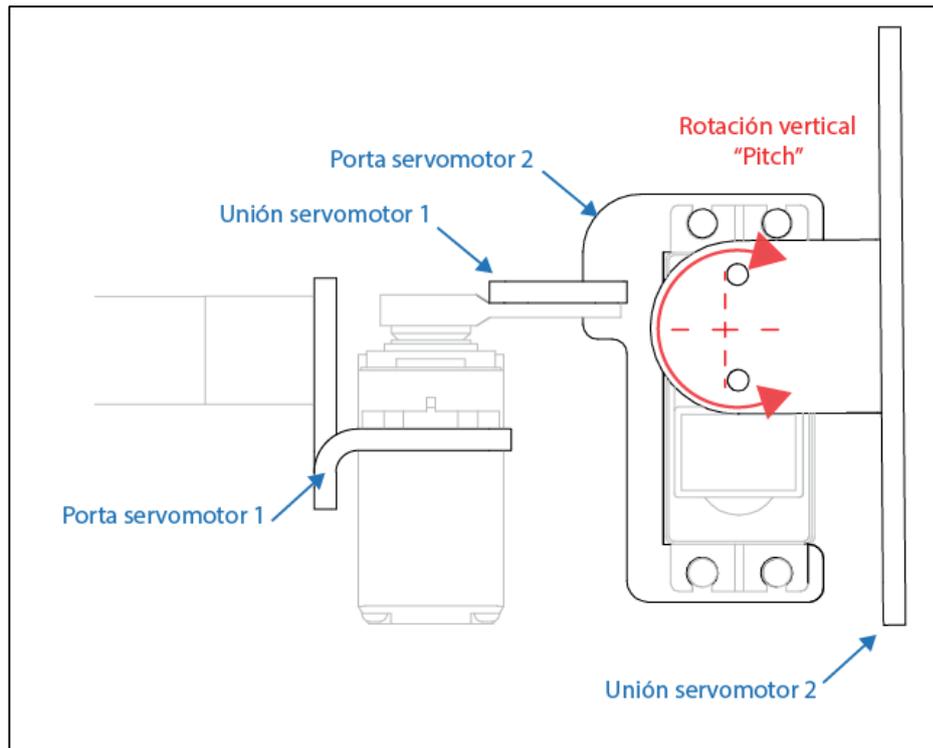


Figura 37 Rotación vertical del conjunto

Fuente: Autor

Antes de unir las diferentes partes se revisó que el rango de movimiento de los servomotores (180°) fuera óptimo para evitar colisiones con el robot, aunque esto deberá ser controlado por software debido a puntos donde puede chocar la cámara con las ruedas.

2.8.6 Porta baterías

Para fijar las baterías al robot se analizó el lugar más conveniente debido a que estas son los elementos más grandes y por lo tanto los que más molestan cuando el robot tiene que comprimirse. Se optó por colocar las baterías debajo de las ruedas en los link simples en la parte posterior, de esta manera seguiría el ángulo de cada rueda cuando el robot tenga que comprimirse para ingresar al tubo. Para esto se construyó dos pequeñas porta baterías que servirían de soporte a cada batería.

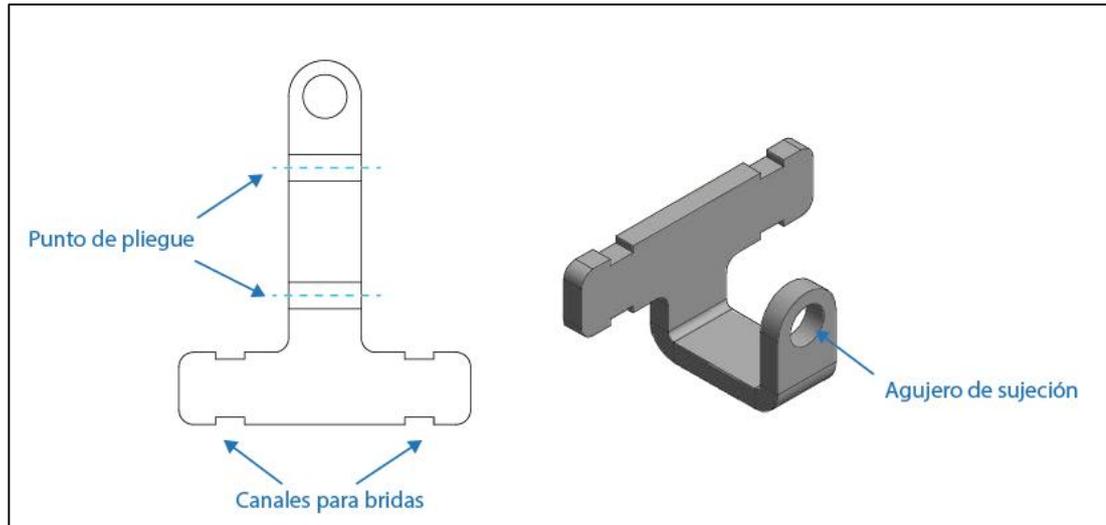


Figura 38 Porta baterías antes y después de doblar

Fuente: Autor

Estos elementos fueron cortados con electroerosión de hilo y la curvatura fue dada golpeando a mano con martillo sujetándolas con una entenalla debido a que eran muy pequeñas para la prensa hidráulica y la forma de “C” dificultaba el doblado.

Estos elementos se sujetan al robot en los links simples solamente con un perno M6, esto permite rotar ambas baterías en caso que necesiten ajustes. A su vez se hicieron dos pequeños surcos para pasar por ahí bridas o amarras plásticas para cable de 3mm de ancho pudiendo sujetar las baterías de forma casi inmóvil sin tener que realizar orificios de sujeción que dañen a estas.

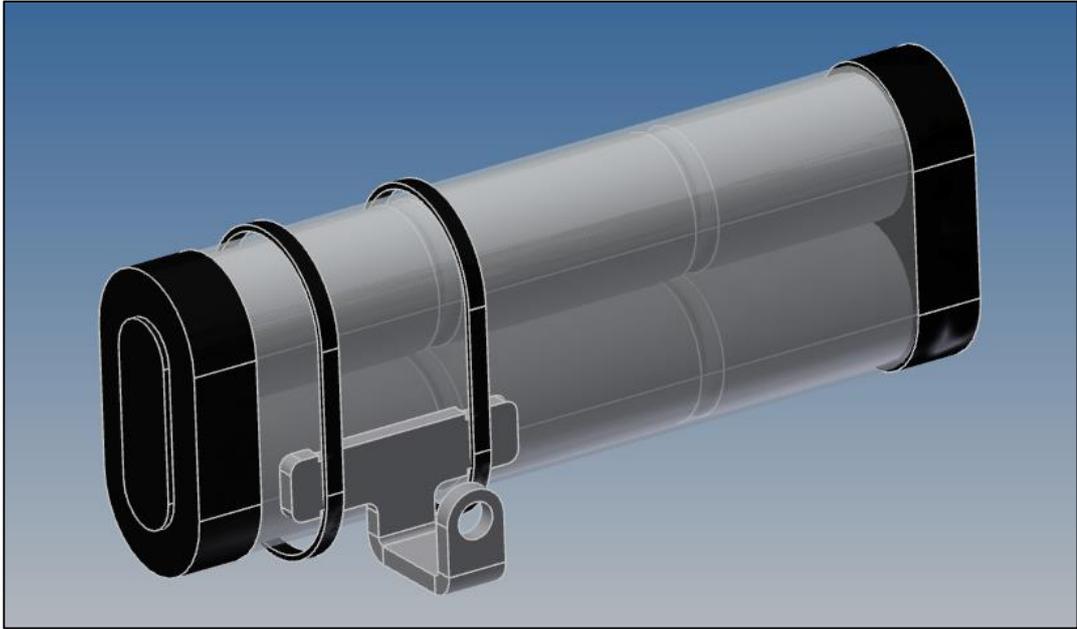


Figura 39 Sujeción de la batería con bridas plásticas

Fuente: Autor

2.8.7 Sujeción PCB

Como se explicará más adelante, se fabricó una placa PCB donde están localizados drivers, borneras y otros componentes. Para sujetar esta placa se cortó un delgado marco de hierro y con 4 tornillos M3 en cada esquina se armó un porta PCB que se une al eje posterior del robot con un perno de cabeza hexagonal. De manera similar al porta baterías se realizó un solo corte de la plancha para luego doblar con martillo el extremo del agujero.

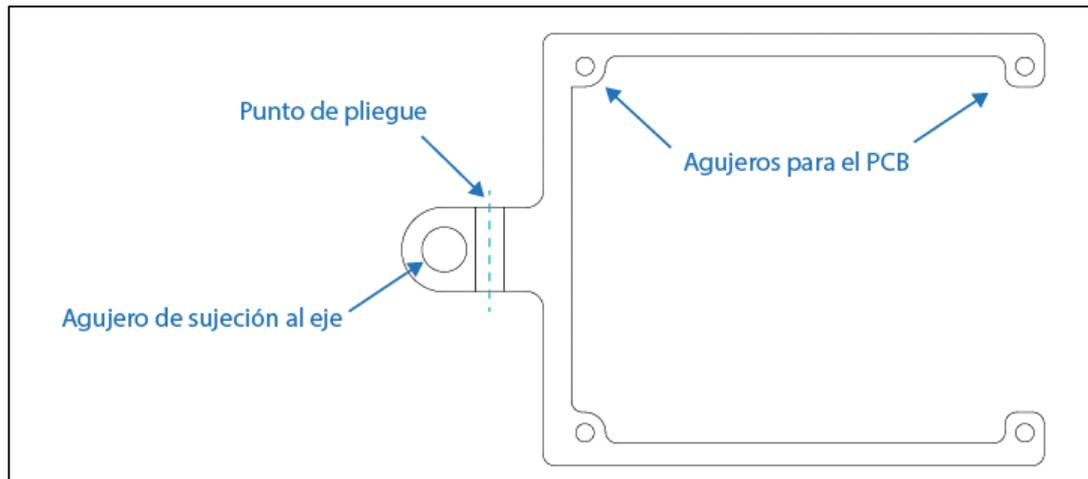


Figura 40 Porta PCB

Fuente: Autor

El acople entre el PCB y el marco es de solamente 0.5mm por lado para que no haga contacto entre los pines y el hierro. Se optó por no apretar demasiado el porta PCB al eje para poder rotarlo en caso de que los cables requieran cierta holgura y no queden muy tensos.

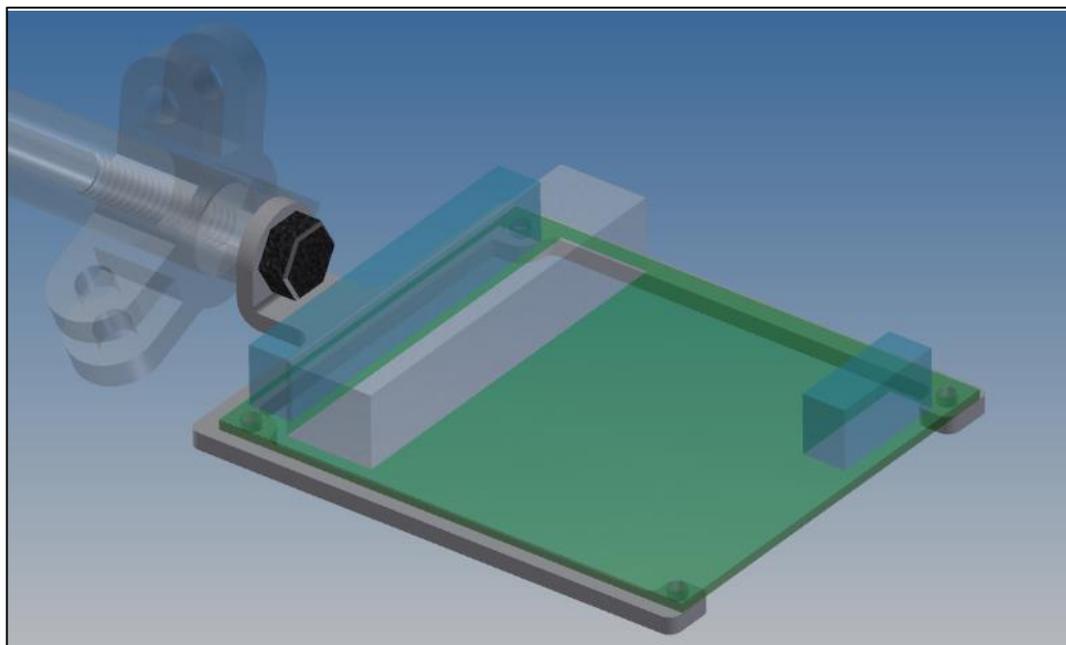


Figura 41 Sujeción porta PCB al eje del robot

Fuente: Autor

2.9 TRATAMIENTO TERMICO (PAVONADO)

Al pasar alrededor de un mes del corte de las piezas empezaron a aparecer pequeñas manchas amarillas que fueron identificadas como principios de oxidación por el contacto de las manos salinas y el hierro. Esto dificultó el movimiento del robot y el rose empezó a hacerse evidente. Para solucionar este inconveniente de manera inmediata se optó por aplicar un tratamiento térmico llamado pavonado.

Según Adriana Montiel en su investigación titulada “ESTUDIO DE LA APLICACIÓN DE TRATAMIENTOS DE CONVERSIÓN QUÍMICA DE PAVONADO SELLADO CON CERIO SOBRE ACERO DE REFUERZO EMBEBIDO EN CONCRETO COMO PROTECCIÓN ANTICORROSIVA” el pavonado es la creación de una fina capa (micras) de óxido ferroso-diférrico mediante una oxidación química controlada al calentar las piezas de hierro en un horno hasta aproximadamente 150°C y sumergirlas inmediatamente en aceite soluble al agua temperatura ambiente. El choque térmico y el aceite crean una capa de color negro muy resistente a la corrosión sin variar las medidas externas de las piezas. Variando la temperatura se puede obtener diferentes tonos de colores en el hierro.



Figura 42 Diferencia entre tornillo en poseso de oxidación y tornillo pavonado

Fuente: (Zona Boxes, s.f.)

Todos los links al igual que el eje y los porta servomotores fueron tratados con este proceso que tomo alrededor de 2 horas asegurando que el óxido no volverá a limitar el movimiento del robot

2.10 ENSAMBLAJE

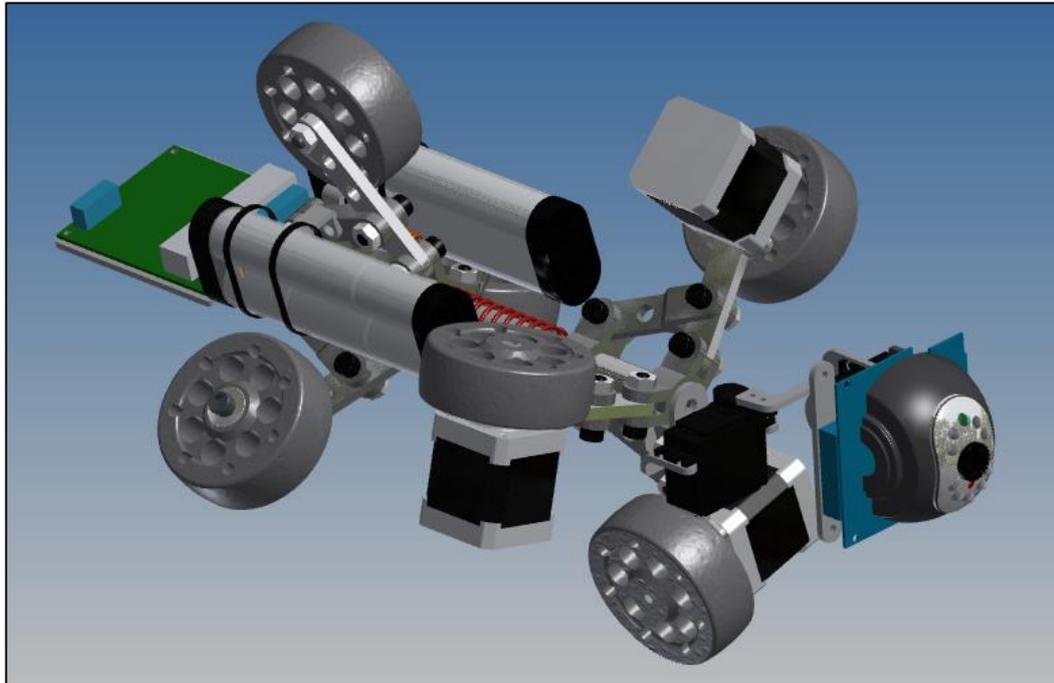


Figura 43 Robot ensamblado

Fuente: Autor

El ensamblaje del robot está descrito por la siguiente secuencia, con ayuda de los planos se puede guiar para la identificación de las partes.

Inicialmente atornillar con un perno M6 cada rueda posterior con un link simple usando dos arandelas para reducir la fricción (Anexo 18). Para las ruedas delanteras, colocar el adaptador en el eje de las ruedas con la ayuda de una prensa hidráulica, luego atornillar cada motor a un link motor e insertas las ruedas (Anexo 19).

Para el eje primero se debe insertar con prensa los links triples a cada eje extremo por el lado de 15mm de diámetro. Seguido insertar la unión ejes rectos en el resorte y los

links triples faltantes en cada extremo de los ejes para finalmente atornillar todo el conjunto (Anexo 21). Luego unir cada uno de los seis links laterales al eje con pernos M6 así como las ruedas delanteras y posteriores. Comprimir el resorte central para pasar un perno entre cada link lateral y los links simples y links motor (Anexo 22).

Para la cámara, primero colocar cada servo en los porta servomotor 1 y 2 con cuatro tornillos avellanados M3. Luego combinar ambas partes mediante la unión 1 y finalmente atornillar la unión 2 al servomotor libre para colocar la placa de la cámara IP con tornillos M3 de 25mm (Anexo23).

Finalmente, unir todo el porta cámara con un tornillo M8 entre el porta servomotor 1 y el eje frontal del robot. Con ayuda de una llave de boca, atornillar los dos porta baterías a un link lateral posterior y sujetar las baterías con amarras plásticas. En la parte posterior colocar con perno hexagonal M8 el porta PCB y atornillar el PCB con tornillos M3. En los extremos de dos links lateral posterior sujetar los dos reguladores de tensión con amarras. Por ultimo cablear el circuito eléctrico de la cámara, motores, baterías y reguladores de tensión hacia el PCB.

CAPITULO III

DISEÑO ELECTRÓNICO

3.1 INTRODUCCIÓN

El diseño electrónico es una de las etapas fundamentales de la creación del robot, ya que por medio de este desarrollo se puede conseguir la movilidad y control del mismo. En este capítulo se considerará el desarrollo para la conexión inalámbrica, fuentes de alimentación, control de cámara y motores.

3.2 ELEMENTOS A UTILIZAR:

3.2.1 Cámara IP

Al ser este proyecto un robot explorador, una de las características más importantes es poder monitorear el entorno, para esto se ha decidido utilizar una cámara de video IP, la cual transmite imágenes en tiempo real directamente por una red inalámbrica.

Para esto, se ha incorporado al robot la cámara EasyN FS-613B-M166 Wireless, adaptada a las necesidades de este proyecto, tiene las siguientes características:

- Conectividad Wireless tipo N, además conectividad mediante cable de red.
- Visión nocturna de hasta 10 m con luz infrarroja (sensor de iluminación).
- Utiliza el estándar 802.11b (WiFi) para transmisión inalámbrica o los protocolos TCP/IP, HTTP, ICMP, DHCP o FTP para conexión directa
- Cámara VGA con resolución de hasta 640 x 480 a 25 cuadros por segundo en formato JPEG.
- Alimentación 5V
- Standard Wireless IEEE 802.11 b/g



Figura 44 Cámara IP EasyN FS-613B-M166 Wireless.

Fuente: (EasyNUSA, 2014)

3.2.2 Conexión inalámbrica

Para conectar los dispositivos (computador, cámara) a una misma red y poder recibir las imágenes del video es necesarios tener un equipo que logre esta comunicación inalámbrica y que transmita los datos de una forma segura al equipo remoto.

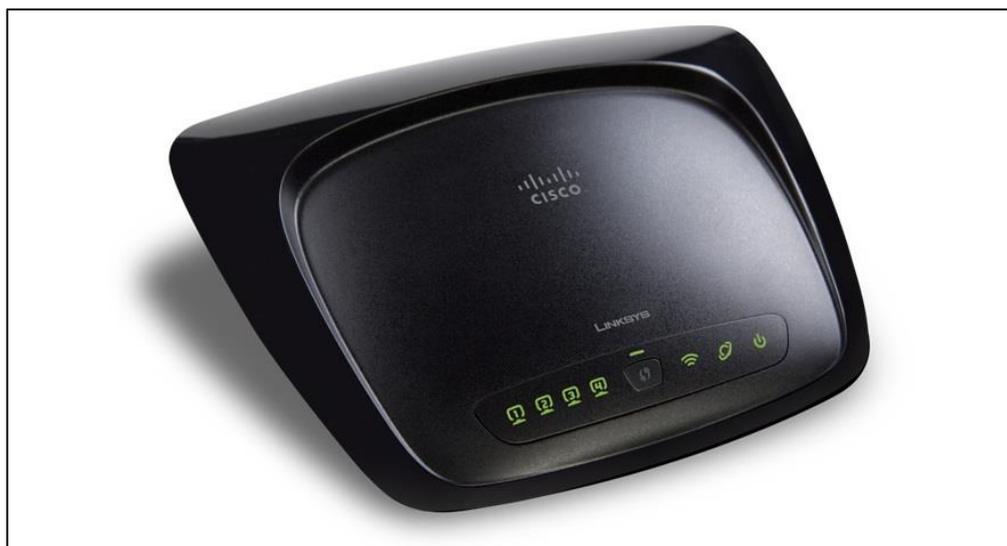


Figura 45 Punto de Acceso Inalámbrico Linksys WRT54G2 Wireless-G Router

Fuente: (LINKSYS, 2014)

El equipo elegido es un punto de acceso inalámbrico Linksys WRT54G2 Wireless-G Router con capacidad de conexión con redes tipo b/g de 11 Mbps y 54 Mbps con 2 antenas incorporadas. Posee cuatro puertos Ethernet 10/100M LAN con conector RJ-45 y seguridad WEP, WPA y WPA2. Es de tamaño y peso reducidos, con un funcionamiento a 12 VDC y un consumo máximo de 0,5 A. Su rango de alcance es de 100 metros en interiores y 300 metros en exteriores.

3.2.3 Arduino

Este instrumento fue desarrollado en Italia, con el principal objetivo de generar una plataforma sencilla de programar, económica, multiplataforma y potente. Para grabar el código en la tarjeta, se utiliza una conexión con cable USB que puede ser usada tanto en MAC como en PC, con lo que no se necesitan equipos adicionales. Cuenta tanto con hardware como con software libre lo haciéndolo una herramienta muy útil para desarrollo e investigación, convirtiéndose en una opción atractiva para estudiantes.

Básicamente es una plataforma basada en una placa electrónica que contiene un microcontrolador Atmel, que provee un cierto número de puertos de entrada y salida, y en un entorno de programación de alto nivel basado en Processing, que programa el microcontrolador mediante USB.



Figura 46 Logo de Arduino

Fuente: (Wikimedia Commons, 2013)

Al contar con todas estas características no pasó mucho tiempo hasta que tuviera gran aceptación en el mercado, tanto para aficionados como para profesionales y estudiantes de ingeniería, reduciendo los tiempos de desarrollo de proyectos. Una de las mayores

ventajas de este dispositivo es que al ser libre, se tiene una gran cantidad de comunidades interesadas en el desarrollo, corrección y mantenimiento de accesorios y programas. Uno de los foros más grandes e interesantes es <http://forum.arduino.cc/>, el cual aporta con ideas y apoyan al desarrollo y mejoramiento de ciertos proyectos. Al tener tanto éxito y acogida a nivel mundial, muchas otras marcas han desarrollado su propia versión de Arduino, siendo estas Netduino, Microchip, SainSmart, entre otras. Paralelamente se desarrollan también una variedad de placas oficiales Arduino, diseñadas aplicaciones y necesidades específicas, una de ellas también ha sido utilizada en este proyecto, de la que se hablara más adelante.

Ahora revisaremos las características de este equipo para poder analizar su utilidad en el proyecto. Como se mencionó anteriormente, la placa cuenta con un microcontrolador principal, Atmel AVR de 8 bits, el cual posee un bootloader que permite la comunicación inicial con el computador.

La fuente de alimentación del circuito generalmente se realiza por medio de obtención de energía del puerto USB, pero en un caso en el que la placa no va a estar conectada al computador, o no se tiene este tipo de alimentación, se cuenta también con un conector que permite obtener la energía de una fuente externa que debe ser entre 3.3V y 5V.

La cantidad de entradas tanto analógicas como digitales y las salidas dependen específicamente del modelo de Arduino que se adquiera. Pero la disposición y modo de conexión es la misma o muy similar en todos ellos. Generalmente estas entradas y salidas se las encuentra como conectores hembra tipo peineta, con separaciones fijas, permitiendo una adaptabilidad muy alta a los sensores o actuadores conocidos comúnmente como shields.

En el mercado existe una gran variedad de placas Arduino, pero en este proyecto de investigación se utilizara la placa Arduino Mega 2560. Este modelo posee un microcontrolador ATmega2560, que posee 16 entradas analógicas y 54 pines de entrada-salida.

La velocidad de procesamiento es de 16MHz y cuenta con una memoria de 256kB.

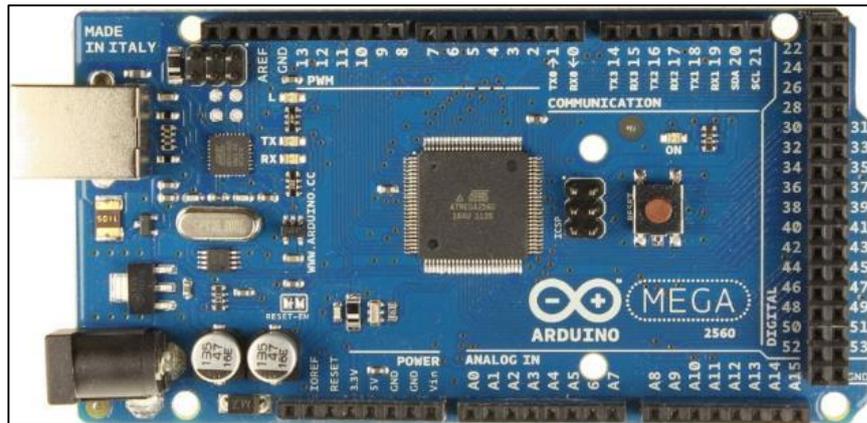


Figura 47 Placa Arduino Mega 2560

Fuente: (ARDUINO, s.f.)

Al ser un proyecto netamente de investigación, con fines educativos y con recursos limitados, siendo considerado como un prototipo, se ha definido que esta es una herramienta muy válida, pero al momento de realizar un proyecto para ser utilizado en el campo, se debería reemplazar por un procesador más robusto y con mayor capacidad de procesamiento, de lo que se hablará en las conclusiones.

Las principales características son las siguientes:

- 15 canales de PWM
- 4 sistemas de comunicación Serial nativo (USART):

Tabla 2 Puertos Serial en Arduino Mega

	RX	TX
SERIAL	0	1
SERIAL 1	19	18
SERIAL 2	17	16
SERIAL 3	15	14

Fuente: Autor

- 6 interrupciones externas

- Sistema de comunicación SPI, (MISO, MOSI, SCK, SS).
- Sistema de comunicación I2C (SDA y SCL).

3.2.4 Shield Bluetooth

Las placas Arduino cuentan con una gran adaptabilidad a las necesidades, para esto se desarrollan diferentes tipos de shields, o placas de circuito impreso que se colocan sobre los conectores hembra de la placa base, muchos de ellos son desarrollados oficialmente por Arduino pero en otros casos, y siendo la mayoría, diseñados por otras empresas, tanto para las comunicaciones (Ethernet, Zigbee, radiofrecuencia, etc.), pantallas, luces, sensores, entre otros.

Para este prototipo se utilizará un Shield de aplicación específica, para la comunicación Bluetooth.

A continuación revisaremos los tipos de conexiones Bluetooth que existen y la que se utiliza en el módulo:

Según la página oficial de Bluetooth (Bluetooth, 2015), éste nace en 1994 creado por Ericsson, una compañía Suiza como alternativa Wireless a la comunicación RS-232. Utiliza radio transmisión para el intercambio de datos y trabaja en las bandas de 2,4 a 2,485 GHz, esta banda generalmente no requiere licencia, con comunicación full-duplex, espectro expandido y una tasa nominal de 1600 saltos/seg, esto quiere decir que esta tecnología es capaz de adaptarse a la interferencia del espectro en el que trabaja (*AFH Adaptive frequency hopping*), es decir aprovecha la frecuencia disponible. Esto se realiza detectando las frecuencias que están siendo usadas y evadiéndolas. Estos saltos se realizan entre 79 frecuencia con separación de 1MHz evadiendo de manera eficiente la interferencia.

Esta tecnología es ampliamente usada en muchos ámbitos, desde los teléfonos celulares hasta aplicaciones y equipos médicos. Permite enviar cualquier tipo de datos, ya sea música, imágenes, videos, etc. A diferencia de las comunicaciones de radio frecuencia que se utilizan para los teléfonos celulares, televisión y radio FM (alcanza

varios Km), la tecnología bluetooth está diseñada para comunicaciones de corto alcance, que pueden llegar hasta los 100m.

Otras de las características que sobresalen de esta comunicación es el bajo consumo y que costo que puede llegar a tener.

Para que exista comunicación entre 2 puntos utilizando bluetooth, ambos deben contar con el software y hardware necesario, y luego se realiza el comúnmente conocido “Pareo” o “Emparejamiento”, básicamente 2 equipos cualesquiera que tenga bluetooth activado pueden emparejarse en cualquier lugar del mundo solamente con estar ubicados dentro del rango de alcance.

Ahora analizaremos las diferentes clases que existen dentro de esta comunicación:

Tabla 3 Comparación de clases de Bluetooth

CLASE	ALCANCE (MAX)	POTENCIA CONSUMO
1	100 m (20 dBm)	100 mW
2	10 m (4 dBm)	2.5 mW
3	1 m (0 dBm)	1 mW

Fuente: Autor

Los alcances y potencias establecidos en la tabla, son valores teóricos, ya que estos parámetros dependerán de varios factores ambientales. La clase más común usada en dispositivos es la clase 2, debido a su bajo consumo y alcance moderado.

Como se indicó anteriormente, para este proyecto se adquirió un Shield bluetooth, las características se describen a continuación:

- Voltaje de alimentación: 3,3V
- Baudrate: 9600, 19200, 38400, 57600, 115200, 230400, 460800
- Distancia de comunicación hasta 10 m
- Interface UART (TTL) con baudrate programable

- Baudrate por defecto 38400
- Bits de datos: 8
- Bit de parada: 1
- Antena integrada en el PCB

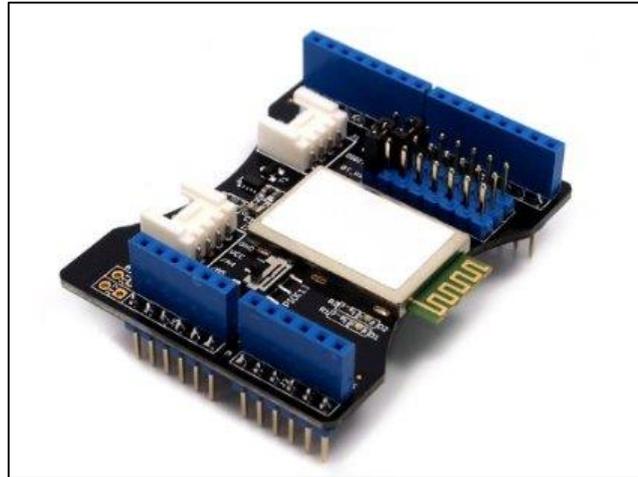


Figura 48 Shield bluetooth para Arduino

Fuente: (Seedstudio, 2014)

Las funciones de la interfaz son como se indica en la siguiente figura:

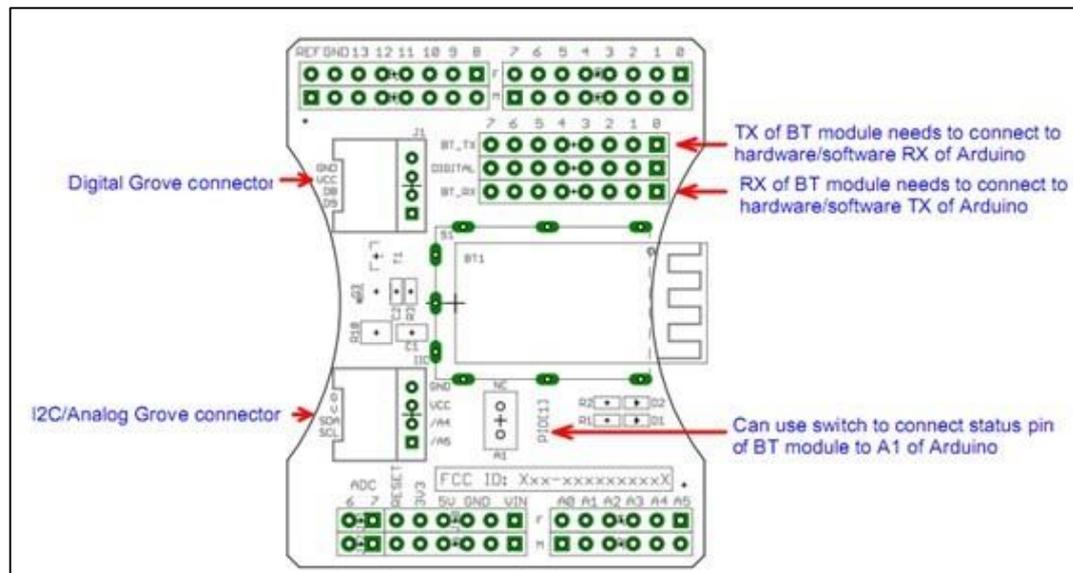


Figura 49 Esquema Shield Bluetooth para Arduino

Fuente: (Seedstudio, 2014)

Tabla 4 Puertos de Shield Bluetooth para Arduino

CONEXIÓN	DESCRIPCIÓN
PIO1	Puerto para indicar la conexión del dispositivo: HIGH-conectado LOW-desconectado
BT_RX	Entrada del UART del módulo bluetooth
BT_TX	Salida del UART del módulo Bluetooth
CONECTORES	Uno digital (D8 y D9), otro I2C/Analógico (A4 y A5)

Fuente: (Seedstudio, 2014)

Hay que tener en cuenta que el alcance de la conexión Bluetooth en este proyecto es limitada, debido al instrumento que se está utilizando, el alcance máximo es de 10 m sin obstáculos, pero se ha considerado que al ser un prototipo, y debido a la conveniencia de costos se ha decidido utilizar un módulo Bluetooth, en los siguientes capítulos se explicarán las opciones de mejora que se pueden implementar.

Para poder comunicar los dispositivos de este proyecto (Computador – Arduino), se deberá desarrollar un código para establecer dicha comunicación, más adelante se explicará este desarrollo.

3.2.5 Motores

En el mercado existen gran cantidad de motores que podrían servir en este proyecto, pero a continuación analizaremos las distintas opciones y decidiremos la mejor de ellas.

Motores DC: Son capaces de transformar la energía eléctrica en mecánica y de esta forma se produce una rotación en el eje provocando un movimiento rotatorio. El principio de funcionamiento es muy sencillo y recae en el hecho de que los polos opuestos se atraen, es por esto que un motor DC está constituido por un estator con devanados denominados polos (imanes que generan campo magnético) y un rotor, también devanado alimentado con corriente directa. Al momento que circula corriente

pero el hilo conductor del rotor se genera un electroimán que genera una fuerza al rechazar los polos del imán permanente, lo que genera la denominada fuerza de Lorentz, que hace girar el eje del motor.

Estos motores tienen como característica principal la rotación continua, cuando se los alimenta, empezarán a moverse sin parar y a una velocidad constante, dependiendo de la alimentación, hasta que se retire la corriente. Estos motores generalmente funcionan a altas revoluciones. Se puede llegar a controlar su velocidad, por medio del uso de PWM (*pulse width modulation* – modulación de ancho de pulso) y se puede invertir su sentido de giro, invirtiendo la polaridad de la corriente, esto generalmente se realiza utilizando un circuito externo (“Puente H”).

Servo motores: Los servo motores se basan en el funcionamiento de los Motores DC, es por esto que generalmente están compuestos de un motor DC, un circuito de control, un sensor de posición (generalmente un potenciómetro) y un conjunto de engranajes. Para el control de estos motores se usa igualmente PWM. Este tipo de motores nos permiten controlar específicamente la posición, lo que no sucede con los motores DC; por medio de una salida se aplica la señal PWM al motor, lo cual genera un movimiento hasta llegar a la posición determinada por el sensor. Este tipo de motores son usados específicamente para aplicaciones en las que se necesite controlar la posición de una manera certera. Otra de las limitaciones de este tipo de motores, es que el rango de giro puede variar solamente entre 0 y 180 grados, lo cual sería inútil para el uso en un robot de ruedas.

Motores de pasos: Un motor de pasos utiliza una serie de imanes dispuestos alrededor de un rotor. La conmutación debe ser realizada de manera externa mediante un controlador, para energizar cada uno de los electroimanes o bobinas y determinar la posición del motor. Para hacer girar al motor se debe utilizar una secuencia específica y la cual generará el movimiento del motor ciertos grados, a cada uno de estos movimientos se los denomina pasos, por lo que el motor puede girar un número específico de pasos, son un ángulo determinado alrededor de los 360 grados de giro.

Existen dos tipos de estos motores: unipolares y bipolares, los más potentes son los bipolares. Para invertir el sentido de giro, solamente se requiere cambiar la secuencia de control.

Teniendo en cuenta todo lo expuesto anteriormente se ha realizado un cuadro comparativo:

Tabla 5 Tabla comparativa de motores comerciales

<i>MOTOR</i>	<i>VENTAJAS</i>	<i>DESVENTAJAS</i>
<i>DC</i>	<ul style="list-style-type: none"> • Rotación continua • Gran torque 	<ul style="list-style-type: none"> • Veloz • Requiere cajas de reducción
<i>PASOS</i>	<ul style="list-style-type: none"> • Rotación precisa • Fácil control • Menor costo • Bloqueo en <i>stand by</i> • Torque menor que los DC o servos, con respecto al tamaño normalizado 	<ul style="list-style-type: none"> • Mayor consumo de corriente • Requiere circuito de control
<i>SERVO</i>	<ul style="list-style-type: none"> • Gran torque • Posicionamiento preciso 	<ul style="list-style-type: none"> • Rango limitado de rotación

Fuente: Autor

Con todos estos criterios y según nuestras necesidades básicas e imprescindibles que son:

- Rotación de 360 grados
- Bloqueo de las ruedas cuando el motor no esté en funcionamiento
- Bajo costo
- Fácil control
- Rotación controlable en el mayor rango posible
- Gran torque

Se puede determinar que la mejor opción de motores para este proyecto son los motores de pasos, ya que cumple con la mayoría de los requisitos.

Ahora que se ha decidido el tipo de motor a utilizar, se analizará más a profundidad el funcionamiento y características.

Existen dos tipos de motores paso a paso de imán permanente:

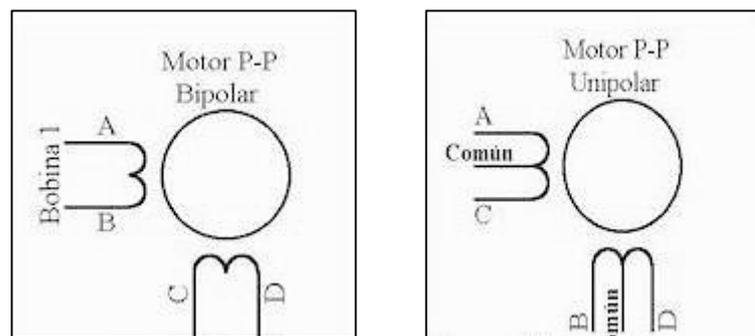


Figura 50 Tipos de motores de pasos de imán permanente.

Fuente: (ISLAS, 2007)

- **Bipolar:** Para ser controlados necesitan cierto tipo de secuencia para realizar un movimiento, haciendo circular una corriente inversa en las bobinas. Cada inversión de polaridad genera un movimiento denominado paso en el motor. El sentido de giro está determinado por la secuencia en la siguiente tabla.

Tabla 6 Secuencia para control de giro de motor bipolar

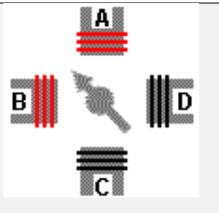
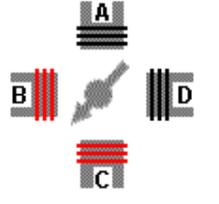
<i>PASO</i>	<i>TERMINALES</i>			
	A	B	C	D
1	+V	-V	+V	-V
2	+V	-V	-V	+V
3	-V	+V	-V	+V
4	-V	+V	+V	-V

Fuente: (ISLAS, 2007)

- **Unipolar:** Este tipo se caracteriza por ser más simple de controlar. Existen tres secuencias que se pueden utilizar para este tipo de motores. Para cambiar el sentido de giro sencillamente se debe seguir la secuencia en sentido inverso.

En la tabla a continuación se puede ver la secuencia más utilizada, la cual hace que el motor se mueva un paso por vez. Al tener 2 bobinas activas en el mismo momento se obtiene un gran torque.

Tabla 7 Secuencia para control de giro de motor unipolar de 2 bobinas.

PASO	BOBINA A	BOBINA B	BOBINA C	BOBINA D	
1	ON	ON	OFF	OFF	
2	OFF	ON	ON	OFF	

3	OFF	OFF	ON	ON	
4	ON	OFF	OFF	ON	

Fuente: (ISLAS, 2007)

La otra secuencia (wave drive) activa solo una bobina a la vez, por lo tanto el torque disminuye pero se obtiene una mayor precisión y suavidad en el movimiento.

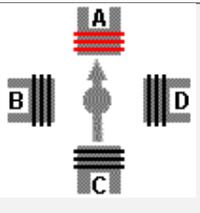
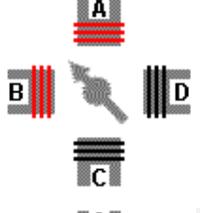
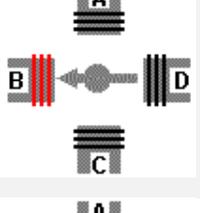
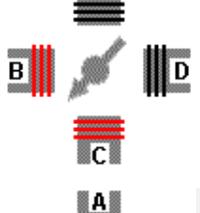
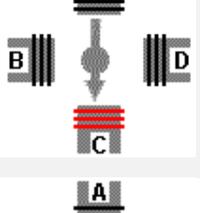
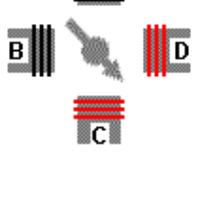
Tabla 8 Secuencia para control de giro de motor unipolar tipo Wave drive.

PASO	BOBINA A	BOBINA B	BOBINA C	BOBINA D	
1	ON	OFF	OFF	OFF	
2	OFF	ON	OFF	OFF	
3	OFF	OFF	ON	OFF	
4	OFF	OFF	OFF	ON	

Fuente: (ISLAS, 2007)

Por último se tiene la secuencia de medio paso, las bobinas son activadas generando un medio paso del original. Primero se activan 2 bobinas y luego 1, luego se repite. Esta secuencia consta de 8 movimientos.

Tabla 9 Secuencia para control de motor de pasos tipo "Medio Paso"

PASO	BOBINA A	BOBINA B	BOBINA C	BOBINA D	
1	ON	OFF	OFF	OFF	
2	ON	ON	OFF	OFF	
3	OFF	ON	OFF	OFF	
4	OFF	ON	ON	OFF	
5	OFF	OFF	ON	OFF	
6	OFF	OFF	ON	ON	

7	OFF	OFF	OFF	ON	
8	ON	OFF	OFF	ON	

Fuente: (ISLAS, 2007)

Debido a las limitaciones del mercado local, se tuvo que adquirir los motores en el exterior, para esto se ha tenido en cuenta varios aspectos, como son: rendimiento, torque, tamaño y costo. Por lo que se determinó que el adecuado sería el motor Pololu NEMA 17, con las siguientes especificaciones:

Este motor de pasos híbrido tiene un paso angular de 1.8° (200 pasos por revolución). Cada fase consume 1.2 A a 4V, permitiendo un torque de 3.2 kg-cm (44 oz-in). El motor consta de seis cables coloreados con terminales metálicos que permiten controlarlo mediante drivers unipolares o bipolares. Cuando se usa el driver unipolar se utilizan los seis cables, para un driver bipolar los cables amarillo y blanco quedan desconectados utilizando el par rojo-azul para una bobina y el par negro-verde para la otra. La recomendación del fabricante es usar un driver bipolar.



Figura 51 Motor Pololu NEMA 17

Fuente: (Pololu Robotics & Electronics, 2014)

Tiene un conveniente tamaño de 42.3 mm (ancho y largo) × 48 mm (alto), no incluido el eje (NEMA 17), su peso es de 350 g y con un eje de 5mm de diámetro. Todas estas características lo hacen muy elegible para ser utilizado en el prototipo de este trabajo de grado.

Para poder controlar de mejor manera un motor de pasos, existen varias opciones de los denominados DRIVERS en el mercado, que brindan, gran controlabilidad, facilidad de manejo de motores, estabilidad, bajo costo, manejo correcto de voltaje y corriente de la alimentación de los motores, en comparación con un control realizado artesanalmente. Es por esto que se ha decidido adquirir estos dispositivos, como se describe a continuación.

3.2.6 Driver

Para tener un adecuado funcionamiento del motor, se debe elegir un driver correcto. Para poder seleccionar el driver adecuado se debe tener en cuenta los siguientes aspectos:

- Voltaje de funcionamiento del motor a controlar
- Corriente que consume el motor por fase
- Resolución del paso que se requieren obtener

A continuación se puede observar una tabla comparativa de varios Drivers existentes en el mercado, con las características de cada uno, junto a las características del motor a utilizar.

Tabla 10 Comparación de drivers para motor de pasos

Modelo	MOTOR POLOLU NEMA 17	A4988	DVR8824	DVR8834	DVR8825
Imagen					
Voltaje de operación mínimo	4V	8V	8.2V	2.5V	8.2V
Voltaje de operación máximo		35V	45V	10.8V	45V
Corriente continua por fase	1,2A	1A	0.75A	1.5A	1.5A
Corriente máxima por fase		2A	1.2A	2A	2.2A
Voltaje lógico mínimo	N/A	3V	2.5V	2.5V	2.5V
Voltaje lógico máximo	N/A	5.5V	5.25V	5.5V	5.25V
Resolución de micro pasos	200 Pasos por revolución	full, 1/2, 1/4, 1/8, y 1/16	full, 1/2, 1/4, 1/8, 1/16, y 1/32	full, 1/2, 1/4, 1/8, 1/16, y 1/32	full, 1/2, 1/4, 1/8, 1/16, y 1/32

Fuente: Autor

Como se puede ver en el cuadro comparativo, es muy complicado conseguir Driver de bajo voltaje, generalmente los más comerciales son para 8V en adelante, en este caso se ha conseguido el “DRV8834”, que cumple con las necesidades del motor. Este modulo opera entre 2.5V a 10.8V, y entrega 1.5A por fase continuamente sin necesidad de un disipador, inclusive se puede llegar a los 2 amperios si se acopla un disipador adecuado. Cuenta con una protección de sobre corriente y temperatura y una resolución de 6 microsteps. Por lo tanto se debe utilizar uno de cada uno de estos para cada motor. Hay que tener en cuenta el excedente de corriente con el que se cuenta en el driver, nos garantiza que en cualquier caso, no se compromete el funcionamiento tanto del motor como del driver.

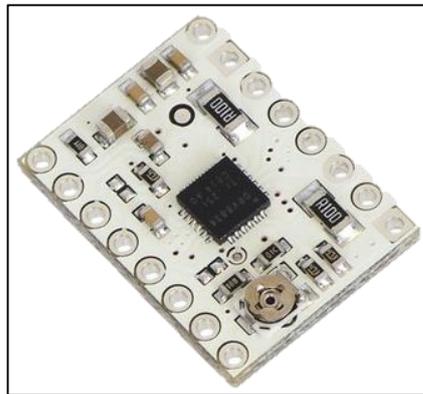


Figura 52. Driver DRV8834 para motores de pasos.

Fuente: (POLOLU ROBOTICS & ELECTRONICS, 2014)

Este controlador de motores es capaz de manejar las secuencias antes mencionadas para el control del motor, para esto cuenta con las siguientes entradas:

- VMOT - GND: En estas entradas se debe conectar la fuente de alimentación para el motor, como se mencionó anteriormente, el driver puede manejar voltajes entre 2,5 y 10,8 V. En este caso y teniendo en cuenta que los motores funcionan a 4V, el ingreso de la fuente debe variar entre 3,8 y 4,2V.
- M0 y M1: Estas dos entradas nos proporcionan un control del tipo de paso que vamos a manejar en los motores, esta es una gran ventaja ya que se puede

cambiar el paso del motor simplemente programación. Se debe tener en cuenta las siguiente tabla:

Tabla 11 Configuración de Paso de un motor de pasos por Estados

M0	M1	TIPO DE PASO
LOW	LOW	Paso completo
HIGH	LOW	1/2 paso
SIN CONEXIÓN	LOW	1/4 de paso
LOW	HIGH	1/8 de paso
HIGH	HIGH	1/16 de paso
SIN CONEXIÓN	HIGH	1/32 de paso

Fuente: Autor

- **SLP:** Sleep, esta entrada nos permite por programación habilitar o deshabilitar el funcionamiento del motor, es decir, si esta entrada se encuentra en HIGH entonces el motor entra a funcionar, caso contrario se encuentra desenergizado.
- **STEP:** Como su nombre lo indica, esta entrada nos permite controlar el paso en los motores, ya con el tipo de paso establecido por medio de M0 y M1, aquí simplemente se debe ingresar un tren de pulsos. Cada pulso que ingresa por STEP representa un paso del motor.
- **DIR:** Esta entrada nos permite controlar el sentido de giro de los motores, es decir entregamos una entrada lógica HIGH, el motor girará en un sentido y al cambiar a LOW, el motor girará en el sentido contrario.

En cuanto a las salidas tenemos:

- A1, A2, B1 y B2: Cada una de estas nos permite conectar a los cables de las bobinas de los motores. Este tipo de driver permite controlar motores de 4, 6 y 8 cables, para este proyecto se ha utilizado un motor de 4 cables
- FLT: Es una alarma de sobre corriente, al momento en que se detecta una sobre corriente en el circuito, esta salida es llevada a un estado lógico LOW.

En la figura a continuación se puede observar cómo se realiza la conexión de los drivers, los motores y el procesador.

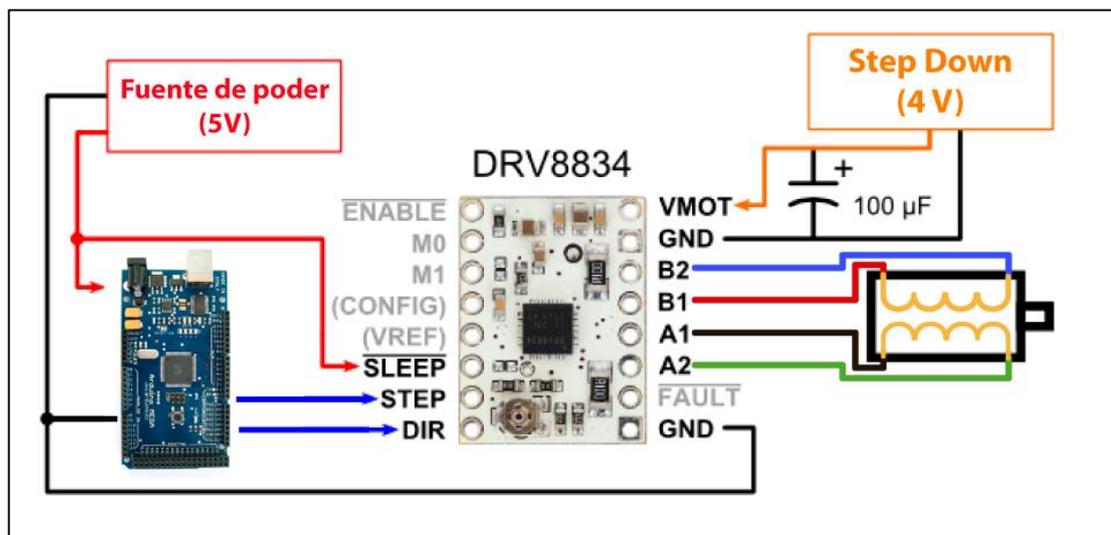


Figura 53 Esquema de conexión de Driver

Fuente: Autor

3.2.7 Baterías

Uno de los puntos interesantes que se han planteado en este proyecto, es la autonomía del robot, tanto de comunicación como de energía, evitando así el uso de cables e intentando quitar las limitaciones que estos implican, básicamente de movilidad; es

por esto que se ha decidido montar una fuente de energía (batería) en el robot que le permita tener una autonomía por el mayor tiempo posible.

Es muy importante contar con un dato aproximado, es decir un valor teórico del consumo de energía que se tendrá con el robot en funcionamiento, para esto se han realizado varias mediciones como se pueden ver a continuación:

Tabla 12 Consumo de energía de los dispositivos

Equipo	Consumo (mA)
Arduino Mega	500
Motores	2000 (X3)
Cámara	1600
Bluetooth (Shield)	50
Servos	600 (X2)
Leds	250
TOTAL	9600

Fuente: Autor

Teniendo en cuenta estos consumos, por medio de una búsqueda de proveedores nacionales e internacionales, se consiguieron baterías de marca Tenenergy, en este caso y según los consumos anteriormente expuestos la que mejor se acopla es la que entrega 7,2 V y 5 000 mAh de NiMH, son baterías recargables y son capaces de proveer una fuente de voltaje sin pérdidas o caídas a un consumo máximo. Con la suma de las 2 baterías tenemos como resultado 10 000 mAh, lo cual cubre la necesidad de corriente para el robot.



Figura 54 Batería recargable de NiMH Tenergy de 5 000 mAh.

Fuente: (Amazon, 2014)

Haciendo un análisis del consumo máximo y de la corriente por hora que nos entregan las baterías, se puede decir que se tendrá un tiempo de autonomía de 60 min en caso de que todos los elementos estén activos durante ese tiempo. Obviamente este no es un dato real, ya que el consumo dependerá de la inclinación de las tuberías, debido a que si los motores tienen un mayor esfuerzo, generarán un mayor consumo. Más adelante en el capítulo V se analizará estos consumos.

Con esto se presenta un nuevo elemento a controlar, debido a que no existen baterías del voltaje y el amperaje buscado, se debe acoplar las que se describen anteriormente al voltaje requerido. Para esto se utiliza el elemento que se describe a continuación.

3.2.8 Step down converter

Al tener baterías de 7,2V y tanto el Arduino como el driver trabajan a voltajes más bajos, se requiere algún tipo de circuito que sea capaz de bajar el voltaje sin comprometer el amperaje que nos entregan las baterías.

La primera opción y la más sencilla para reducir el voltaje de entrada sería un circuito divisor de voltaje como el de la imagen a continuación:

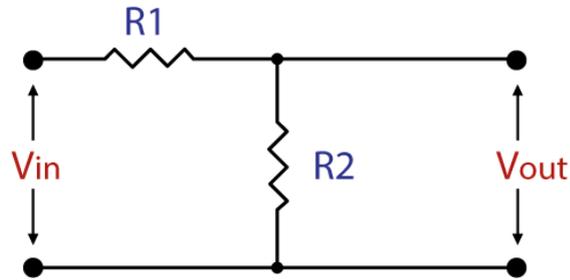


Figura 55 Circuito divisor de voltaje

Fuente: Autor

En este tipo de circuitos se puede calcular el voltaje de salida como se verá a continuación, utilizando la ley de ohm tenemos:

$$V_{out} = I \cdot R_2$$

Y

$$V_{in} = I \cdot (R_1 + R_2)$$

De donde:

$$I = \frac{V_{in}}{R_1 + R_2}$$

Reemplazando:

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2}$$

Por lo tanto si tenemos un voltaje de 7,2 V de ingreso y queremos obtener 4 V a la salida, solamente tendríamos que reemplazar en la fórmula anterior, también debemos suponer que la resistencia R_2 es de 1,2K Ω (valor propuesto).

Entonces tendríamos:

$$4V = 7,2V \frac{1,2K}{R_1 + 1,2K}$$

$$R_1 + 1K = \frac{7,2V}{4v} \cdot 1,2K$$

$$R_1 = 1,16K\Omega$$

Entonces el valor deseado de la resistencia R1 sería 1,16KΩ, que se podría obtener con un potenciómetro. Pero hay que tener en cuenta que esta opción no es la correcta, ya que en esta configuración se pierde corriente en forma de calor y por lo tanto no es eficiente. Es por esto que se necesitó desarrollar algún tipo de circuito con la eficiencia correcta para evitar pérdidas innecesarias, entonces se ha decidido utilizar la configuración denominada “Step down converter” o convertidor de potencia, muy conocida y capaz de realizar una conversión DC/DC, obteniendo un voltaje de salida menor al que ingresa, utilizando circuitos integrados que proveen un control adecuado de la corriente.

El funcionamiento básico es el siguiente:

La conexión del inductor es alternada entre la fuente de alimentación y la carga, esto es controlado por semiconductores. Inicialmente se han realizado pruebas con el integrado LM2576, el cual tiene como características específicas las siguientes:

- Salida ajustable en un rango de 1,23 a 37 voltios
- Corriente de salida de 3A
- Rango de voltaje de entrada de hasta 40V
- Solamente requiere 4 componentes externos
- Frecuencia fija del oscilador de 52kHz
- Alta eficiencia
- Protección de sobre corriente y temperatura

Este tipo de integrados son relativamente sencillos de utilizar, ya que como se mencionó anteriormente solo necesita una cantidad mínima de componentes. Además incluye una compensación interna de frecuencia y un oscilador de frecuencia fija. En la figura a continuación se puede observar la configuración interna:

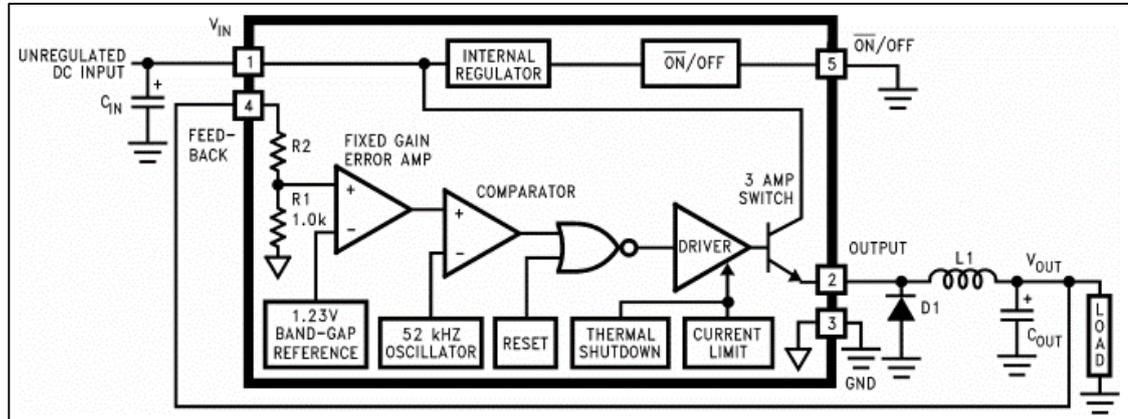


Figura 56 Configuración interna LM2576

Fuente: (Texas Instruments, 2013)

La ecuación principal para describir este proceso es la siguiente:

$$V_o = D \cdot V_i$$

En donde

V_o = voltaje de salida

D = ciclo de trabajo (nunca puede ser mayor que 1)

V_i = Voltaje de ingreso

Entonces se puede deducir que V_o nunca puede ser mayor que V_i de ahí de donde viene el nombre de la configuración.

En la figura a continuación podemos observar la forma de conexión más común o aplicación típica que tienen estos integrados, esta conexión permite que la energía se almacene en el inductor mientras el circuito está cerrado y cuando se abre esta se transfiere a la carga.

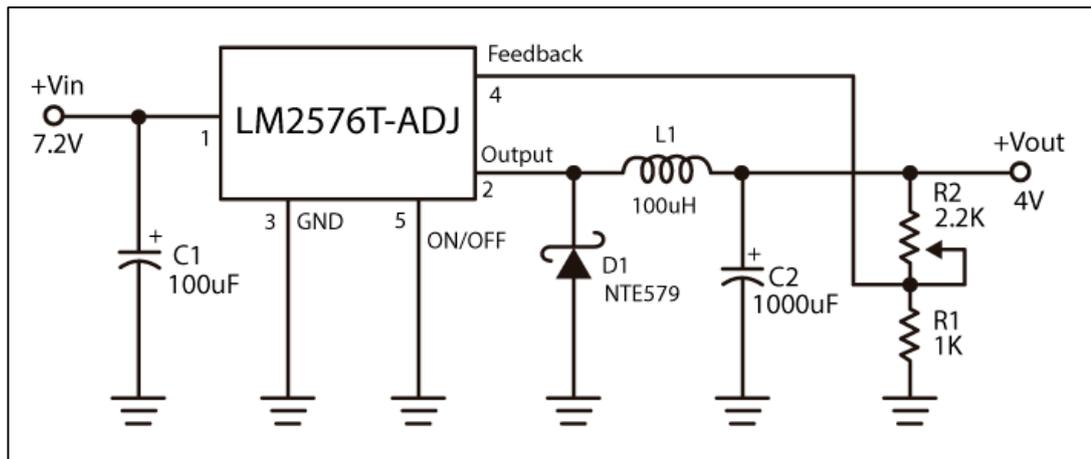


Figura 57 Aplicación típica de LM2576T-ADJ

Fuente: Autor

Al ser un circuito regulable, se debe calcular el valor de las resistencias R1 y R2 para obtener el voltaje de salida deseado. Para esto se utiliza las fórmulas:

$$V_{out} = V_{ref} \left(\frac{R2}{R1} + 1 \right)$$

$$R2 = \left(\frac{V_{out}}{V_{ref}} - 1 \right) R1$$

En donde:

$V_{ref} = 1,23V$ y R1 debe estar entre 1K y 5K. Para nuestros propósitos y tomando en consideración que los motores trabajan a 4V y el Arduino y los drivers trabajan en un rango de 3,3 y 5 V se ha calculado de la siguiente manera:

$$R2 = \left(\frac{4}{1,23} - 1 \right) 1k$$

$$R2 = 2,2k$$

Luego de realizar varias pruebas, se puede ver que los componentes adquiridos localmente no son los adecuados para realizar este tipo de configuración ya que el inductor no es apto para la cantidad de potencia que se maneja en este proyecto y luego de unos segundos de uso pierde su potencia siendo disipada en forma de calor. En la imagen siguiente se pueden ver las diferentes pruebas que se realizaron con el integrado LM2576 y se pudo verificar que hay una limitación de la corriente, ya que como se había visto anteriormente el motor requiere 1,2A por fase, 2,4A en total pero con esta configuración solamente se obtienen 1,8 A debido al inductor que se está utilizando, ya que al momento de entrar en funcionamiento se calienta y la corriente baja.

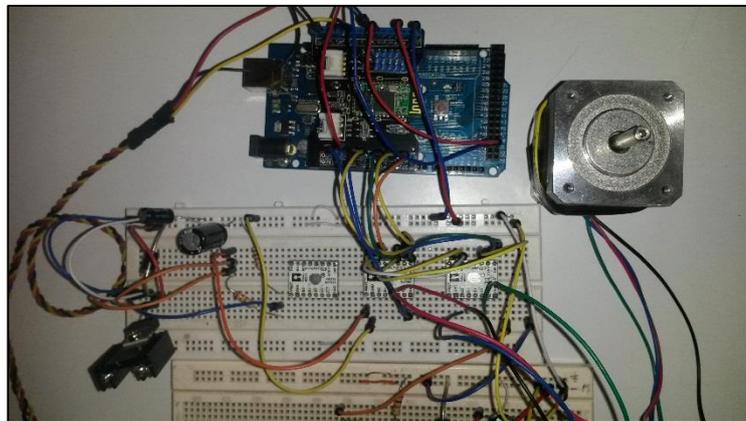
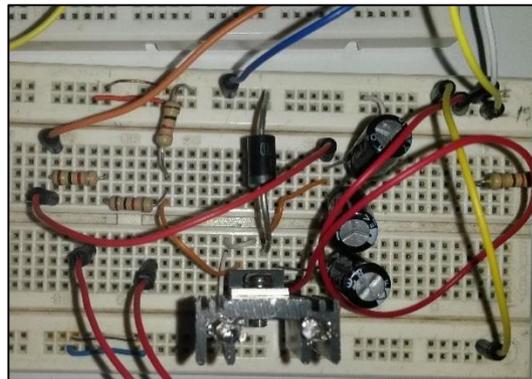


Figura 58 Conexiones de prueba para LM2576T-ADJ

Fuente: Autor

Debido a esta restricción se ha decidido adquirir un circuito *step down converter* fabricado comercialmente, que tenga un costo razonable y que cumpla con los requerimientos para este robot.

A continuación se tiene una tabla comparativa de las diferentes opciones:

Tabla 13 Comparación step dow converter

	Rango de voltaje entrada	Rango de voltaje salida	Corriente	Precio	Imagen
DROK® Adjustable Voltage Regulator Constant Voltage Regulate	4.0~40V	1.25V~3 7V	3A	\$5.19	
DROK® 75W 12V to 5V Volt Adjustable Voltage Regulator	5.0~36V	1.25~32 V	5A (Max 75W)	\$10.50	
DROK® LM2596 Adjustable DC Voltage Switching Regulator 4.0- 40V to 1.25-37V DC	4.0~40V	1.25~37 V	3A	\$7.70	

DROK® DCDC 12A 100W Adjustable DC Step Down Converter Battery Buck	4.5~32V	0.8~30V	12A (Max 200W con disipador)	\$15.45	
DROK® 12A/100W 4.5- 30V to 0.8-30V DC Buck Volt Converter	5.0~40V	1.25~36 V	12A (Max 200W con disipador)	\$17.89	

Fuente: Autor

Como se puede ver en la tabla anterior, los costos no son elevados en ninguno de los casos, pero los beneficios que nos ofrece la opción DROK® DCDC 12A 100W son los que se ajustan a las necesidades sobre todo de corriente, ya que con estos se podría alimentar sin ningún problema tanto a los motores, servos, cámara, luces etc.

Estos equipos funcionan con el mismo principio que se explicó en el punto anterior.



Figura 59 Step Down converter

Fuente: (AMAZON, 2014)

Con este dispositivo, quedaría controlado el tema del voltaje y la corriente, por lo que puede pasar al siguiente paso, una forma sencilla en la que el usuario podría manipular al robot.

Las opciones para manipulación del robot podrían ser muy amplias y abarcar desde el mismo teclado del computador, como una Tablet, un celular. Pero se ha visto que una manera sencilla de controlarlo sería utilizando un control DualShock3 debido a los varios sensores y fácil manejo que tiene.

Esto se describe a continuación.

3.2.9 Control DualShock3

Para controlar los movimientos tanto del robot como de la cámara se ha decidido utilizar un control DualShock3, el cual cuenta con varios mandos que pueden ser acoplados a las necesidades de movimientos del robot, estos son:

- 15 botones
- 2 joysticks análogos

- 2 botones sensibles a la presión
- sensor de movimiento de 6 ejes.
- La conexión al computador se puede realizar por medio de un cable USB o Bluetooth.



Figura 60 Control Dual Shock3

Fuente: (Playstation)

En este caso se usarán uno de los joysticks análogos para el control de los dos ejes de la cámara, los 2 botones sensibles a la presión que nos permitirán controlar la velocidad según la presión aplicada y el sentido de movimiento del robot, 2 botones para el control de encendido de la iluminación.

3.2.10 Control “MOTION JOY”

Para poder adaptar y obtener los datos y lecturas del control, necesitamos alguna forma de conexión del control con el computador, es por esto que se ha utilizado 2 herramientas muy importantes, estas son:

- **MotionJoy V0.7:** Software de código libre que permite la fácil conexión de un control de juego a un computador. El programa reconoce los botones digitales así como los análogos (sensores de presión, joysticks, giroscopios), proveyendo una lectura de los datos y permitiendo leerlos por medio de otros programas.
- **Driver MotionInJoy para usar un control Dualshock 3 en LabVIEW:** Es un programa desarrollado en LabVIEW para reconocer las señales enviadas

por el control y entregar en forma de carácter numérico o alfanumérico la señal decodificada.

En cuanto a la parte de programación, se debe realizar un esquema capaz de recibir la información enviada por el control para ser utilizada como comandos para los movimientos del robot. Es por esto que se tienen los subVIs de los cuales se explicará su funcionamiento.

- Un subVI para detectar el control conectado a MotionInJoy
- Un subVI para leer la información del control

3.3 CONFIGURACIÓN Y PROGRAMACION

3.3.1 Configuración cámara IP

Una cámara IP no es “*plug and play*”, debe ser configurada para conectarse a la red deseada mediante una conexión alámbrica, que luego podrá obviarse para iniciar la transmisión inalámbrica.

Para poder tener una conexión con la cámara se tiene que realizar los siguientes pasos, solamente la primera vez:

- Con el cable de red provisto, conectar la cámara al router WiFi
- Insertar el CD de instalación, en el *autorun* seleccionar “*Search IP Camera*” luego “*Advance mode*”
- En la casilla “*Equipments*” aparece la cámara IPCAM1, Copiar la dirección IP y el puerto que está por defecto como se ve en la figura a continuación

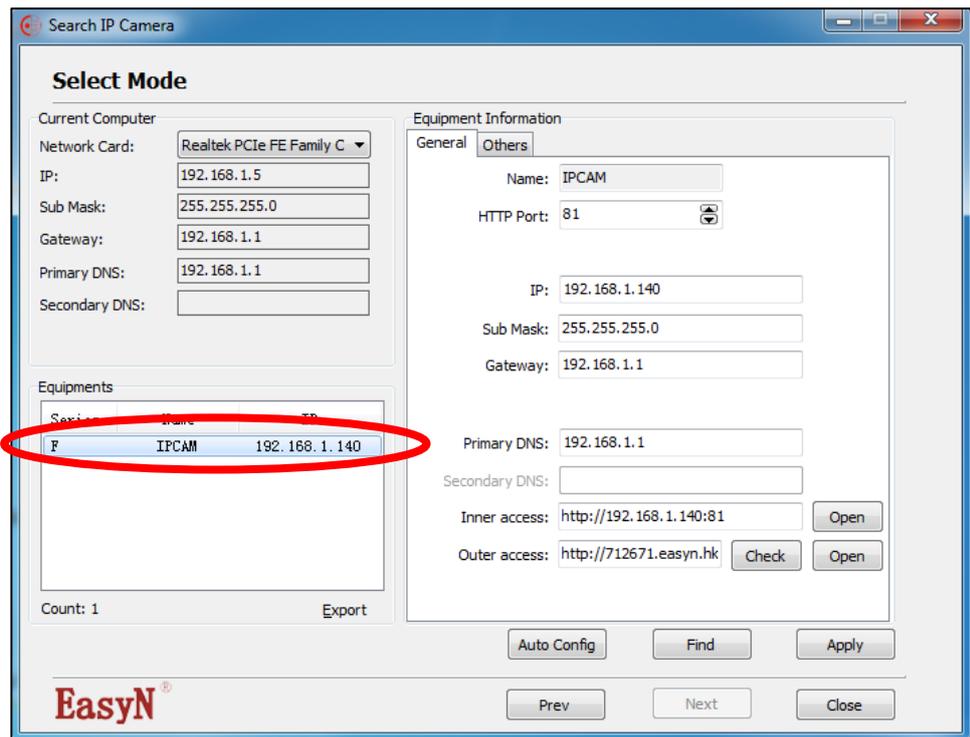


Figura 61 Selección de cámara IP para configuración

Fuente: Autor

- En un navegador escribir la dirección IP seguido dos puntos (:) y el puerto
- En la pantalla que se genera, Ingresar como usuario “admin”, contraseña en blanco
- Luego de haber validado la información, se ingresa a una página en donde se debe Seleccionar la segunda opción “*Server Push Mode*”
- Luego en seleccionar la pestaña “Opciones”
- En la pestaña Red Básica se debe ingresar la información como se ve en la figura 62

Red básica	
Obtener IP de servidor DHCP	<input type="checkbox"/>
Agrupar IP	192.168.1.140
Máscara de subred	255.255.255.0
Puerta de enlace	192.168.1.1
Servidor DNS	192.168.1.1
Puerto HTTP	81
<input type="button" value="Set"/> <input type="button" value="Actualizar"/>	

Figura 62 Configuración de IP para cámara

Fuente: Autor

- La dirección IP fija deberá estar en la misma red del router (en este caso en la red “1” siendo 192.168.1.140), de igual forma debe ser configurado el Gateway
- En la pestaña de usuarios, se puede cambiar el usuario y la contraseña para evitar el acceso de intrusos, en este caso se ha colocado como usuario “tesis” y la contraseña “tesis”
- Hacer Click en Set y la cámara se reiniciará para aplicar los cambios efectuados
- Volver a ingresar a la cámara como se indicó anteriormente y elegir la pestaña LAN inalámbrico
- Ingresar la misma configuración de seguridad inalámbrica del router, como se ve en la figura 63

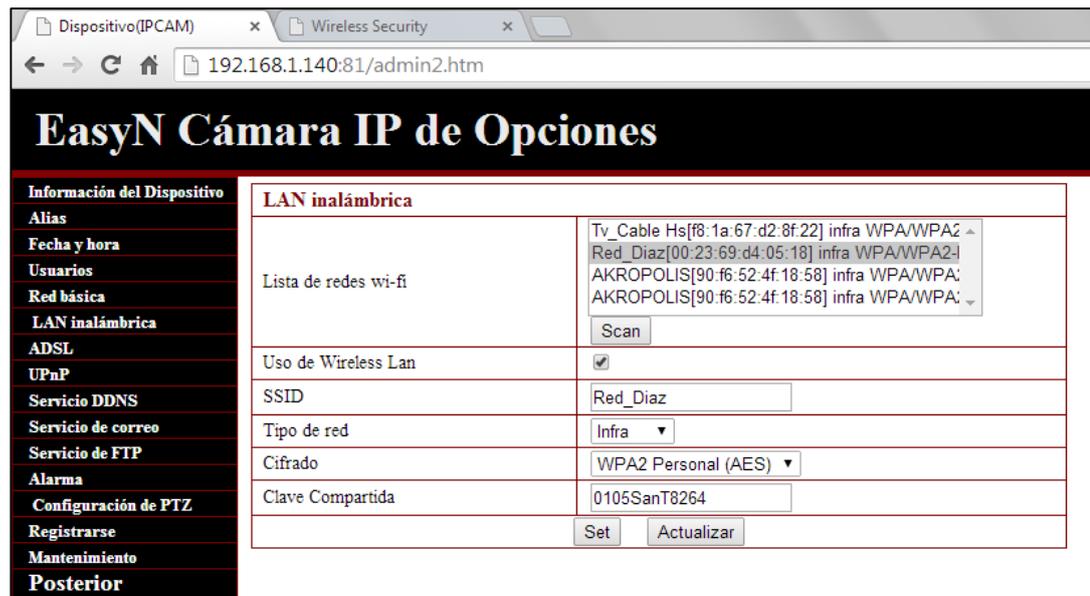


Figura 63 Configuración de LAN Inalámbrica

Fuente: Autor

- Hacer click en Set nuevamente y esperar a que la cámara se reinicie para que se apliquen los cambios
- Finalmente desconectar el cable y comprobar la recepción de video por WIFI, ingresando a la cámara por medio de un navegador como se indicó anteriormente.

Para poder acoplar la cámara al robot ha sido necesario desarmar el dispositivo y seleccionar únicamente los componentes necesarios para evitar peso y espacio innecesario. Por lo que la estructura final de la cámara para ser acoplada ha quedado como se puede ver en la figura a continuación:

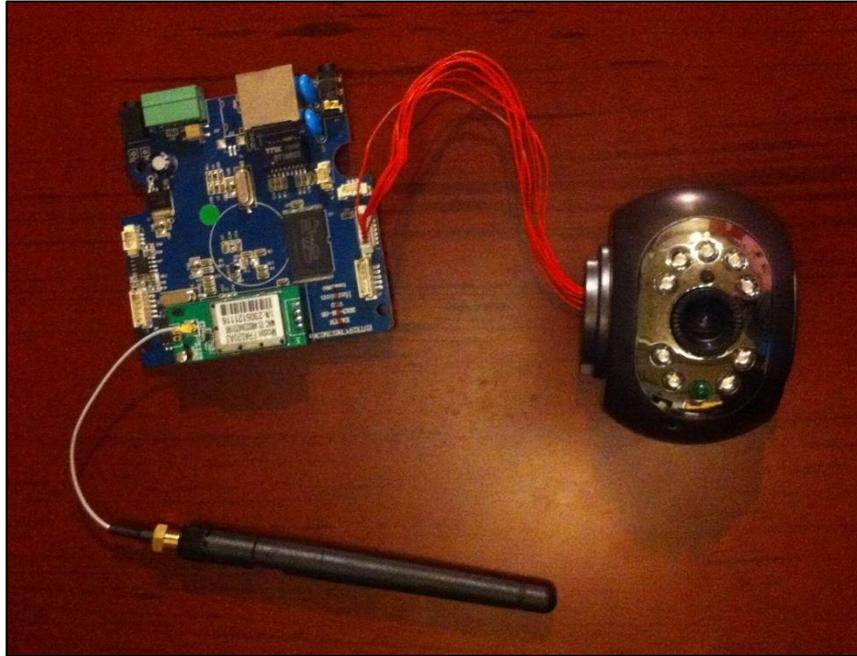


Figura 64 Circuito principal de cámara IP, antena, lente con leds infrarrojos y sensor de luz

Fuente: Autor

3.3.2 Desarrollo y programación Arduino

Como explicamos anteriormente, Arduino es una herramienta muy útil y sencilla de utilizar. Igualmente se había explicado que el entorno de desarrollo IDE (*Interactive Development Environment*) que usa está basado en Processing (basado en Java), que según lo describen en su sitio oficial es “Un lenguaje de programación, entorno de desarrollo y comunidad online” (Processing). Inicialmente fue desarrollado para diseñadores gráficos o artistas, lo que les permitía desarrollar proyectos de manera muy simple.

Un entorno de desarrollo integrado, es una aplicación que cuenta con las siguientes características básicas:

- Depurador
- Editor de código
- Instrumentos de programación

Además debe ser capaz de facilitar el trabajo de programación de nuevos proyectos a los desarrolladores. Todas estas características están para cumplir un objetivo principal, que es el de reducir el tiempo invertido en configuraciones, permitiendo obtener programas de muy buen nivel en un tiempo mucho más corto.

Processing ha dado lugar a otro proyecto, Wiring, que utiliza el IDE de Processing con una versión simplificada del lenguaje C ++. En base a estos se ha desarrollado el IDE para Arduino.

En la página de Arduino se puede encontrar un pequeño cuadro comparativo para ver las diferencias que existen entre los lenguajes de programación:

Tabla 14 Comparativo entre Arduino y Processing

Arduino	Processing
Matrices (Array)	
Int bar[8];	int[] bar = new int[8];
bar[0] = 1;	bar[0] = 1;
int foo[] = { 0, 1, 2 };	int foo[] = { 0, 1, 2 };
Bucles	
int i; for (i = 0; i < 5; i++) { ... }	for (int i = 0; i < 5; i++) { ... }
Impresión	
Serial.println("hola mundo");	println("hola mundo");
int i = 5; Serial.println(i);	int i = 5; println(i);
int i = 5; Serial.println(); Serial.print("i = "); Serial.println(i); Serial.println();	int i = 5; println("i = " + i);

Fuente: Arduino, 2015

A continuación se hará un breve resumen de las características del IDE que se puede encontrar en la página oficial de Arduino (Arduino, 2015).

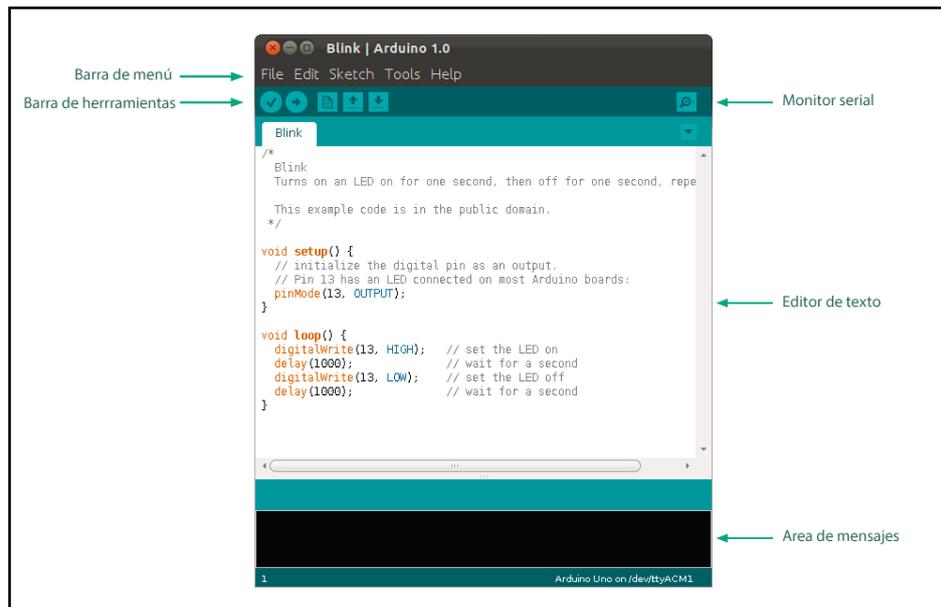


Figura 65 Entorno de desarrollo Arduino (IDE)

Fuente: Autor

El software que se desarrolla en Arduino en el área del editor de texto se denomina Sketch, estos se guardan con la extensión .ino. Este entorno está dividido como se ve en la Figura anterior en las siguientes áreas principales:

- Editor de texto
- Área de mensaje y consola de texto: entrega un *feedback* de los errores que se dan al momento de guardar o exportar, etc.
- Barra de menús: tenemos los menús comunes tales como Archivo, Editar, Sketch, Herramientas (Permite elegir la tarjeta, el puerto y el programador que se va utilizar) y por último el menú de Ayuda
- Barra de herramientas con los botones de funciones más comunes, estas son:



Verificar: Comprueba errores en el código



Cargar: Compila y carga el programa al hardware.



Nuevo: Crea un Nuevo sketch.



Abrir: Abre un archive existente.



Guardar: Guarda el sketch actual.



Monitor Serial: Abre el monitor del Puerto serial para verificar los datos que se están enviando y recibiendo.

En estos entornos también cuentan con librerías que proveen funcionalidades adicionales a nuestro Sketch. Son programas ya desarrollados previamente para cumplir un propósito específico, estos pueden ser incluidos a un nuevo sketch con el comando:

```
#include <Librería.h>.
```

Incluir estas librerías va a incrementar el tamaño del programa debido a que se carga conjuntamente con el sketch que hemos desarrollado.

Algunas de las librerías son incluidas con el software de Arduino aunque también se pueden desarrollar librerías propias, es muy simple y se pueden conseguir tutoriales en cualquier foro.

Ahora que hemos revisado los conceptos básicos, revisaremos los esquemas de programación y funciones necesarias para el desarrollo de un proyecto nuevo.

Todo programa de contar con 2 partes fundamentales:

- *Setup* (argumentos): En esta primera función se configurarán todos los programas necesarios para el funcionamiento de la tarjeta Arduino, se puede definir un pin tanto de entrada como de salida, establecer velocidad del puerto serial, inicializar variables, asignar nombres a los puertos, etc. Esta primera función se ejecuta solo una vez como configuración inicial al encender o reiniciar la tarjeta.
- *Loop* (programación): Luego de que se ha configurado el microcontrolador con sus puertos y variables se necesita entrar en el bucle de funcionamiento infinito. Es aquí donde se debe colocar la programación que se va a ejecutar una y otra vez para cumplir el objetivo del proyecto.

En el caso de este proyecto se tienen diferentes subprocesos dentro del *loop* principal para cumplir los distintos propósitos. Estos serían:

- Control de motores de pasos: Para este control se utiliza la librería *Stepper.h*, a continuación se definen las variables a utilizar estas almacenaran el comportamiento de los motores. Como se indicó anteriormente el driver nos permite controlar la dirección, el paso y activación/desactivación del motor, por lo que se deben establecer 4 pines de salida digital (dirección, M0 y M1, *Enable*). También la librería *stepper* nos permite establecer los puertos de salida para las bobinas del motor, en este caso solamente se necesita un tren de pulsos para activar un paso con el driver. Esto se realizará con el puerto 8 de la tarjeta y con la ayuda de la función *myStepper.setSpeed (valor)*.
- Control Servo motores para Visión: En este proceso se deben controlar los 2 servomotores que permiten la movilidad de la cámara en 2 ejes. Para esto utilizamos la librería *Servo.h*, con esta librería se debe definir el puerto en el que se va a manipular el servo motor, es muy importante tener en cuenta que el puerto que se utilice debe tener capacidad PWM que en el Arduino Mega estos puertos van del 2 al 13, en este caso se han elegido los puertos 3 y 5. Primero se debe definir el la variable con el nombre del servomotor, para esto se utiliza el comando “*Servo variable;*” y para establecer el puerto de esta variable se utiliza “*variable.attach (número del puerto);*” por último para establecer el ángulo al que se debe mover el servomotor se utiliza el comando “*variable.write (valor del ángulo);*”.
- Conexión Bluetooth: Este control es fundamental para establecer la conexión entre el Arduino y el computador. Para este proceso se ha utilizado la librería *SoftwareSerial.h*, la que nos facilita la comunicación serial con el *shield* para Arduino que se explicó anteriormente. Este punto es más complejo que los anteriores

Todas estas rutinas y detalles de la programación se pueden ver en el Anexo 25

3.3.3 Configuración de conexión Control DualShock3 – Computador

Mediante esta conexión podremos recibir datos del periférico de control de movimiento del robot.

El proceso de instalación y reconocimiento del control en el computador es el siguiente:

- Descargar e instalar MotionInJoy, el cual está disponible de manera libre en la página <http://www.motioninjoy.com/download>, aquí se debe seleccionar el programa correcto para el sistema operativo que se esté utilizando.



Figura 66 Página de Descargas para MotionJoy

Fuente: Autor

- Abrir la aplicación si no se ha ejecutado automáticamente e instalar el programa siguiendo los pasos.

- Cuando se haya terminado de instalar buscar el ícono de acceso directo e ingresar al programa, como se ve en la figura:



Figura 67 Pantalla principal de configuración de Control Dual Shock3

Fuente: Autor

- Conectar el control con cable USB y esperar que Windows instale los drivers necesarios. Cuando el control ha sido reconocido por el computador, aparecerá el nombre en la pestaña que se muestra en la figura:

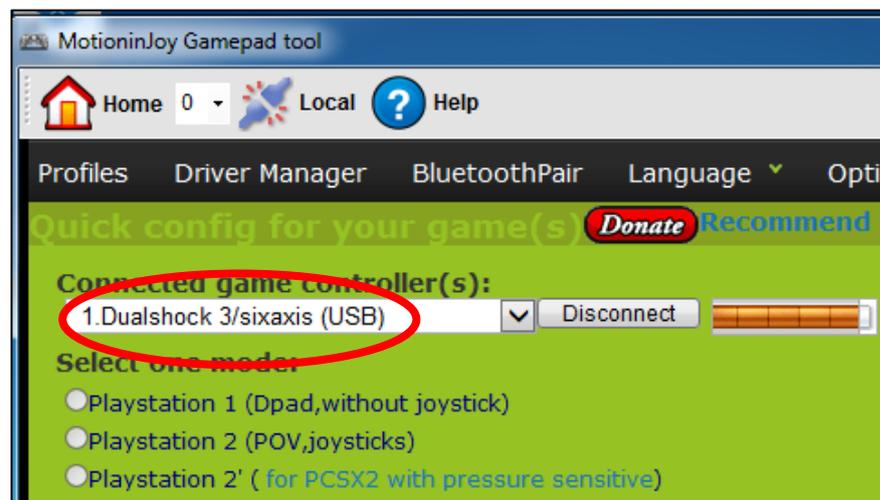


Figura 68 Selección de Control a configurar

Fuente: Autor

- Para asegurarse que los drivers fueron instalados se puede ir a la pestaña “driver manager”, se selecciona el hardware correspondiente al control y se seleccionar “load driver”

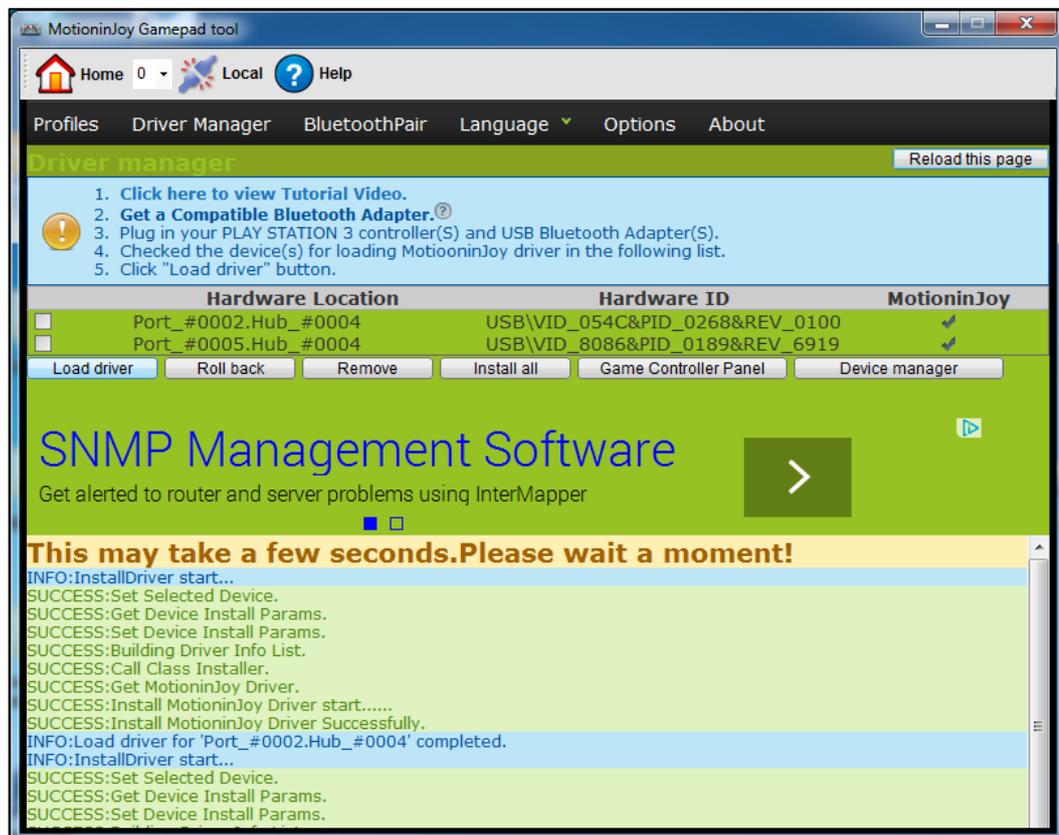


Figura 69 Instalación de Drivers para control MotionJoy

Fuente: Autor

- A continuación se debe regresar a “profiles” y seleccionar control dualshock3 (puede ser necesario desconectar y reconectar el control USB).
- Luego se debe verificar que la forma de leer los controles esté configurada de la manera que necesitamos, para esto se debe verificar en PlayStation 3 debajo de “Select One Mode”, seleccionar la casilla y luego dar click en opciones para editar los controles.

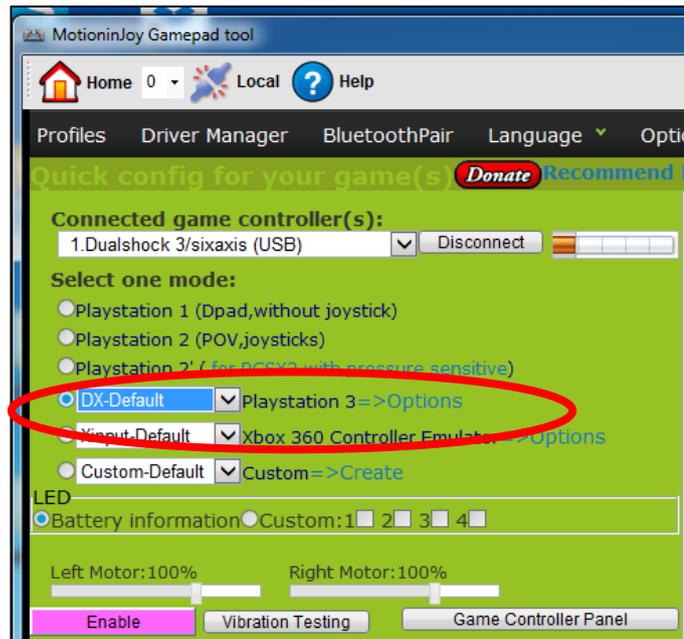


Figura 70 Selección de Control a Configurar

Fuente: Autor

- Editar los controles para que estén de acuerdo a las figuras a continuación, que según hemos determinado es lo que se acopla a las necesidades del robot y seleccionar guardar cambios:

DS3/Sixaxis	Dxinput	Auto	Turbo	Macro
Triangle	Button 1	<input type="checkbox"/>	<input type="checkbox"/>	
Circle	Button 2	<input type="checkbox"/>	<input type="checkbox"/>	
Cross	Button 3	<input type="checkbox"/>	<input type="checkbox"/>	
Square	Button 4	<input type="checkbox"/>	<input type="checkbox"/>	
L1	Button 5	<input type="checkbox"/>	<input type="checkbox"/>	
R1	Button 6	<input type="checkbox"/>	<input type="checkbox"/>	
L2	Button 7	<input type="checkbox"/>	<input type="checkbox"/>	
R2	Button 8	<input type="checkbox"/>	<input type="checkbox"/>	
SELECT	Button 9	<input type="checkbox"/>	<input type="checkbox"/>	
L3	Button 10	<input type="checkbox"/>	<input type="checkbox"/>	
R3	Button 11	<input type="checkbox"/>	<input type="checkbox"/>	
START	Button 12	<input type="checkbox"/>	<input type="checkbox"/>	
PS	Button 13	<input type="checkbox"/>	<input type="checkbox"/>	
D-pad up	D-pad up	<input type="checkbox"/>	<input type="checkbox"/>	
D-pad right	D-pad Right	<input type="checkbox"/>	<input type="checkbox"/>	
D-pad down	D-pad down	<input type="checkbox"/>	<input type="checkbox"/>	
D-pad left	D-pad left	<input type="checkbox"/>	<input type="checkbox"/>	
Triangle	None	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 71 Configuración de Control

Fuente: Autor

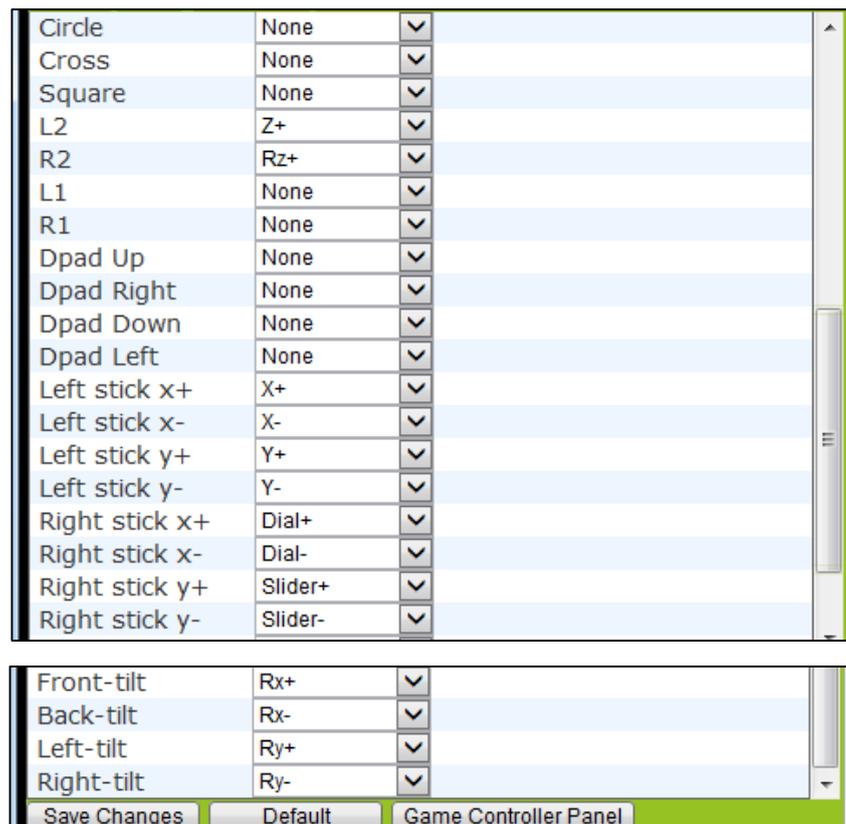


Figura 72 Parámetros de configuración de control DualShock3

Fuente: Autor

Al seguir estos pasos se ha configurado la manera en la que el computador va a recibir los datos por parte del control, cabe recalcar que esta configuración sólo se debe realizar la primera vez que se conecta el control al computador, para las demás conexiones simplemente se debe conectar y las configuraciones se mantendrán.

3.4 DISEÑO DE PCB

3.4.1 Altium Designer

Para el diseño y modelado del circuito impreso se ha utilizado el software ALTIUM DESIGNER, programa que inició en el año 1985 como uno de las primeras herramientas para diseñar PCBs.

Se ha elegido este programa por su excelente diseño asistido por computadora de placas de circuito impreso, que integra la mayor cantidad de herramientas necesarias en una sola interfaz, permitiendo reducir tiempos de diseño. Este programa es elegido por muchos profesionales del diseño de PCBs para crear dispositivos electrónicos para el entretenimiento, industria, defensa, educación, etc. Además cuenta con gran cantidad de librerías de componentes y sus huellas (*footprints*). También tiene potentes herramientas que permiten crear y diseñar nuevos componentes y huellas para acoplarse de mejor manera al diseño que se esté realizando.

Las características básicas del programa que se pueden encontrar en la página inicial son las siguientes:

- Interfaz de diseño y captura
- Diseño físico de un PCB
- Diseño de hardware para FPGA
- Implementación de sistemas FPGA y depuración
- Desarrollo de software embebido
- Simulación de circuitos
- Fabricación de PCB

Para tener un entendimiento básico del entorno de trabajo, se describirá las herramientas principales:

- **MENÚ DEL SISTEMA:** Accede a las características incluyendo el entorno de preferencias e información del servidor
- **MENÚS/BARRA DE HERRAMIENTAS/ATAJOS:** Recursos que cambiando de acuerdo al documento activo
- **NAVEGACIÓN:** Provee controles para desplazarse entre documentos particulares vistos hacia adelante o atrás, además de acceder a la página de inicio
- **PESTAÑAS DE DOCUMENTOS:** Cada documento abierto tiene su propia pestaña que se visualizará en esta área.

- VENTANA DISEÑO PRINCIPAL: Muestra y ordena documentos abiertos
- PANELES DE TRABAJO: Varios paneles proveen funcionalidad específica a un editor en particular.
- PANEL DE ACCESO: Los paneles de trabajo se acceden por medio de estas opciones.

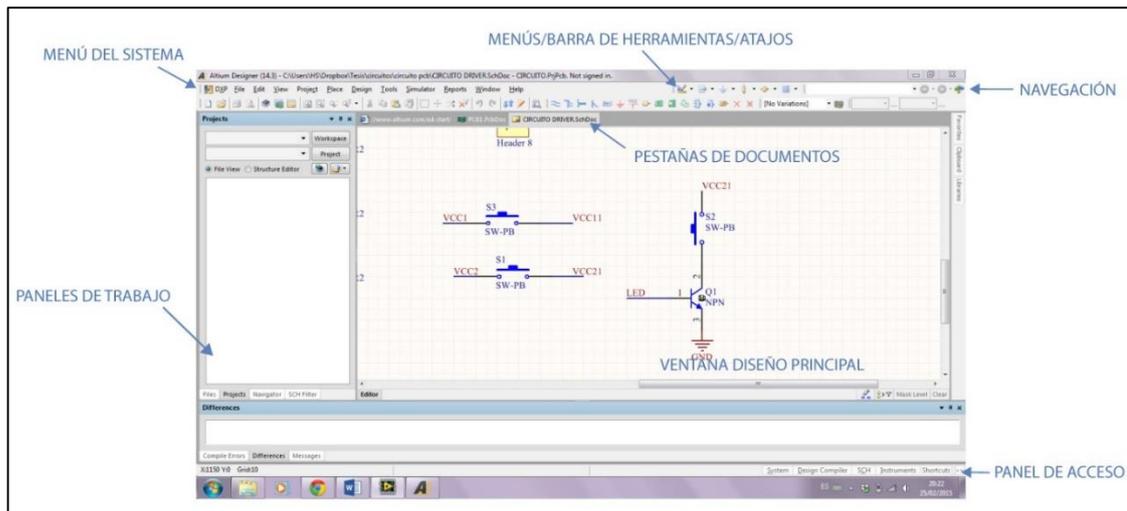


Figura 73 Interfaz ALTIUM DESIGNER

Fuente: Autor

3.4.2 Elaboración del circuito

Para diseñar el PCB se usaron las herramientas que se explicaron anteriormente, primero se realizó el diseño esquemático del circuito.

Al usar un driver particular como el DRV8834, no existe un *footprint* creado por lo que se debió hacer este proceso.

Para crear un nuevo componente se deben seguir los siguientes pasos:

1. Crear una nueva Librería

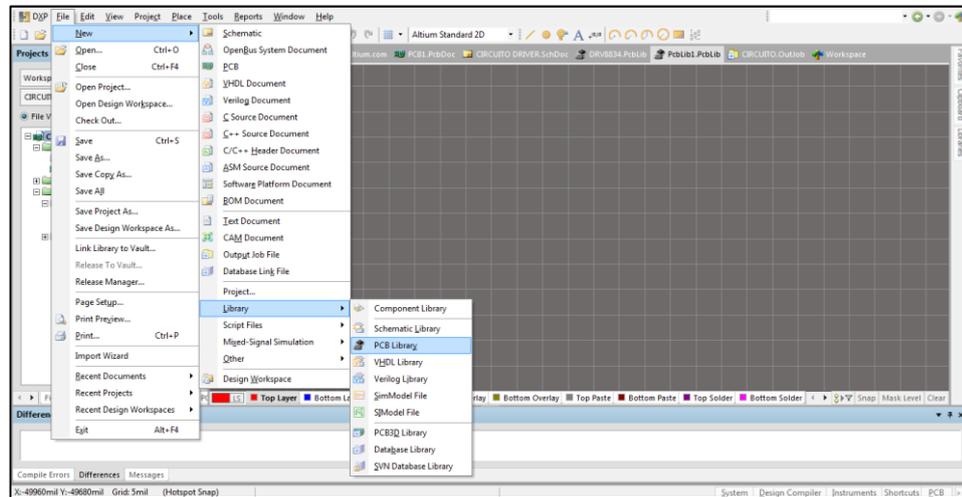


Figura 74 Menú para creación de Librería

Fuente: Autor

2. Por medio del asistente de creación de componente se crea la huella

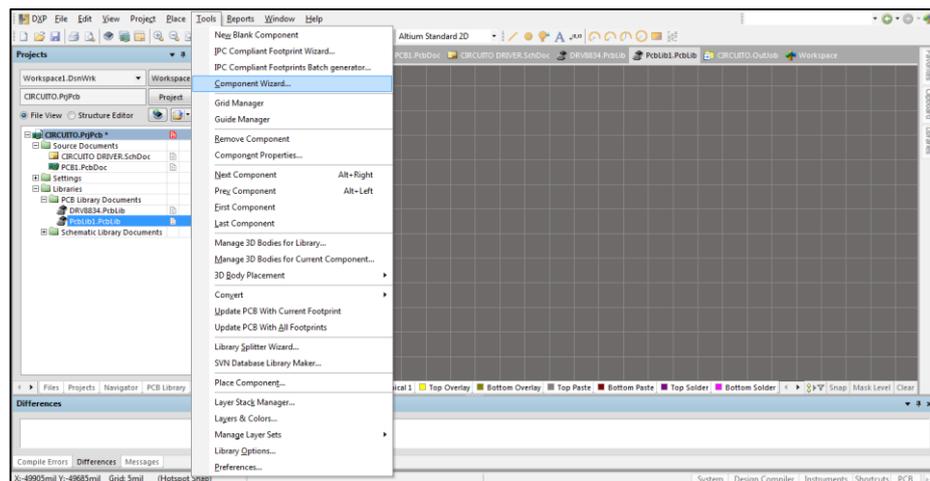


Figura 75 Menú para creación de componente

Fuente: Autor

3. Seguir el asistente para creación de componentes y llenar la información según la hoja de datos. Aquí se configura el tamaño, número y distancia entre Pads. Al igual que el nombre y tipo del componente (SMT o *Through hole*)

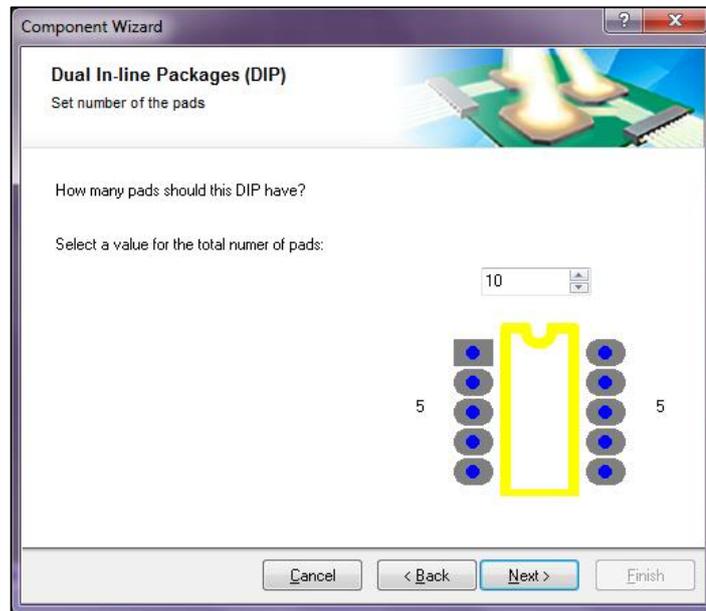


Figura 76 Asistente para creación de componentes

Fuente: Autor

Entonces se obtiene la librería como la imagen que sigue:

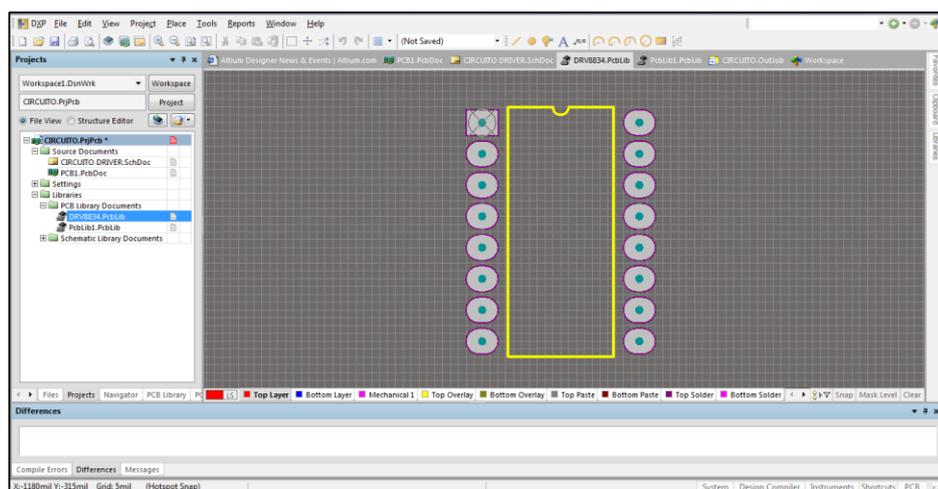


Figura 77 Librería de driver DRV8834

Fuente: Autor

Ahora que tenemos creado el *footprint*, también podemos crear el esquemático del componente, para esto se deben seguir los siguientes pasos:

1. Crear una nueva librería esquemática

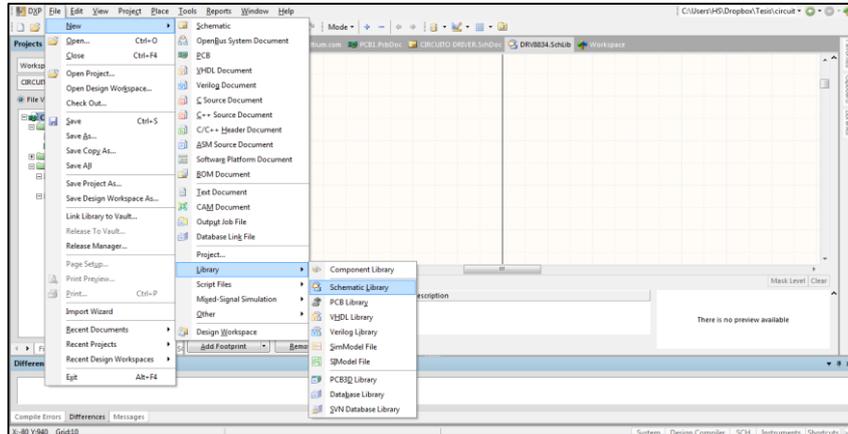


Figura 78 Menú para creación de librería esquemática

Fuente: Autor

2. Se debe crear el gráfico que se va a utilizar dentro del esquemático, para esto usamos la herramienta de formas básicas para colocar un cuadrado, círculos, arcos, etc. También nos permite colocar los pines correspondientes y configurar su nombre y número el tipo de configuración eléctrica (input, output, etc.), según la hoja técnica

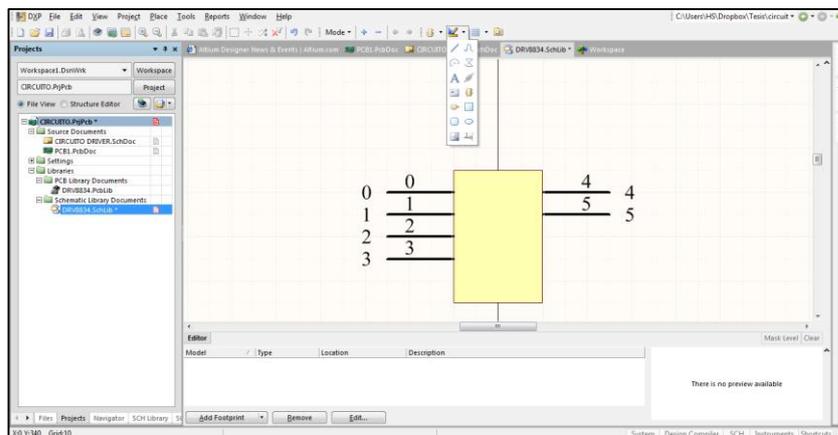


Figura 79 Creación de componente en librería esquemática.

Fuente: Autor

3. Para vincular la librería esquemática con la del *footprint* se puede usar el menú que se ve en la imagen a continuación. Y se localiza el .PcbLib que se creó anteriormente.

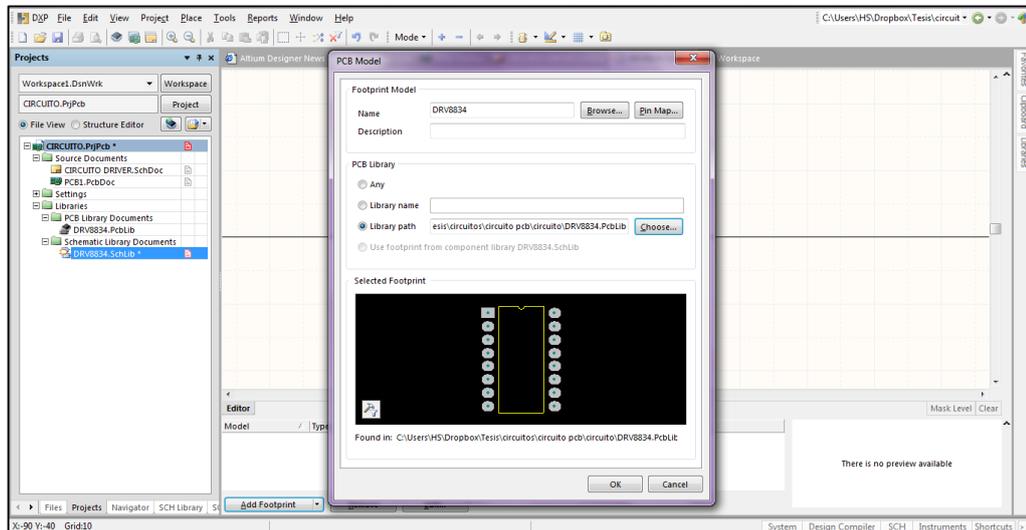


Figura 80 Vinculación de librería esquemática con footprint.

Fuente: Autor

Ahora que se tiene creado tanto el esquemático como el *footprint*, se puede crear ya el circuito.

Para esto se ha tomado en cuenta los siguientes parámetros:

- El control de los 3 motores, cada uno de ellos debe contar con su propio driver, estos drivers van conectados a las salidas correspondientes del Arduino.
- El control de los servomotores, que entregan 2 grados de libertad a la cámara, cada uno de ellos también debe ir conectado a una salida del Arduino
- Una salida adicional debe controlar el encendido de los Leds, por medio de un transistor
- Se deben conectar a 2 entradas diferentes la salida de las baterías para controlar el voltaje de cada una de ellas
- Se debe colocar borneras para la conexión del voltaje de entrada de las 2 baterías

- Para tener un control del encendido del robot, se debe colocar un *switch*
- Se necesitan borneras de salida para las bobinas de los motores
- Algo muy importante a considerar es que para acoplar esta placa de control al Arduino y al Shield se necesita considerar las separaciones estándar que ellos maneja, para colocar peinetas y acoplar perfectamente todos los circuitos.

Al realizar el circuito con todos los parámetros antes mencionados se obtiene el circuito del Anexo 24.

Con este circuito se envió a fabricar la placa de doble lado y se obtuvo el PCB de la imagen a continuación:

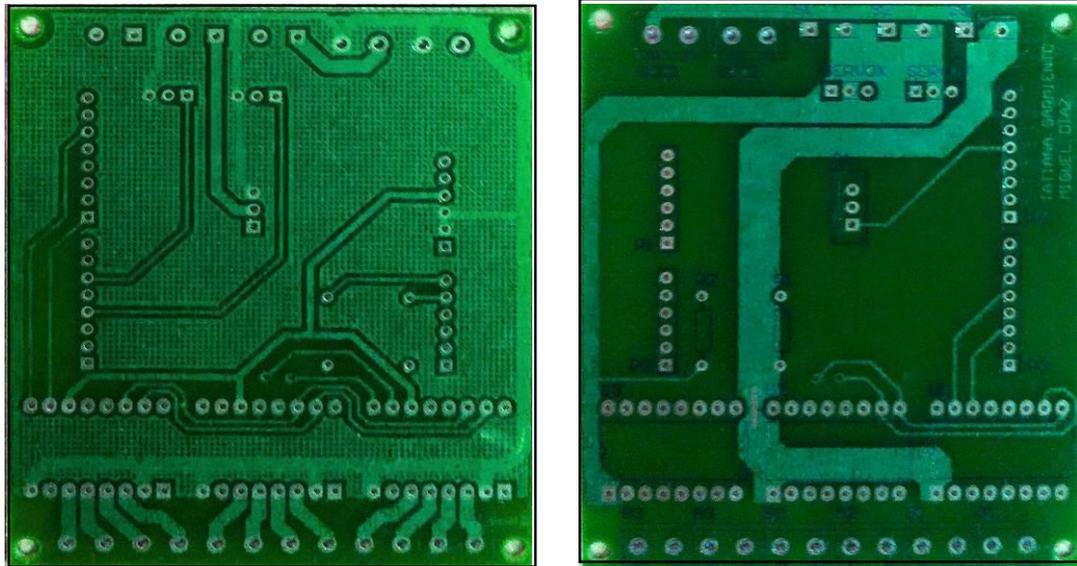


Figura 81 Capas PCB

Fuente: Autor

CAPÍTULO IV

DISEÑO DE LA INTERFAZ DE USUARIO

4.1 INTRODUCCIÓN

El diseño de la interfaz de usuario en este y muchos otros prototipos es un parte muy importante para el desarrollo del robot. Esta interfaz se ha realizado mediante LabVIEW, el que nos permitirá recibir la señal de los diferentes sensores al igual que las imágenes capturadas por el explorador.

Además, se debe tener en cuenta que la interfaz de usuario debe ser amigable para poder utilizar de una manera sencilla y por cualquier persona.

4.2 LabVIEW

La interfaz y procesamiento de datos en el computador está desarrollado en un plataforma muy útil y conocida, esta es LabVIEW que en inglés sus siglas hacen referencia a *Laboratory Virtual Instrumentation Engineering Workbench*, creada por National Instruments.

La programación es gráfica, esta se desarrolla en un entorno denominado VI o instrumento virtual, reciben este nombre debido a que inicialmente su principal objetivo era crear sistemas de medición electrónica. Actualmente se tienen aplicaciones mucho más amplias y entre ellas tenemos instrumentación, automatización, control, comunicaciones, entre otros. El lenguaje utilizado para programación es conocido como Lenguaje G, debido a que es desarrollo gráfico.

Estos bloques gráficos, permiten conectar entradas, salidas, parámetros de configuración, tiempos, etc. Para facilitar la programación y organización de los programas, también es posible crear SubVI's, que son subprogramas creados en un VI en insertado dentro del VI principal. (SHIRALKAR & National Instruments, 2007)

Para familiarizarnos con el entorno de programación se realizará una breve explicación:

4.2.1 Paneles

- **Front Panel:** Panel frontal, en este panel se desarrolla todo el diseño de interfaz del usuario, aquí se pueden visualizar todas las entradas y salidas disponibles. Dentro de este panel se tiene también un menú muy importante que es el de Conexión. Esto nos permite colocar las entradas y salidas que queremos que se nos genere para utilizarlo como SubVI, de esta manera se crearía un bloque para la programación.
- **Block diagram:** Diagrama de bloques, este panel contiene todo el código y procesamiento que se requiere para realizar una tarea específica, realizado de manera gráfica como se indicó anteriormente.
- **Menús:** Se pueden tener menús específicos para cada aplicación: Indicadores (Indicadores numéricos, led, booleanos, gráficos, etc), Estructuras, Arreglos, Operaciones numéricas, I/O, Procesamiento de señales, Visión artificial, entre otros.

4.2.2 Controles

- *Run:* Ejecuta una sola vez el programa



Figura 82 Opción RUN de un VI

Fuente: Autor

- *Run Continuously*: Ejecuta el programa continuamente

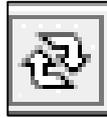


Figura 83 Opción Run Continuously de un VI

Fuente: Autor

- *Abort*: Detiene el Programa



Figura 84 Opción Stop de un VI

Fuente: Autor

- *Pause*: Pausa el programa en la interacción actual

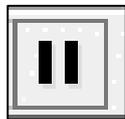


Figura 85 Opción Pause de un VI

Fuente: Autor

4.2.3 Datos

Como en varios lenguajes de programación, existen varios tipos de datos. Los que se usan generalmente son: Booleano (verdadero o falso), Numérico (Entero, Flotante, Complejos) *String* (texto).

4.2.4 Estructuras

Las estructuras nos permiten de alguna manera controlar el flujo en que se desarrollará el programa, establecer condiciones de ejecución, y tiempos.

Los principales y más usados son:

Bucle for: realiza una acción un número N-1 de veces establecido en la entrada N, se puede obtener el número de iteración de la salida i.

Bucle while: Realiza una acción mientras se cumpla una condición, al momento en que la condición deja de cumplir se sale del bucle y se continúa con la programación fuera de ella.

Case structure: Estructura de caso, nos permite seleccionar la acción a realizar en caso de cumpla una condición, el número de casos depende de las necesidades. Los casos por defecto son booleanos, pero por programación se puede hacer ingresar cualquier tipo de valor, ya sea numéricos, booleanos, *string*, etc.

4.2.5 Funciones

- **Funciones matemáticas:** Permiten realizar todo tipo de operación matemática.
- **Funciones booleanas:** Permiten realizar operaciones básicas
- **Funciones de comparación:** Se puede usar con cualquiera de los datos antes mencionados y el resultado siempre será un valor booleano.
- **Función de navegador web:** Permite conectarse a una dirección IP, la cual en este caso será muy útil ya que nos permitirá conectar la cámara web y recibir las imágenes.

Estas se pueden encontrar dentro del menú .NET, para poder configurar la cámara de manera correcta, se tiene que especificar la dirección IP como se ve en la figura a continuación.

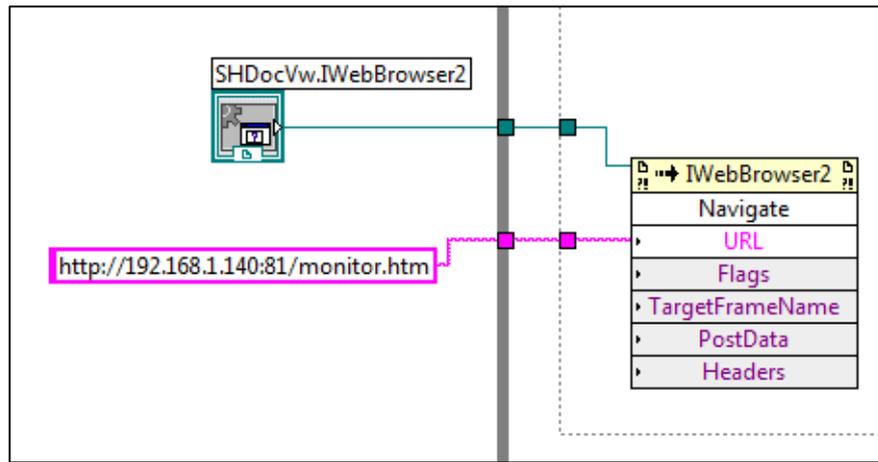


Figura 86 Configuración URL de cámara IP

Fuente: Autor

4.3 COMUNICACIÓN CÁMARA IP

Para la transmisión de video se optó por embeber un navegador de internet en una ventana en el panel frontal para facilitar la comunicación debido a que la cámara transmite por el protocolo IP.

El siguiente diagrama de bloques realiza la comunicación con la cámara, a continuación se detalla su funcionamiento.

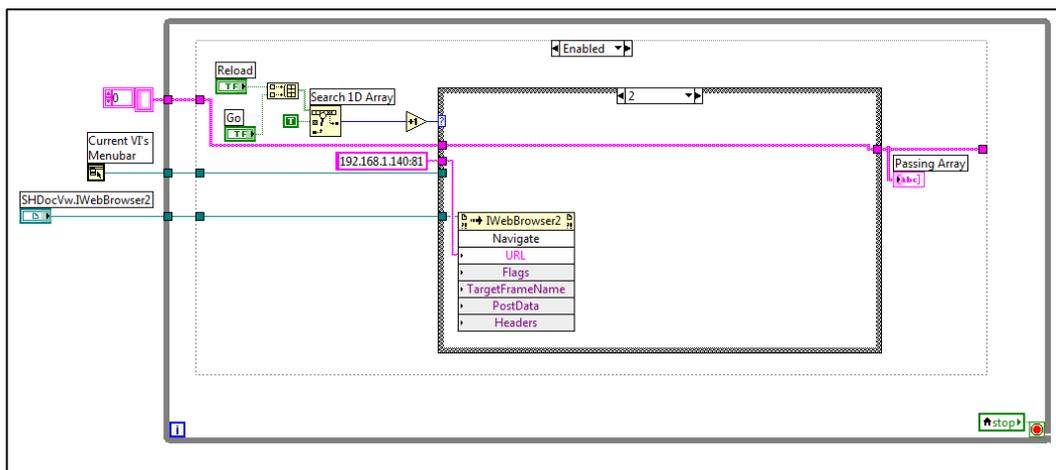


Figura 87 Diagrama de bloques comunicación cámara IP

Fuente: Autor

Para establecer la conexión con la cámara se ha decidido colocar 2 botones, uno de *Start* (Inicio) para iniciar a recibir datos de la cámara y otro de *Reload* (Recargar) para volver a cargar la conexión en caso de que se pierda la comunicación.

El siguiente diagrama reconoce el botón que se ha presionado en el panel frontal ordenándolos en un arreglo devolviendo valores 0 ó 1, finalmente al valor se suma uno para que no sea confundido por el valor default del *case structure*.

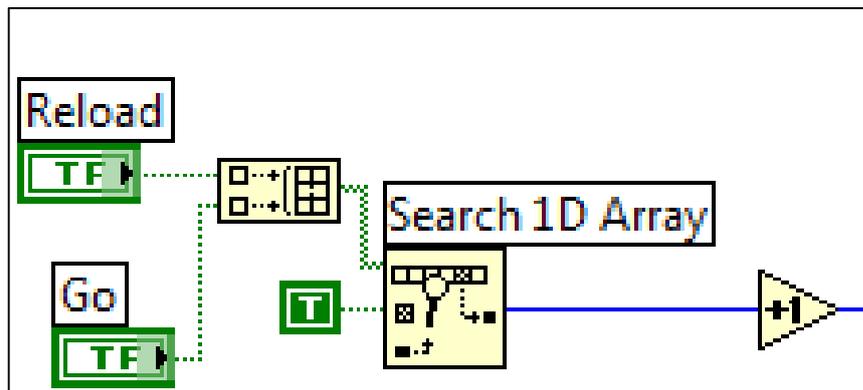


Figura 88 Reconocimiento de botones

Fuente: Autor

En la comunicación existen los siguientes casos que se encuentran explicados con una imagen respectiva:

- 0, Default

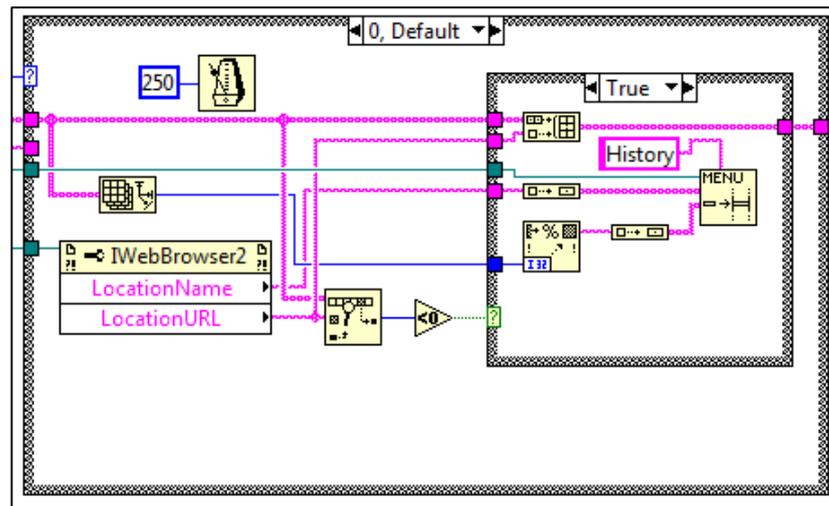


Figura 89 Caso 0, Default

Fuente: Autor

Es la condición con la que inicia la comunicación a la cámara. Inicializa los arreglos para el menú y la pantalla de visualización del explorador mediante la herramienta IWebBrowser2.

- Caso 1, *Reload*

Envía el comando de refrescar la ventana del navegador en caso que la transmisión de video se detenga en cierto fotograma.

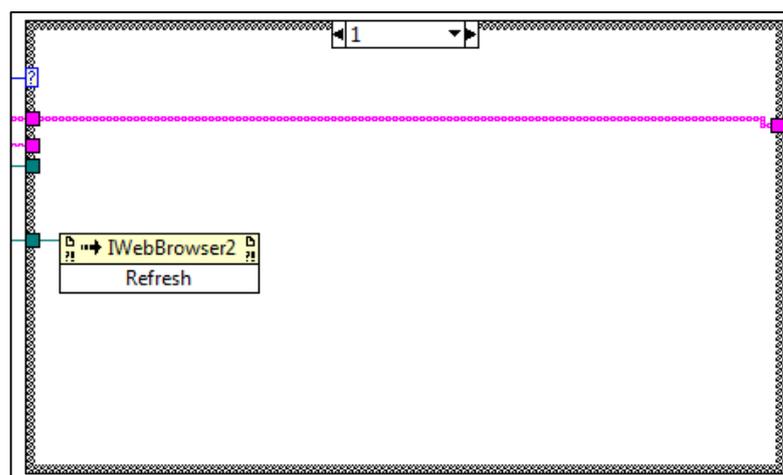


Figura 90 Caso 1, Reload

Fuente: Autor

- Caso 2, *Go*

Inicia el navegador en una dirección URL específica, en este caso la dirección de la cámara IP (192.168.1.140:81)

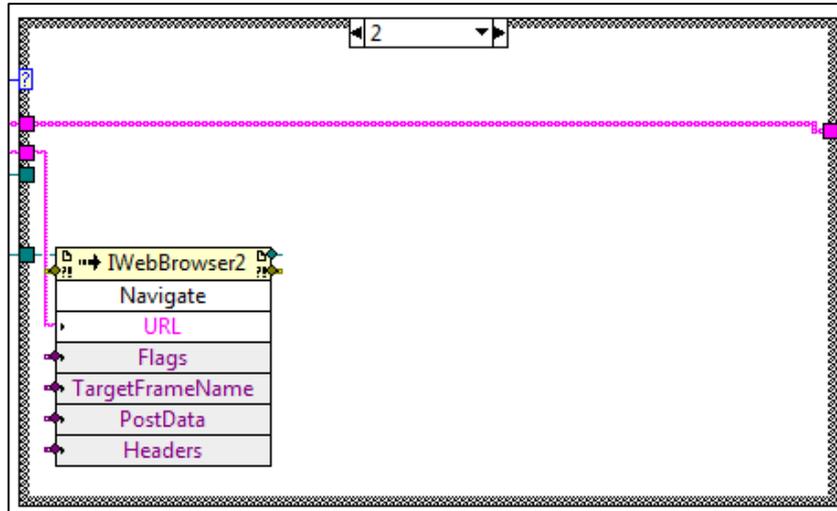


Figura 91 Caso 2, Start

Fuente: Autor

Finalmente un *shift register* mantiene la ventana en el estado anterior si no se ha pulsado un botón.

4.4 SUBVI ADQUISICIÓN DE DATOS DUALSHOCK 3

Para poder obtener los datos del control DUALSHOCK 3 se ha tenido que utilizar el SubVI que se explicará a continuación.

Primero que todo se debe realizar la conexión del control con el computador, y que el programa reconozca esta conexión, para esto se utiliza el subVI

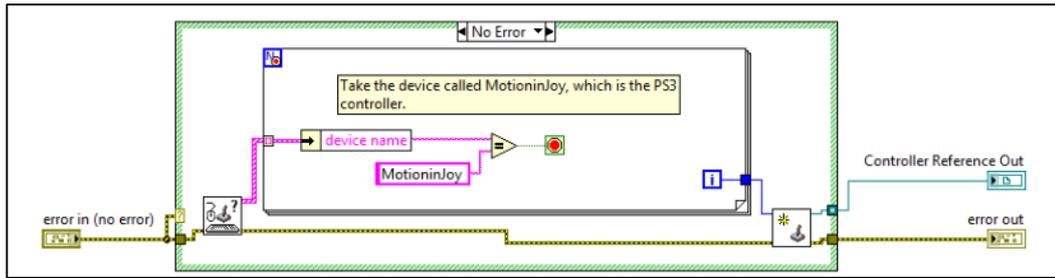


Figura 92 SubVI Conexión control

Fuente: Autor

Ahora que el dispositivo ha sido reconocido, se puede empezar a recopilar los datos. Inicialmente se obtienen los datos realizando una conexión con la herramienta *Acquire Input Data* del menú conectividad:

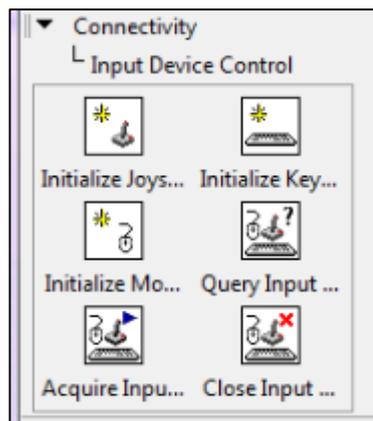


Figura 93 Menú de conectividad

Fuente: Autor

En la siguiente imagen se aprecia la lógica de control que permite recuperar la información del mando, separándola en un *array* según el nombre de la variable.

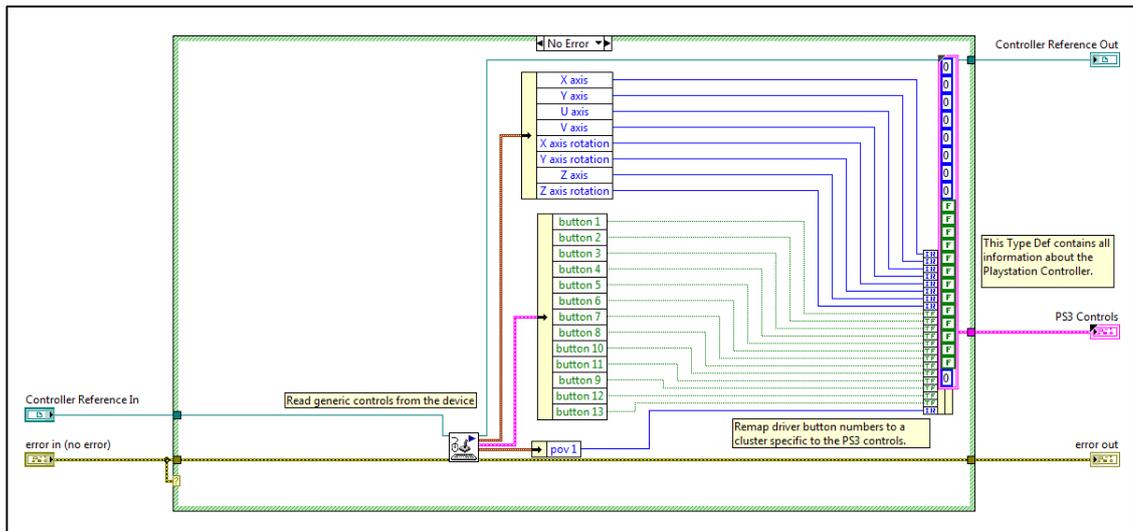


Figura 94 Obtención de datos de control DUALSHOCK

Fuente: Autor

Ahora que existe una conexión y los datos pueden ser recopilados en un arreglo, se necesita crear un SubVI que nos permita recopilar únicamente lo necesario para este proyecto, es por eso que se ha modificado el siguiente VI:

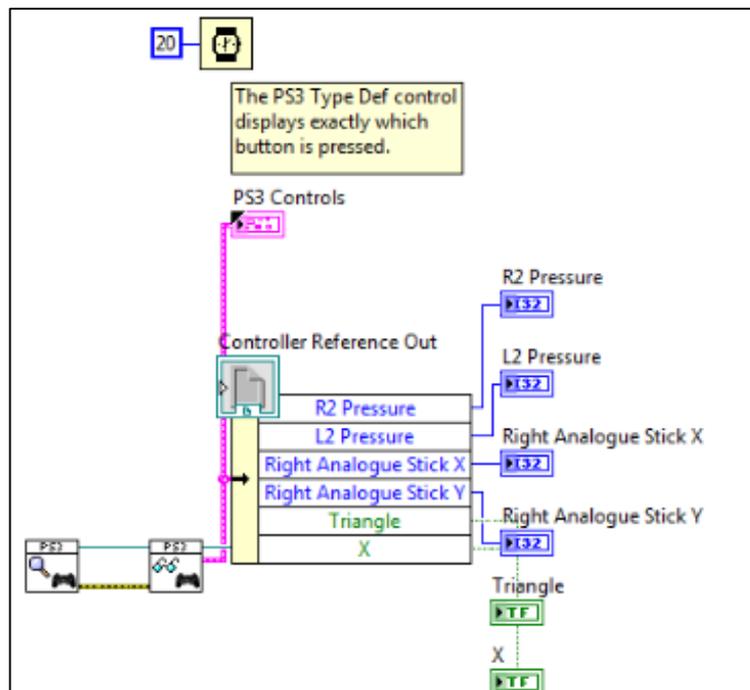


Figura 95 SubVI de adquisición de datos

Fuente: Autor

Con esto hemos podido recuperar la información necesaria para cumplir con los objetivos.

4.5 PROCESAMIENTO DE DATOS

4.5.1 Sentido de avance

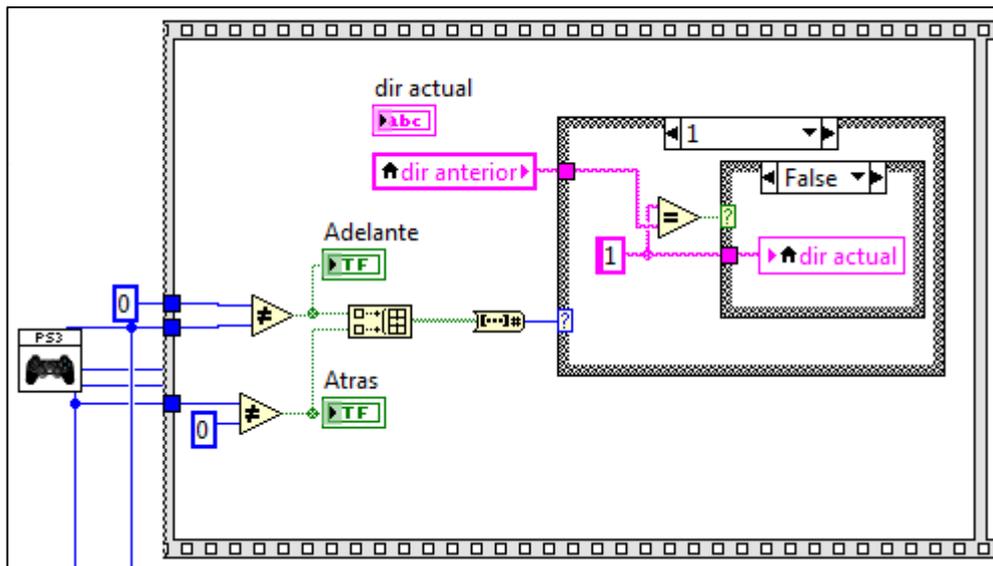


Figura 96 Diagrama de sentido avance

Fuente: Autor

El sentido de avance del robot se controla mediante los botones R2 y L2 del joystick siendo R2 de avance y L2 de reversa. El siguiente diagrama revisa todos los casos posibles en el que los botones pueden haber sido pulsados. Estas combinaciones son almacenadas en un arreglo teniendo como resultado la siguiente tabla:

Tabla 15 Posibles combinaciones de control de avance

R2	L2	DATO
DESACTIVADO	Desactivado	0
ACTIVADO	Desactivado	1

DESACTIVADO	Activado	2
ACTIVADO	Activado	3

Fuente: Autor

Posteriormente este dato es enviado aun *case structure* y actualizara la variable “dir actual” según el caso correspondiente. Adicionalmente se tiene un control en caso de que la dirección no ha variado para evitar la congestión en él envío de datos, el dato es enviado solamente cuando la variable ha cambiado de estado.

A continuación se detalla cada caso:

Caso 0, no se envía ningún dato porque no se ha presionado ningún botón.

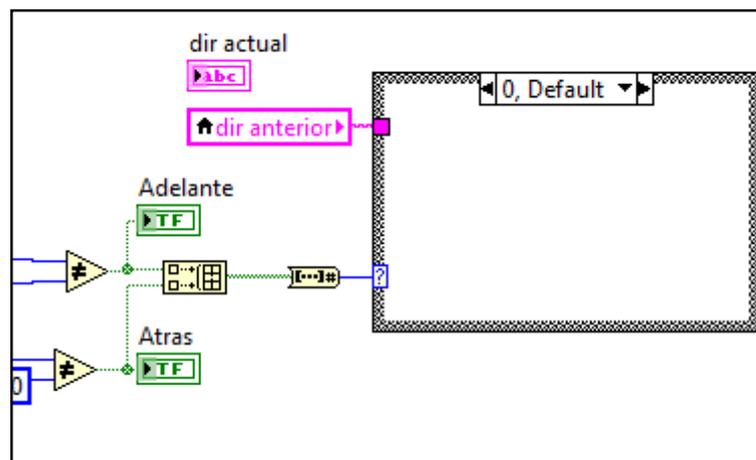


Figura 97 Caso 0 sentido de avance

Fuente: Autor

Caso 1, la variable dir actual cambia a 1 al presionar R2 de avance, en caso de que se mantenga el valor de la variable no se realiza ninguna acción (true).

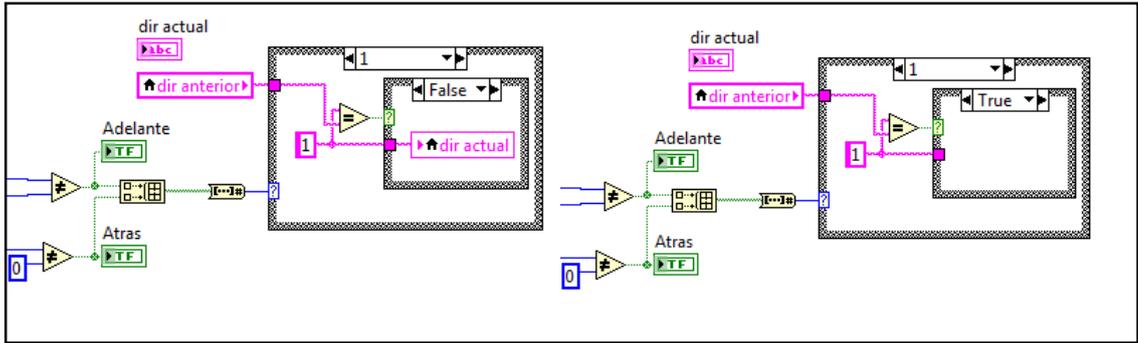


Figura 98 Caso 1 sentido de avance

Fuente: Autor

Caso 2, la variable dir actual cambia a 0 al presionar L2 de reversa.

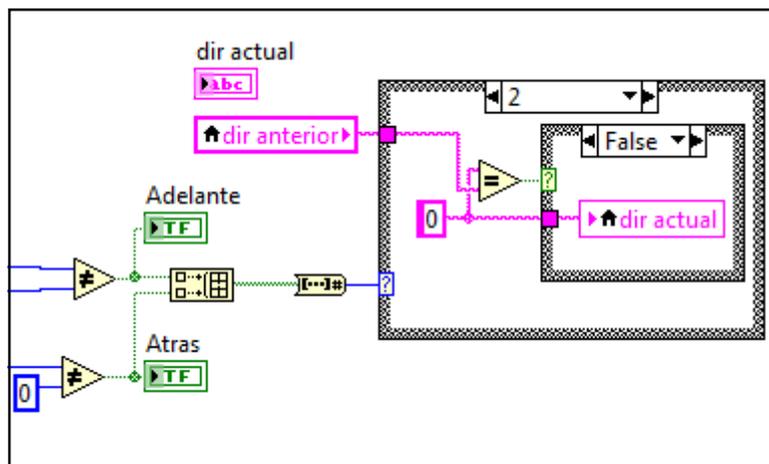


Figura 99 Caso 2 sentido de avance

Fuente: Autor

Caso 3, al presionar ambos botones no se envía ningún dato y no hay movimiento del robot

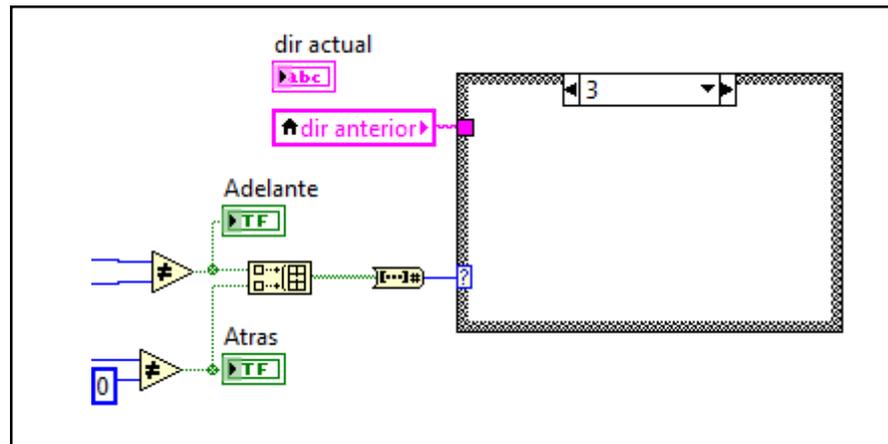


Figura 100 Caso 3 sentido de avance

Fuente: Autor

4.5.2 Velocidad de avance

Como se explicó anteriormente, los botones que controlan el movimiento tienen sensores de presión, con esto podemos obtener diferentes rangos de datos y delimitar 4 velocidades. Cada botón envía un número desde 0 cuando está desactivado hasta 35000 cuando está totalmente presionado. Mediante la herramienta "In Range and Coerce" se verifica el rango al que pertenece el dato siendo:

- 0 – Paro
- 1 – Lento
- 2 – Medio
- 3 – Rápido

Estos valores son enviados a la variable velocidad.

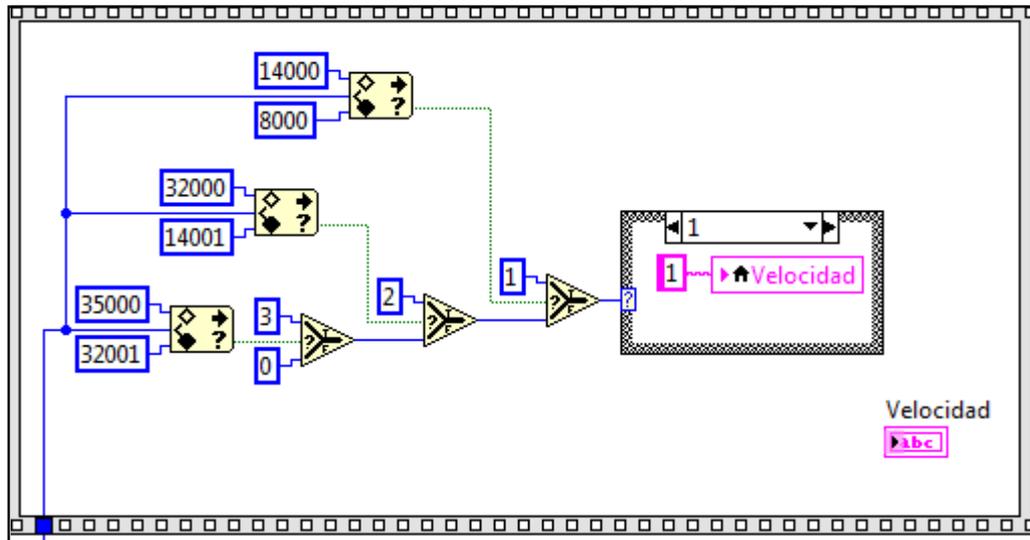


Figura 101 Control de velocidad

Fuente: Autor

Posición servomotores

Los dos servomotores que controlan la orientación de la cámara son controlados por una palanca que se mueve en los ejes X y simultáneamente. Esto genera dos datos en el rango de -32768 a 32768 siendo 0 el punto neutro. Los servomotores deben recibir un dato entre 0 y 180 grados, por lo tanto con la siguiente formula ajustamos el valor para que pueda ser reconocido por el servomotor.

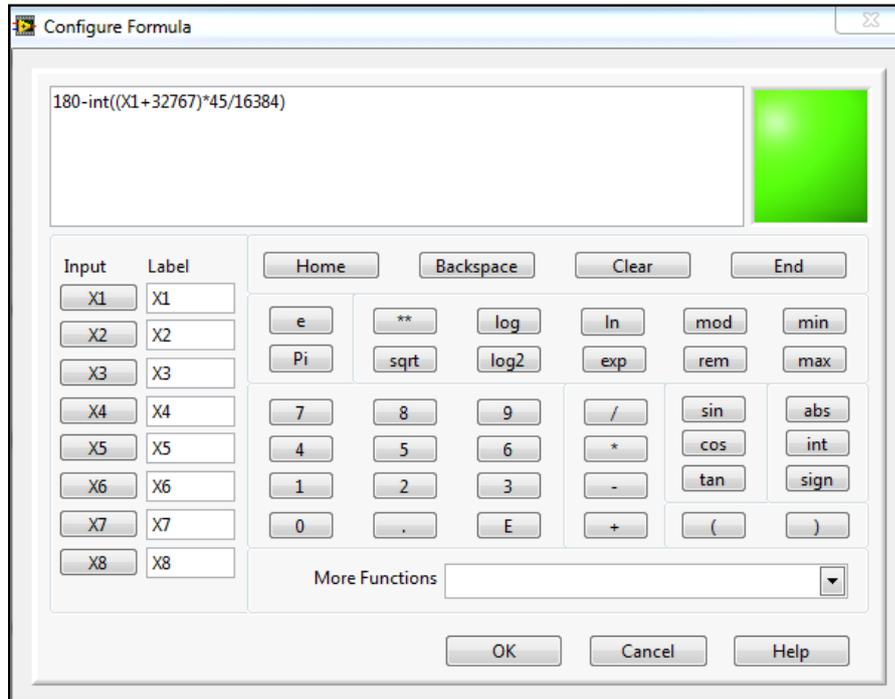


Figura 102 Fórmula de corrección de rango servomotor

Fuente: Autor

Posteriormente se usaron comparadores para delimitar el rango de movimiento de la cámara porque existían colisiones con el robot, estos rangos son para el eje X de 50 a 140 grados y para el eje Y de 50 a 117 grados.

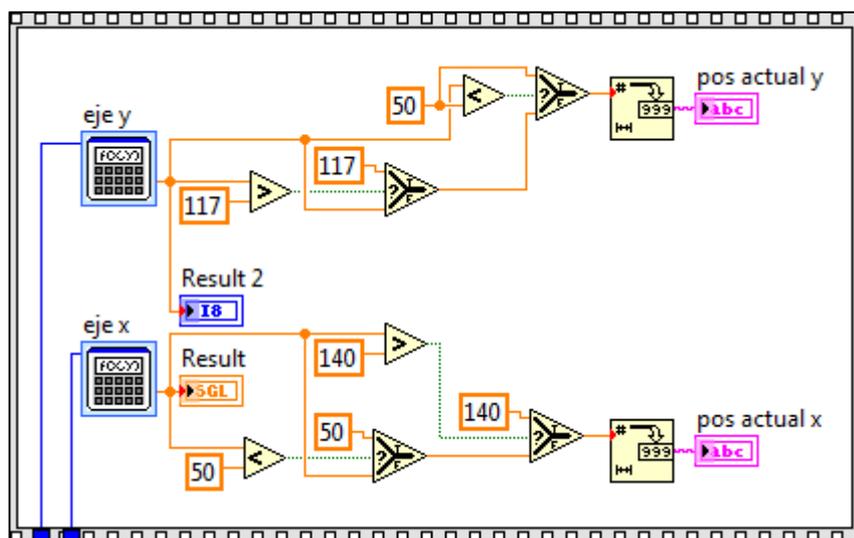


Figura 103 Limitación de movimiento de la cámara

Fuente: Autor

Finalmente los valores de cada servomotor son almacenados en las variables “pos actual x” y “pos actual y”.

4.5.3 Envío de trama de datos a Arduino

Ahora que se ha hecho todo el análisis los movimientos y ángulos de los motores se debe enviar de alguna manera estos datos a la Placa Arduino por medio de bluetooth. Para esto utilizamos una trama de datos, separados con “,” y en un orden específico:

1. Dirección de avance
2. Posición del servomotor en X
3. Posición del servomotor en Y
4. Velocidad
5. Iluminación

En la imagen a continuación se puede ver como se forma esta trama, para indicar el final de la misma se ha colocado de manera arbitraria la letra h.

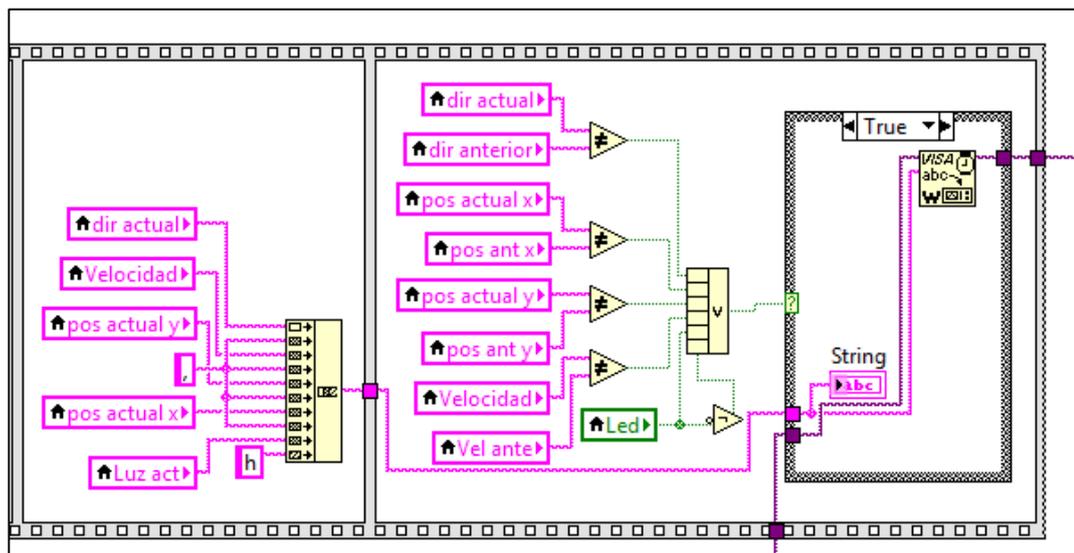


Figura 104 Formación trama de datos

Fuente: Autor

CAPÍTULO V

PRUEBAS DE FUNCIONAMIENTO, DETECCIÓN Y CORRECCIÓN DE ERRORES.

Mediante las diferentes pruebas se intentará llegar a una alta eficiencia, para lo cual se realizarán dichas pruebas en distintos escenarios.

5.1 PRUEBA DE INGRESO AL TUBO

Inicialmente al haber finalizado la estructura metálica con las ruedas se hizo pruebas de ingreso del robot (sin componentes electrónicos y con seis ruedas posteriores) al tubo y no hubo ningún inconveniente. Después de ensamblar por completo el robot, las ruedas delanteras ingresaban sin mayor esfuerzo al tubo pero las ruedas posteriores al comprimirse chocaban con baterías lo que las bloqueaba e impedía el avance. Para solucionar este inconveniente de espacio, se optó por solo usar tres ruedas (una por extremidad) y aprovechar el espacio extra para ubicar las baterías.

Con la estructura finalizada se apretaron las tuercas de seguridad dejando aproximadamente un milímetro de holgura para no impedir el movimiento de compresión. Al probar nuevamente el robot ingresó en el tubo sin mayor esfuerzo y con un leve empuje se deslizaba suavemente en el tubo en ambas direcciones.

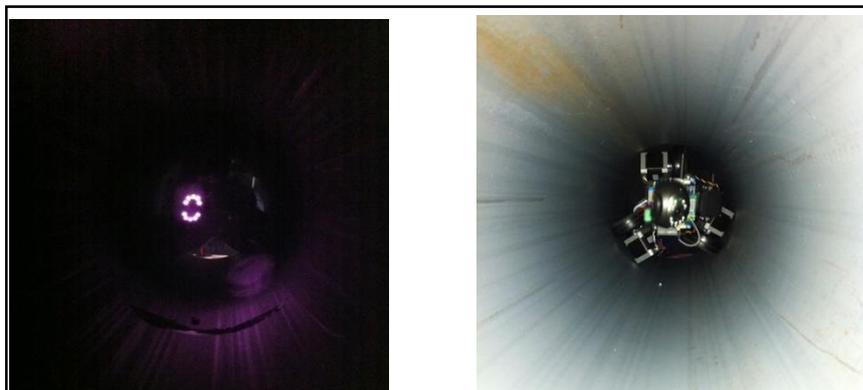


Figura 105 Robot dentro de la tubería

Fuente: Autor

5.2 PRUEBA DE ÁNGULO MÁXIMO DE OPERACIÓN

Normalmente los tubos donde operará el robot no se encuentran en óptimas condiciones, pueden tener polvo, lodo, humedad, moho etc. Esto limita la capacidad de adherencia y tracción de las ruedas al disminuir el coeficiente de rozamiento estático en comparación de un tubo seco y limpio. Por este motivo se realizó una serie de pruebas para comparar el máximo ángulo de elevación en el que el robot trabaja sin problemas con respecto a varios entornos.

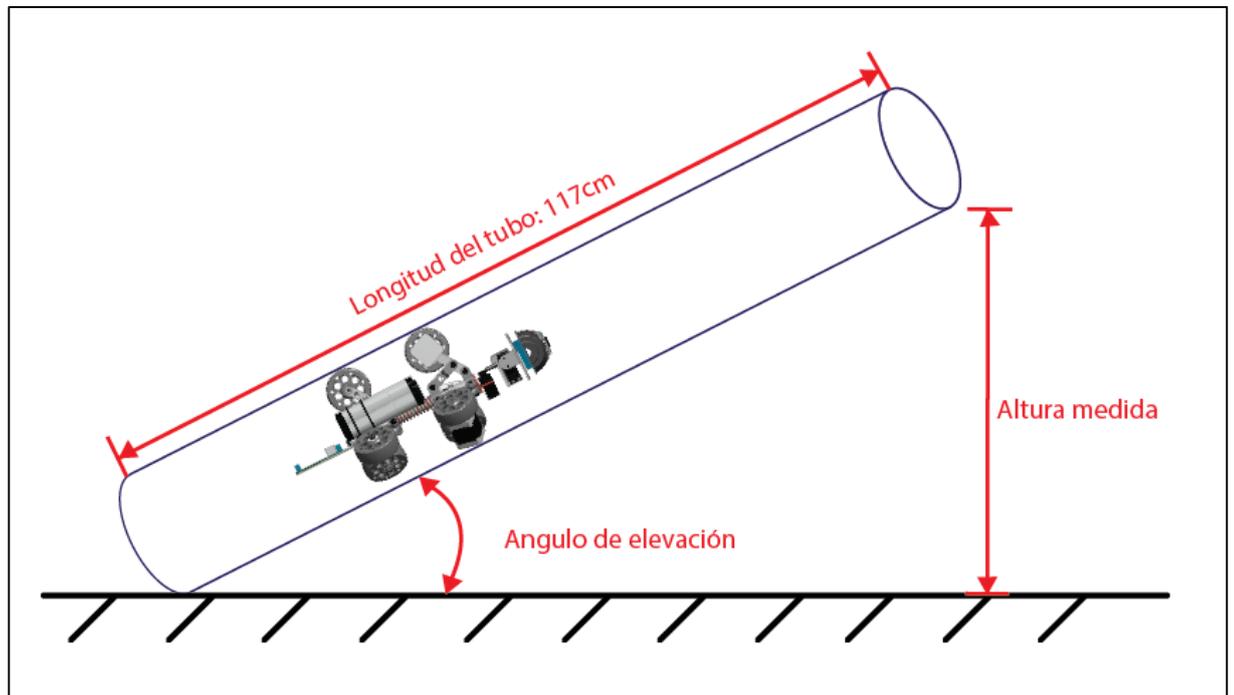


Figura 106 Medición del ángulo de elevación

Fuente: Autor

Se midió la longitud del tubo de pruebas (117cm) y se elevó lentamente un extremo del tubo hasta que las ruedas pierdan tracción y el robot ya no pueda avanzar o se resbale. Conociendo esta altura y la longitud del tubo podemos obtener fácilmente el ángulo máximo de elevación.

$$\sin \alpha = \frac{\text{Altura medida (cm)}}{117\text{cm}}$$

$$\alpha = \text{arcSin}\left(\frac{\text{Altura medida (cm)}}{117\text{cm}}\right)$$

Las pruebas se realizaron tomando en cuenta 4 estados del robot: Inmóvil, velocidad baja, velocidad media y velocidad alta. A su vez se consideró 3 posibles entornos: tubo seco y limpio (como referencia), tubo seco y con polvo, y un tubo húmedo con paredes jabonosas.

Para esto se usó un tubo que por haber estado almacenado varios meses contaba con una capa de polvo y se realizó la primera prueba, luego se lo lavo con jabón y se aprovechó la espuma formada para la segunda prueba y finalmente la última prueba se realizó con el tubo limpio y seco.

Los datos obtenidos están registrados en la siguiente tabla

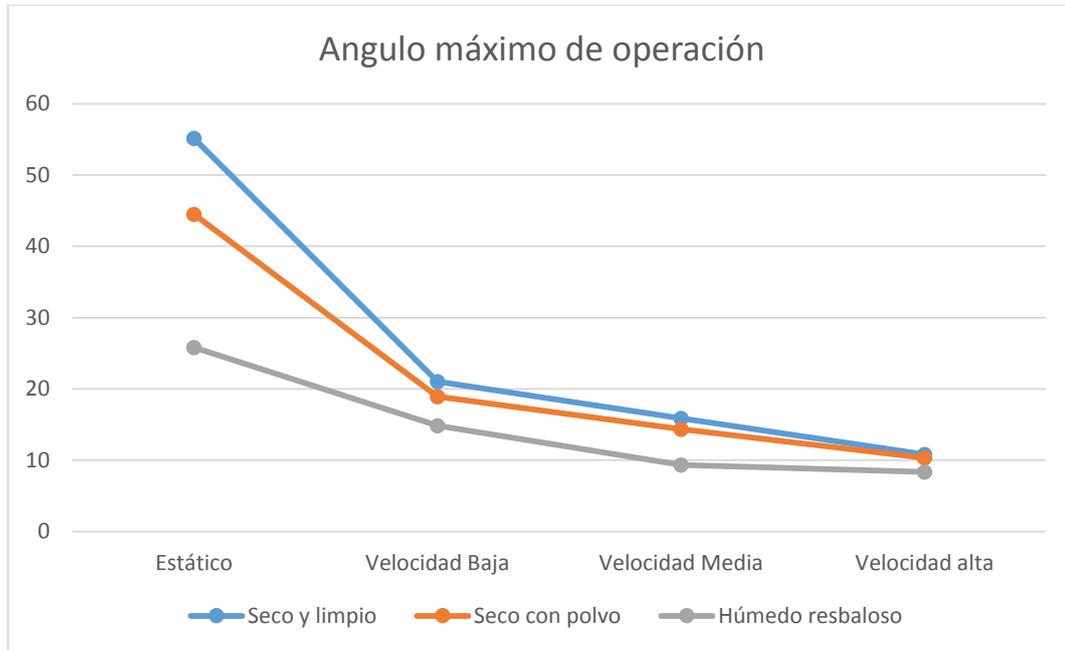
Tabla 16 Ángulos máximos operación

Entorno	Estado	Altura medida (cm)	Angulo máximo
SECO Y LIMPIO	Estático	96	55.13°
	Velocidad baja	42	21.03°
	Velocidad media	32	15.87°
	Velocidad alta	22	10.83°
SECO CON POLVO	Estático	82	44.50°
	Velocidad baja	38	18.95°
	Velocidad media	29	14.35°
	Velocidad alta	21	10.34°
HUMEDO RESBALOSO	Estático	51	25.84°
	Velocidad baja	30	14.86°
	Velocidad media	19	9.36°
	Velocidad alta	17	8.35°

Fuente: Autor

En el siguiente gráfico se observa más claro la diferencia de tracción en los diferentes entornos.

Tabla 17 Comparación ángulos máximos de operación



Fuente: Autor

Se observa que existe gran diferencia en los ángulos máximos de operación cuando el robot se encuentra estático y cuando está en movimiento, llegando a ser más del doble. En comparación con un tubo limpio y seco, el tubo con polvo no demuestra mucha diferencia cuando en movimiento pero detenido necesita 10 grados menos para no resbalarse al reducirse la fricción por el polvo. En el ambiente húmedo y resbaloso el ángulo máximo de elevación estático se reduce a la más de la mitad, mientras que en movimiento no la diferencia es de alrededor del 25%.

CONCLUSIONES

Después de haber construido el robot en su totalidad a partir de materia prima, equipos de comunicación, actuadores, etc., llegamos a las siguientes conclusiones:

- Como sucedió, el hierro se oxida al pasar pocas semanas aun en ambientes secos debido a la manipulación con las manos. Por lo tanto el tratamiento térmico de protección pavonado debe realizarse tan pronto como las piezas estén listas, para evitar limar o limpiar el óxido antes de pavonar.
- Se observó que el mayor inconveniente es el peso de la estructura del robot. Para disminuir la masa total, se la puede realizar de aluminio (densidad 2.7 g/cm^3) en comparación del hierro (densidad 7.15 g/cm^3). Esto reduciría el peso de la estructura a un 37.7% del peso actual. Por motivos de costos, se usó plancha de hierro que fue donada por la empresa INGMATRICOM para el prototipo debido a que se puede usar en la electroerosión de hilo. El aluminio es más liviano y barato pero no se lo puede cortar con electroerosión de hilo. Para maquinarse en las piezas en aluminio se necesitaría fresadoras y tornos lo que se traduce a mayor tiempo de máquina, desgaste de herramientas y necesidad de un operario, esto sería ideal en caso de producir grandes cantidades en serie.
- De igual manera se podría reemplazar los motores por unos de mayor torque para garantizar el ascenso del robot por tuberías con ángulos mayores a 55° como se revisó en el capítulo anterior. Se pueden encontrar motores NEMA 17 con las mismas características que los que se están utilizando actualmente pero con un mayor torque, de aprox. 125 oz-in, es decir 3 veces mayor al actual. Obviamente el costo es más elevado pero en caso de necesitarlos solamente se reemplazaría la pieza del robot sin necesidad de hacer ningún cambio adicional en la estructura.

- Las ruedas usadas no son las idóneas para el robot ya que podrían ser más delgadas, livianas y de un caucho más suave para mejorar la tracción. En el mercado local no se encuentra este tipo de ruedas por lo que se usó las actuales mejorando los bordes y realizando agujeros para reducir el peso.
- Para mejorar el alcance de comunicación inalámbrica se podría utilizar la tecnología de radio frecuencia (RF), que generalmente trabajan en bandas de alrededor de los 900Mhz, para esto están disponibles en el mercado, Shields para Arduino que ofrecen un alcance desde 150 a 2 Km en interiores y de 200 a 3 Km en exteriores y en línea de visión directa.

BIBLIOGRAFÍA

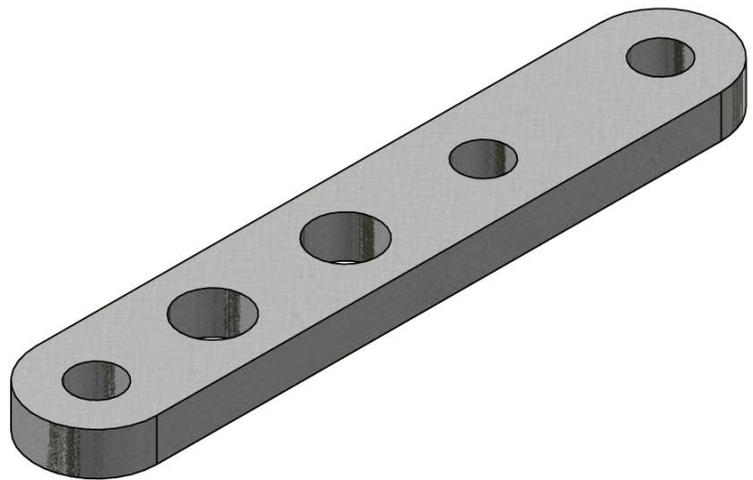
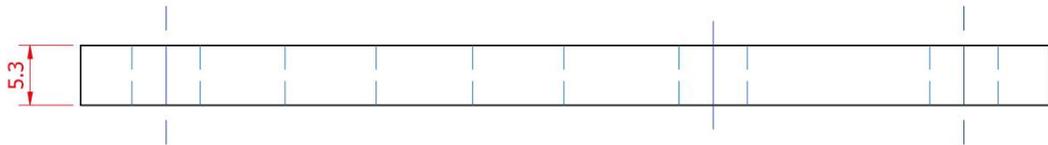
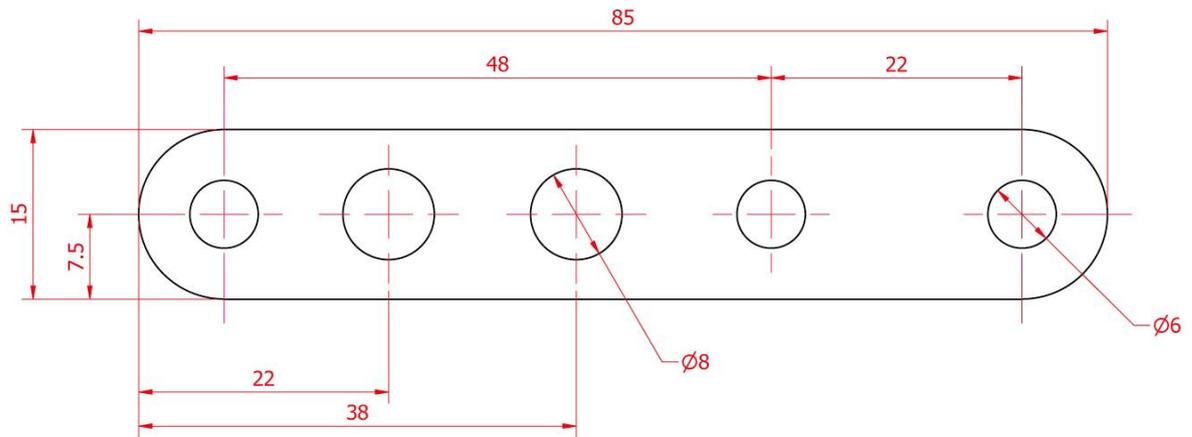
- AMAZON. (2014). Obtenido de
http://www.amazon.com/gp/product/B002KFPU7G/ref=oh_aui_detailpage_o04_s00?ie=UTF8&psc=1
- AMAZON. (2014). Obtenido de http://ecx.images-amazon.com/images/I/71xAxG8oVrL._SL1200_.jpg
- ARDUINO. (Abril de 2014). Arduino. Obtenido de
http://arduino.cc/en/uploads/Main/ArduinoUno_R3_Front.jpg
- ARDUINO. (2015). Recuperado el 01 de 2015, de
<http://arduino.cc/en/pmwiki.php?n=Guide/Environment>
- ARDUINO. (s.f.). ARDUINO. Obtenido de
http://arduino.cc/en/uploads/Main/ArduinoMega2560_R3_Front.jpg
- BANZI, M. (2009). Getting Started with Arduino (First Edition ed.). (B. Jepson, Ed.) CA, USA: O'Reilly Media Inc.
- BITTER, R., MOHIUDDIN, T., & NAWROCKI, M. (s.f.). LabVIEW Advanced Programming Techniques (Second Edition ed.). Boca Raton, FL, US: CRC Press.
- BLUETOOTH. (2015). Recuperado el enero de 2015, de
<http://www.bluetooth.com/Pages/Fast-Facts.aspx>
- BORENSTEIN, J., EVERETT, H., & FENG, L. (1996). Where am I? Sensors and Methods for Mobile Robot Positioning (Vol. 119). University of Michigan.
- BRÄUNL, T. (2008). Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems (Second Edition ed.). Berlín: Springer.
- CHENG, I.-M., YEO, S. H., & GAO, Y. (2001). Locomotive gait generation for inchworm-like robots using finite state approach. En *Robotica* (pág. 535).
- CHOI, H., & ROH, S. (2007). In-pipe Robot with Active Steering Capability for moving inside pipelines. *Walking and Climbing Robots*. (M. K. Habib, Ed.)
- EasyNUSA. (13 de Agosto de 2014). Easynusa. Recuperado el 2014, de
<http://easynusa.com/pan-tilt-cameras/easyn-fs-613b-m166i-wireless-ip-camera>

- ESPINOZA Velasquez, M. E. ([200-]). "Diseño Y Construcción De Un Robot Móvil Teleoperado Para La Localización De Obstáculos En Tuberías". Sangolquí, Ecuador.
- FERGUSON, D., MORRIS, A., HÄNEL, A., BAKER, C., OMOHUNDRO, Z., REVERTE, C., . . . THRUN, S. (2003). An Autonomous Robotic System for Mapping Abandoned Mines. NIPS. Obtenido de http://books.nips.cc/papers/files/nips16/NIPS2003_AP03.pdf
- GNU. (25 de agosto de 2013). Obtenido de GNU General Public License: <http://www.gnu.org/copyleft/gpl.html>
- IDS. (Junio de 2013). FZI Forschungszentrum Informatik Karlsruhe - Abteilung. Obtenido de <http://www.fzi.de/en/forschung/projekt-details/lauron/>
- INSPECTAHIRE. (2014). Obtenido de <http://www.inspectahire.com/page4/page25/page25.html>
- INTEREMPRESAS. (12 de 03 de 2012). Recuperado el 2014, de <http://www.interempresas.net/MetalMecanica/Articulos/102921-Mecanizado-por-electroerosion.html>
- International Federation of Robotics. (2014). Obtenido de <http://www.ifr.org/industrial-robots/statistics/>
- IONESCU, H. (Julio de 2014). Wikipedia. Obtenido de "6DOF en" by Horia Ionescu - I (Horia Ionescu) Licensed under [Publhttp://commons.wikimedia.org/wiki/File:6DOF_en.jpg#mediaviewer/File:6DOF_en.jpg](http://commons.wikimedia.org/wiki/File:6DOF_en.jpg#mediaviewer/File:6DOF_en.jpg)
- ISLAS, R. (mayo de 2007). Academia de electrónica CONALEP San Martín. Obtenido de <http://acadelectronconatex.blogspot.com/2007/05/interpretacin-de-documentacin-tnica.html>
- MARGOLIS, M. (2012). Arduino Cookbook (Second Edition ed.). (S. W. Jepson, Ed.) CA, USA: O'Reilly Media.
- MARK W. Spong, S. H. (2006). Robot Modeling and Control. Estados Unidos: Jhon Wiley & Sons, Inc.
- Medical and electromechanical design lab. (2007). Obtenido de http://research.vuse.vanderbilt.edu/MEDLab/research_files/prevproj/crawler.htm

- MONTIEL Garcia, A. (Agosto de 2014). ESTUDIO DE LA APLICACIÓN DE TRATAMIENTOS DE CONVERSIÓN QUÍMICA DE PAVONADO SELLADO CON CERIO SOBRE ACERO DE REFUERZO EMBEBIDO EN CONCRETO COMO PROTECCIÓN ANTICORROSIVA. Mexico. Recuperado el 2015, de <http://tesis.ipn.mx:8080/xmlui/handle/123456789/13305>
- PASTOR Morris, W. (2012). Informe de Gestión 2012. Ministerio de Recursos Naturales No Renovables del Ecuador. Quito: Dirección de Comunicación Social 2013. Recuperado el 12 de 09 de 2013, de http://www.recursosnaturales.gob.ec/wp-content/uploads/2013/02/informe_gestion_2012.pdf
- PLAYSTATION. (s.f.). Obtenido de <http://us.playstation.com/ps3/accessories/dualshock-3-wireless-controller-ps3.html>
- POLOLU Robotics & Electronics. (22 de Agosto de 2014). Obtenido de Stepper Motor: Unipolar/Bipolar, 200 Steps/Rev, 42×48mm, 4V, 1.2 A/Phase: <http://www.pololu.com/picture/view/0J2976>
- POLOLU Robotics & Electronics. (Septiembre de 2014). Obtenido de <https://www.pololu.com/product/2134>
- PROCESSING. (s.f.). Obtenido de <https://processing.org/>
- ROBOTICS PORTAL. (1 de OCTUBRE de 2012). Obtenido de <http://www.roboticsportal.it/en/Robot>
- RYEW, B., RYU, J., & ROH, C. (2000). In-pipe inspection robot system with active steering mechanism. Proc of IEEE/RSJ Int. Conf on Intelligent Robots and Systems, 3.
- SCHRODER. (2013). Alibaba. Obtenido de http://szans.en.alibaba.com/product/547576065-210420096/Schroder_professional_pipeline_sewer_drainage_inspection_robotic_crawler_SD_9901.html
- SEEDSTUDIO. (Diciembre de 2014). Obtenido de http://www.seedstudio.com/wiki/index.php?title=Bluetooth_Shield
- SHAFFER, G., & STENTZ, A. (1992). A Robotic System for Underground Coal Mining. Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference.

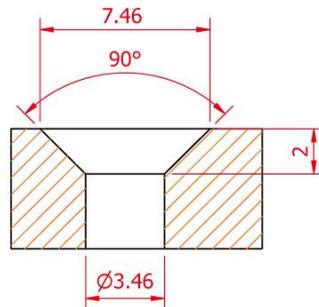
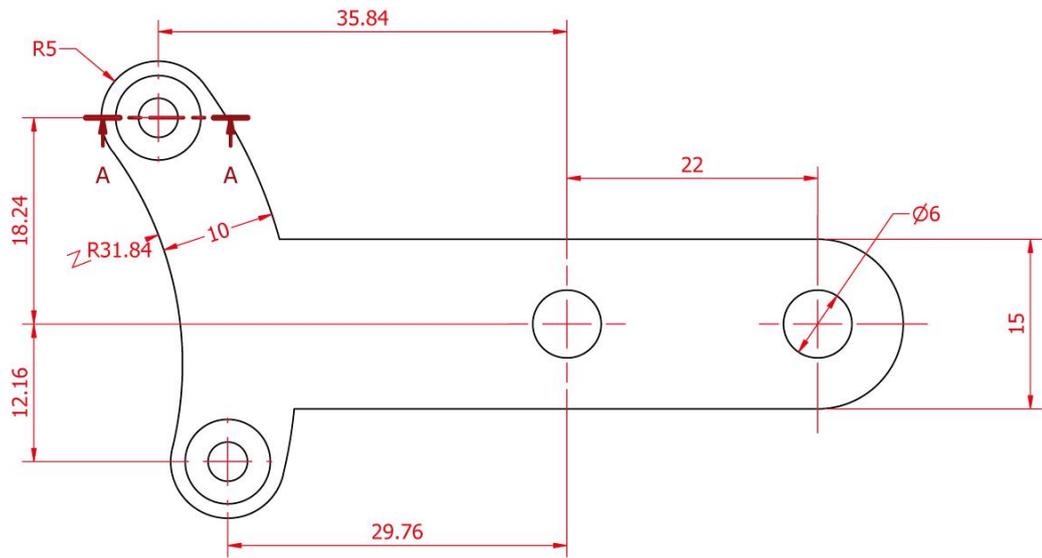
- SHIRALKAR, M., & National Instruments. (2007). LabVIEW Graphical Programming Course. (N. Instruments, Ed.) Houston, Texas, US: Rice University. Obtenido de <<http://cnx.org/content/col110241/1.4/>>
- SIEGWART, R., Nourbakhsh, I., & SCARAMUZZA, D. (2011). Introduction to Autonomous Mobile Robots. Massachusetts: MIT press.
- STACHNISS, C. (2009). Robotic Mapping and Exploration. Berlin: Springer-Verlag.
- SUZUMORI, K., MIYAGAWA, T., KIMURA, M., & HASEGAWA, Y. (s.f.). Micro Inspection Robot for 1-in Pipes.
- T, N., K, O., & TAMURA. (2010). Development of a Simulation for Inner-Gas-Pipe Inspection Robot: SPRING.
- TATAR, O., MANDRU, D., & ARDELEAN, I. (2007). Development of mobile minirobots for in pipe inspection tasks.
- Texas Instruments. (Abril de 2013). Obtenido de <http://www.ti.com/lit/ds/symlink/lm2576.pdf>
- TUV. (Diciembre de 2013). TUV. Obtenido de http://www.tuev-sued.de/technical_installations/steam_and_pressure_technology/pipelines/pipeline_test_equipment
- Wikimedia Commons. (12 de 2013). Obtenido de http://commons.wikimedia.org/wiki/File:Arduino_Logo.svg
- ZAGLER, A., & PFEIFLER, F. (2003). Pipe Crawler for Tube Junctions.
- ZHANG, Y., FINGER, S., & BEHREN, S. (s.f.). Carnegie Mellon University. Recuperado el 2014, de <http://www.cs.cmu.edu/~rapidproto/mechanisms/chpt4.html#HDR54>
- Zona Boxes. (s.f.). Obtenido de <http://www.zonaboxes.net/t1779-brico-pavonado-de-tornillos>

ANEXOS

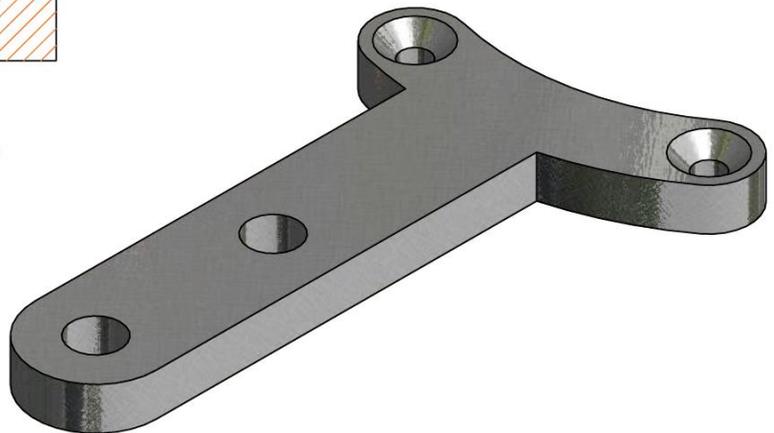


ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

 <p>UNIVERSIDAD DEL AZUAY</p>	<p>Escala 1.5:1</p>	<p>Material: Hierro</p>	<p>04/09/14</p>
	<p>TITULO</p> <p>Anexo 1: Link Simple</p>		<p>CANT.</p> <p>3</p>

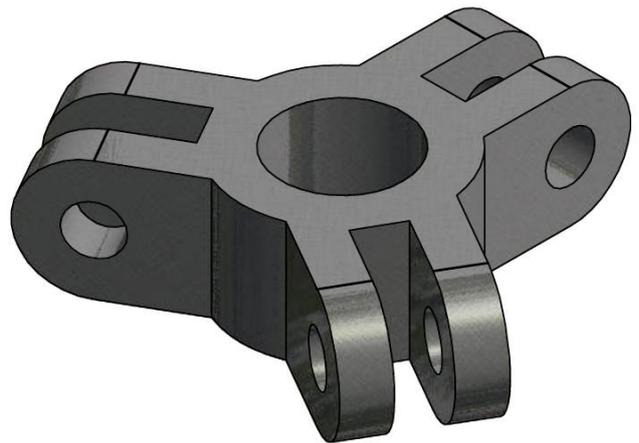
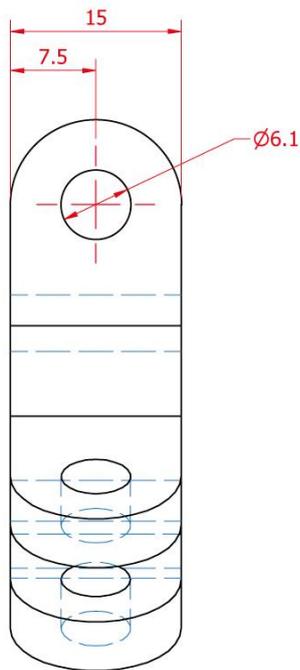
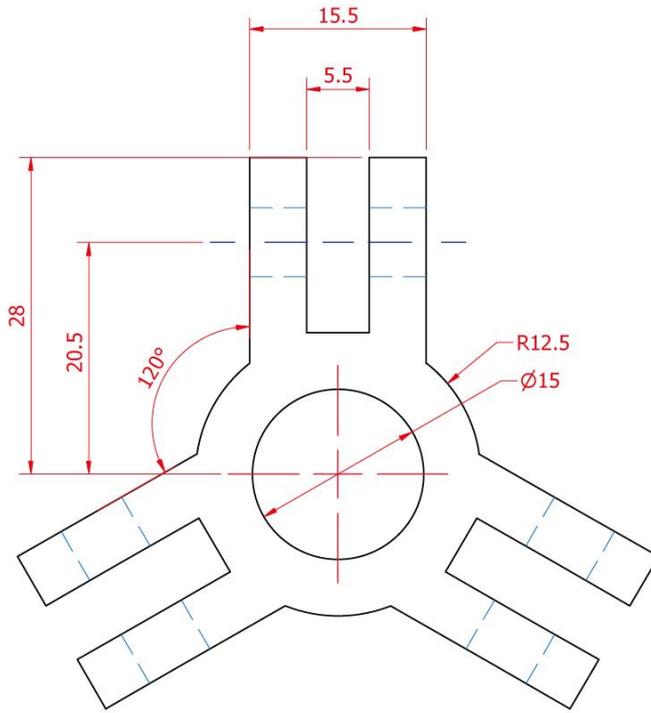


SECCION A-A
ESCALA 3:1



ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

	Escala 1.5:1	Material: Hierro	04/09/14
	TITULO Anexo 2: Link Motor		CANT. 3



ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO



Escala 1.5:1

Material: Hierro

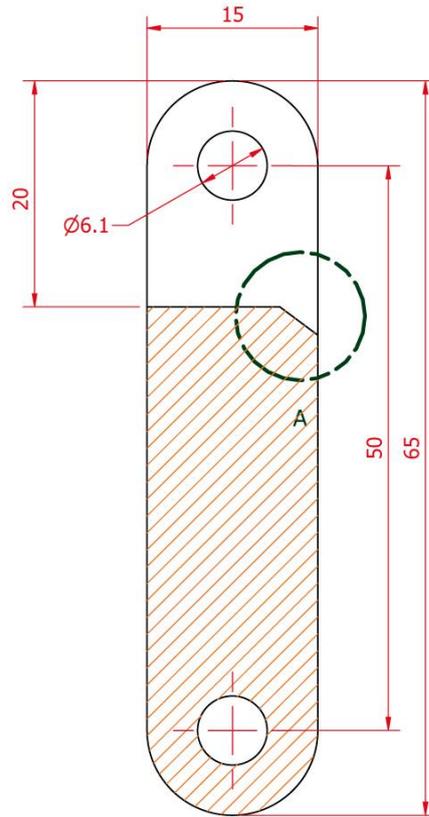
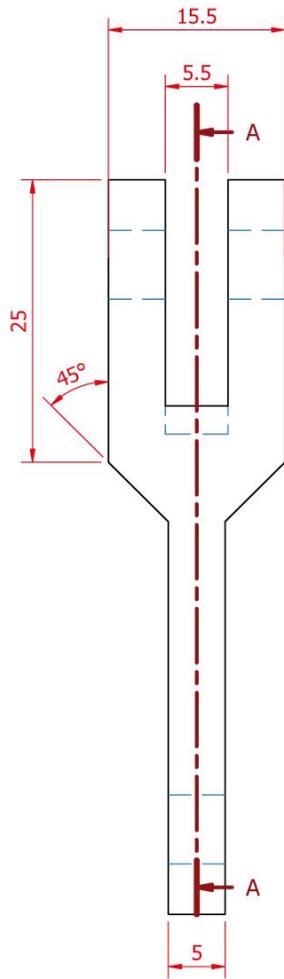
04/09/14

TITULO

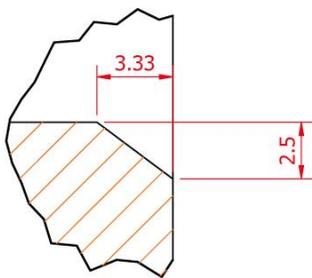
Anexo 3: Link unión triple

CANT.

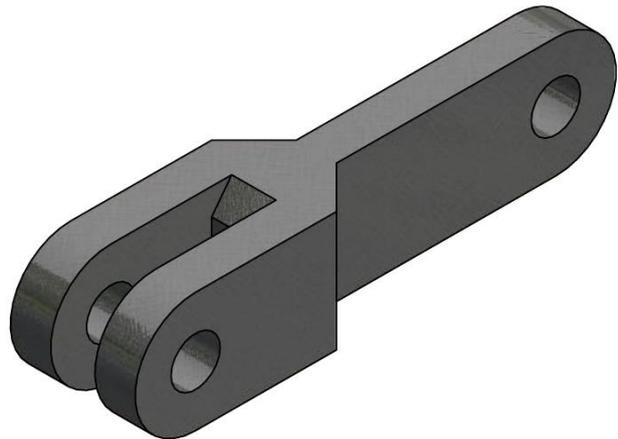
4



SECCION A-A
ESCALA 1.5

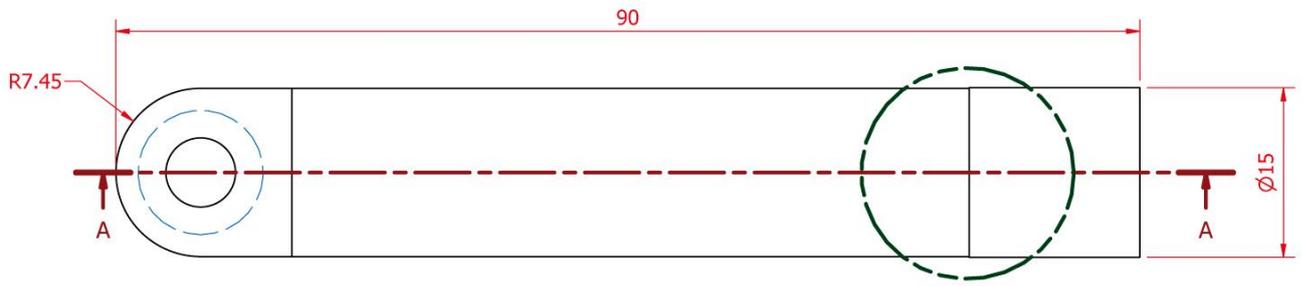


DETALLE A
ESCALA 3 : 1

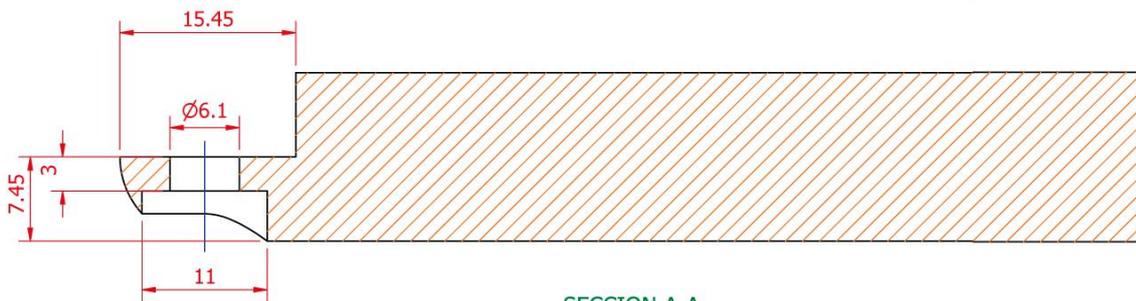


ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

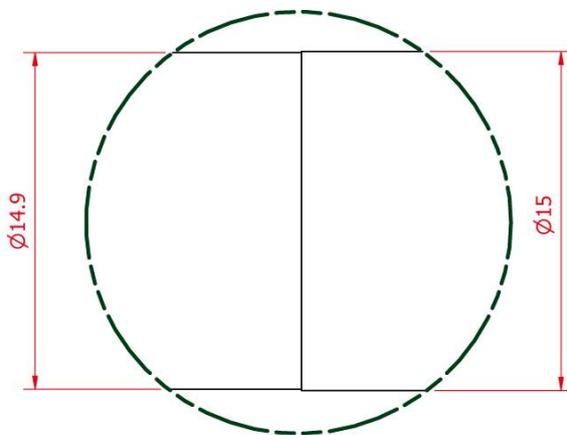
 <p>UNIVERSIDAD DEL AZUAY</p>	Escala 1.5:1	Material: Hierro	04/09/14
	<p>TITULO</p> <p>Anexo 4: Link lateral</p>		<p>CANT.</p> <p>6</p>



B



SECCION A-A
ESCALA 1.5

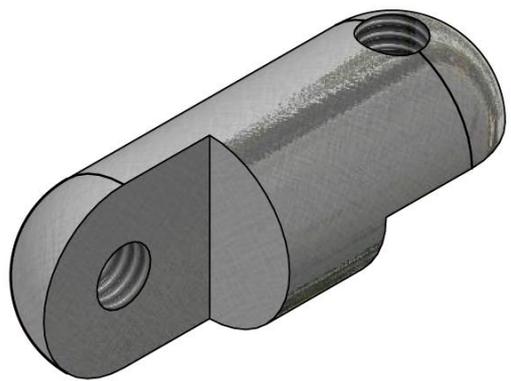
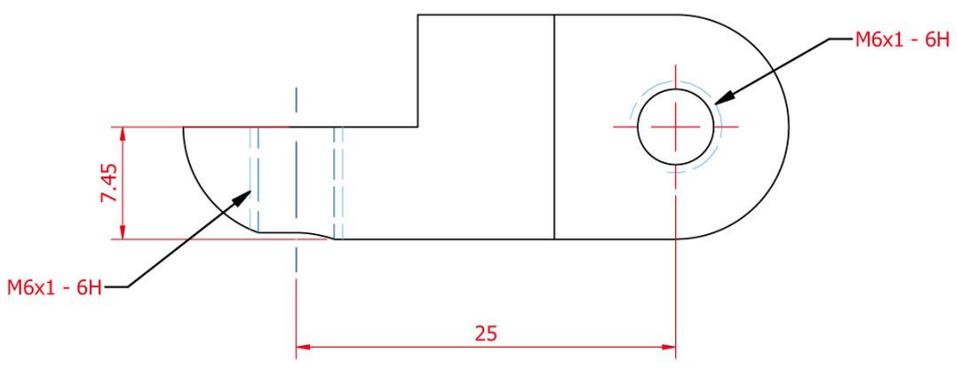
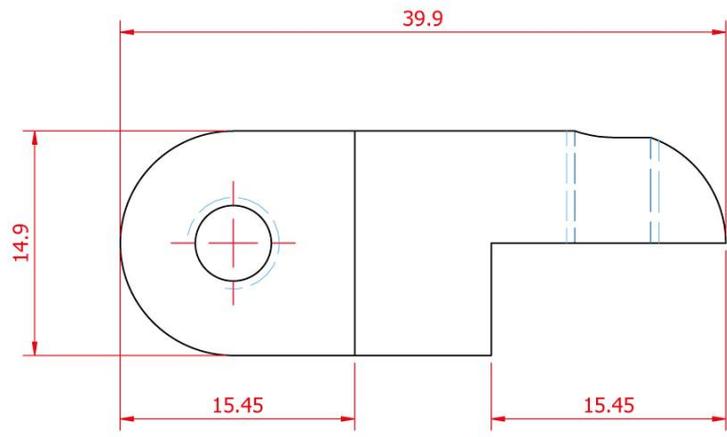


DETALLE B
ESCALA 3 : 1

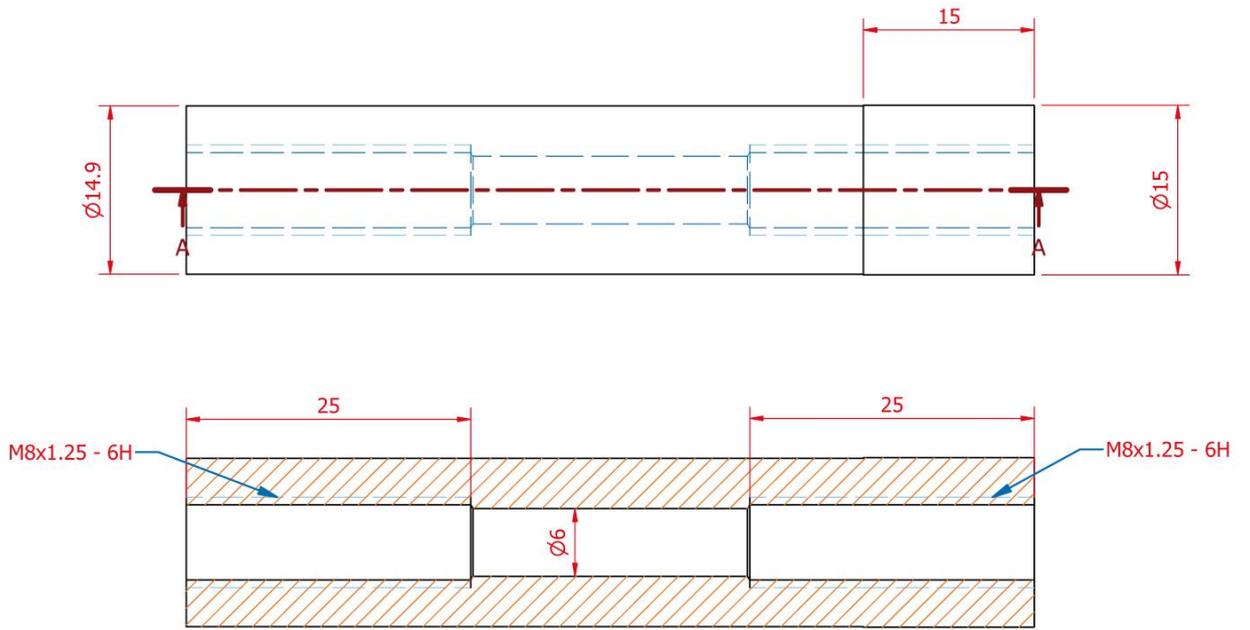


ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

 UNIVERSIDAD DEL AZUAY	Escala 1.5:1	Material: Hierro	04/09/14
	TITULO Anexo 5: Eje extremo		CANT. 2



ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO			
	Escala 2:1	Material: Hierro	04/09/14
	TITULO		CANT.
	Anexo 6: Unión ejes		1

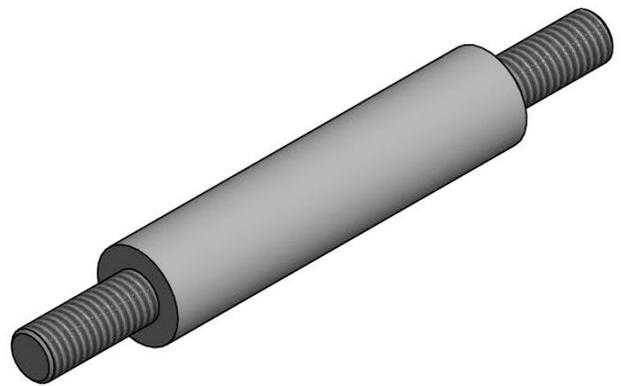
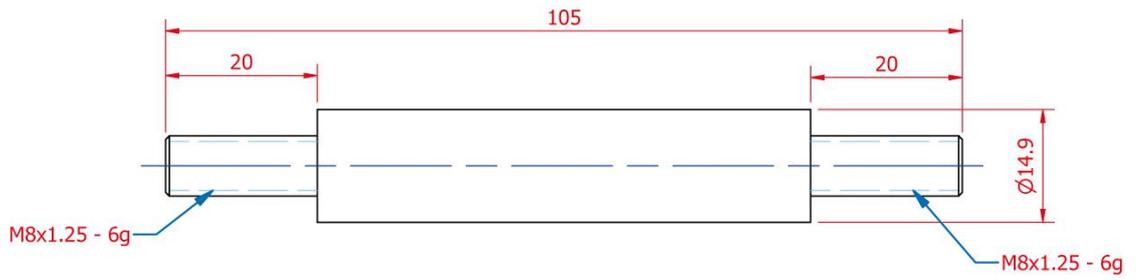


SECCION A-A
ESCALA 1.5



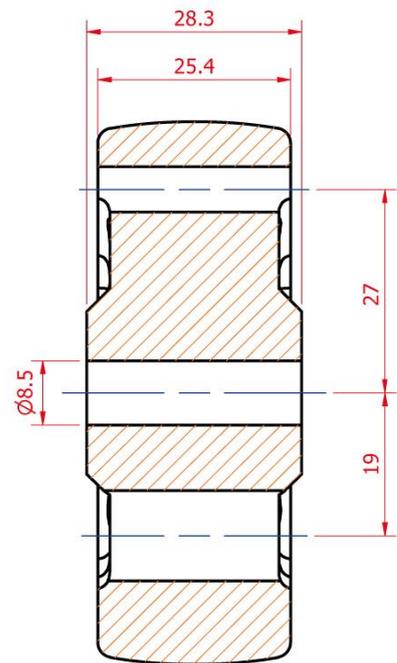
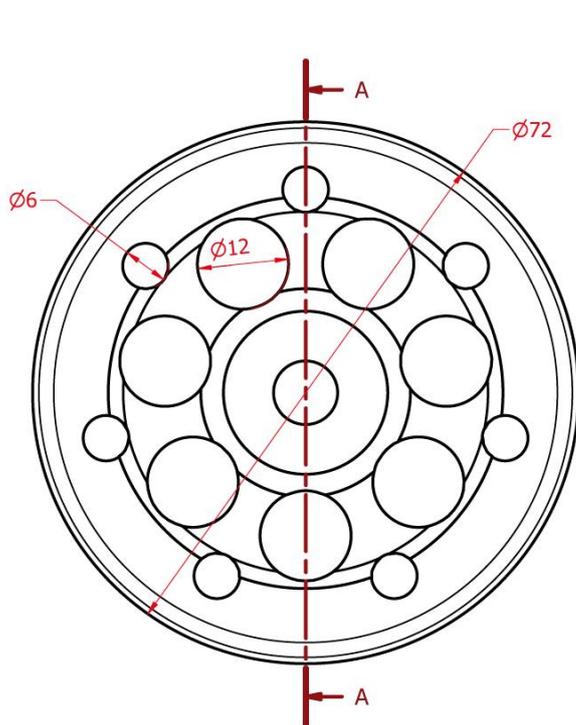
ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

 <p>UNIVERSIDAD DEL AZUAY</p>	<p>Escala 1.5:1</p>	<p>Material: Acero</p>	<p>09/10/14</p>
	<p>TITULO</p> <p>Anexo 7: Eje extremo V2</p>		<p>CANT.</p> <p>2</p>



ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

	Escala 1:1	Material: Acero	09/10/14
	TITULO Anexo 8: Unión ejes rectos		CANT. 1



SECCION A-A
ESCALA 1 : 1



ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO



Escala 1:1

Material: Caucho

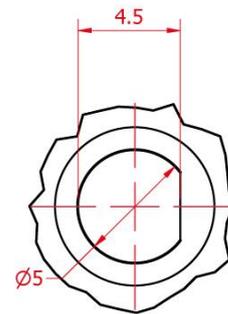
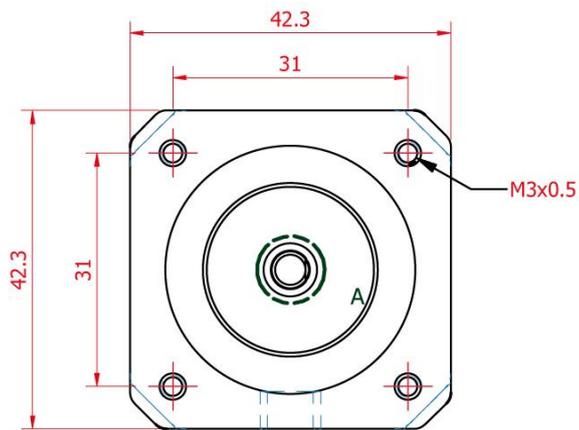
04/09/14

TITULO

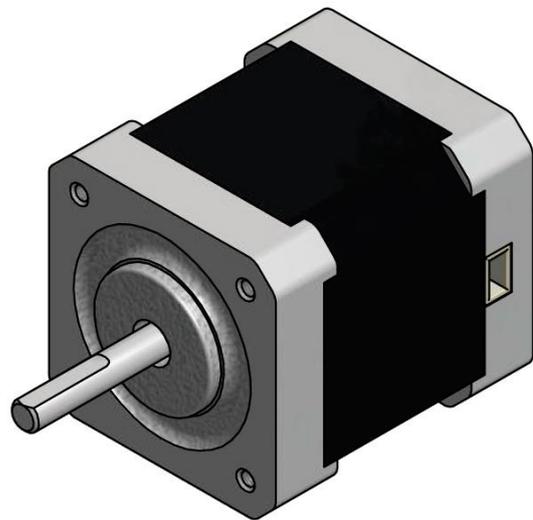
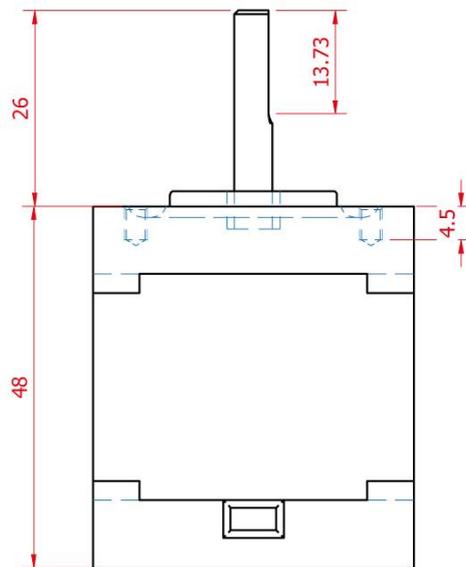
Anexo 9: Rueda

CANT.

9



DETALLE A
ESCALA 3:1



ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO



Escala 1:1

Material: -

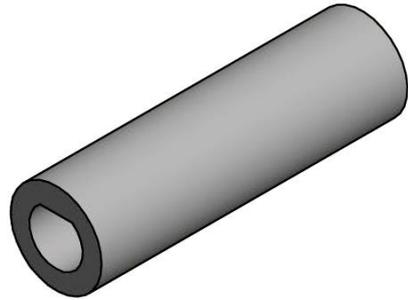
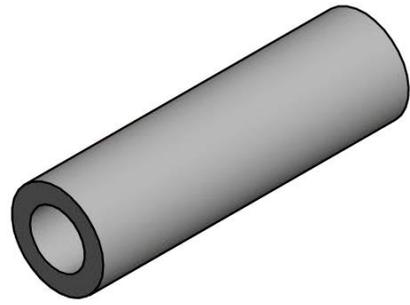
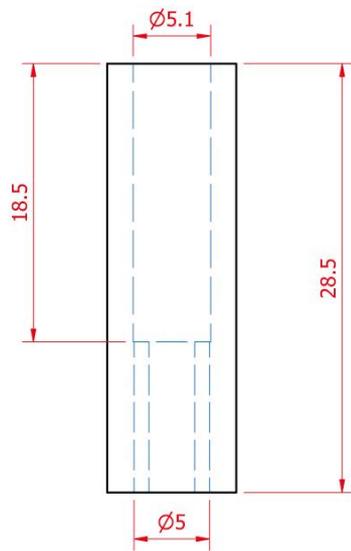
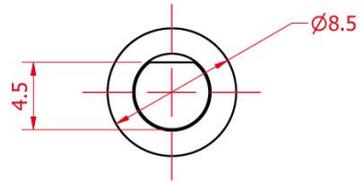
06/07/14

TITULO

Anexo 10: Motor Nema 17

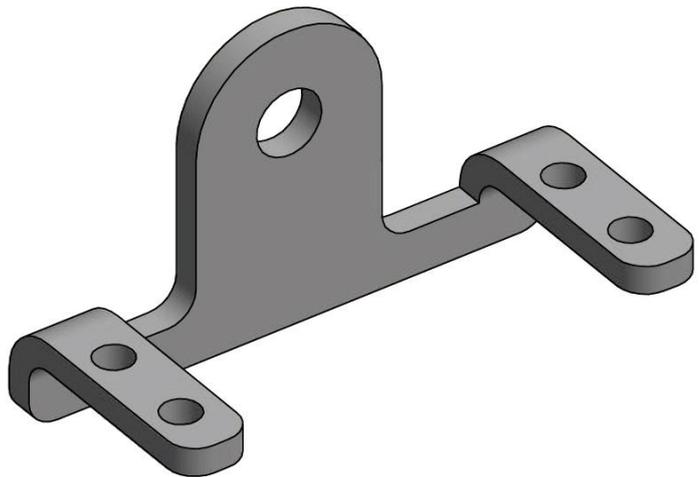
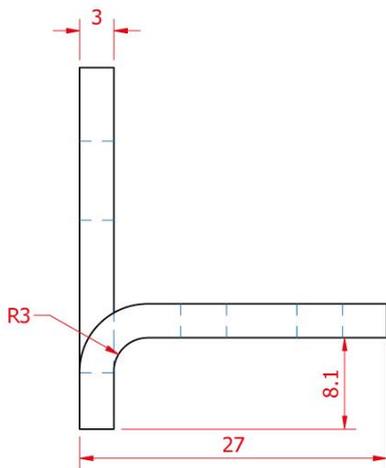
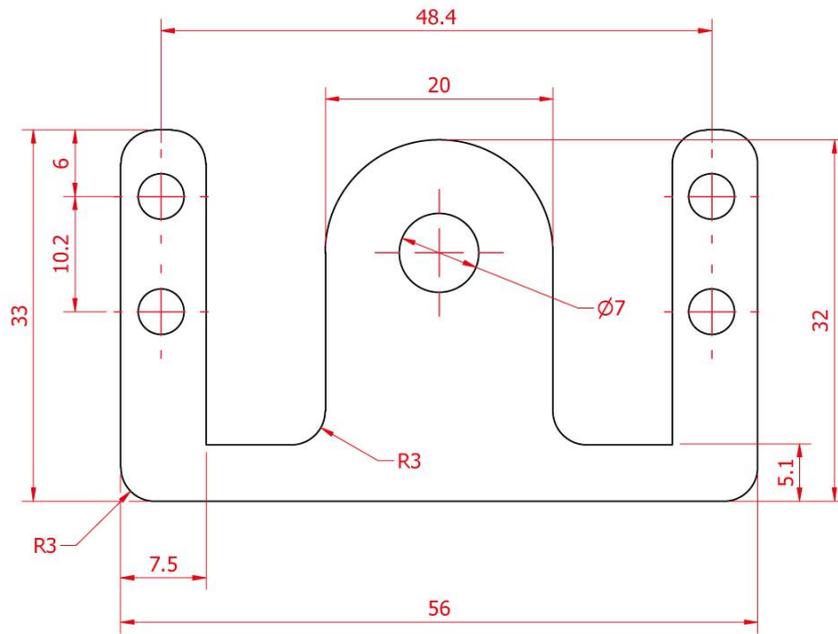
CANT.

3



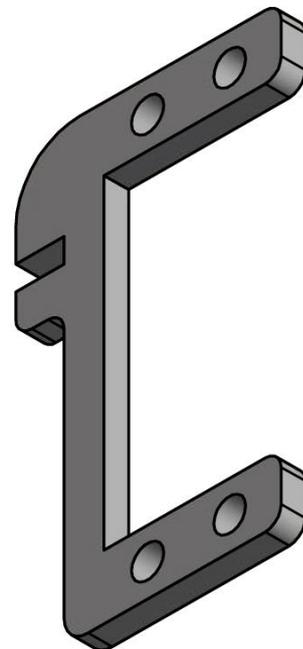
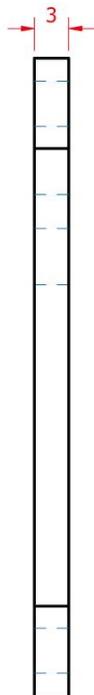
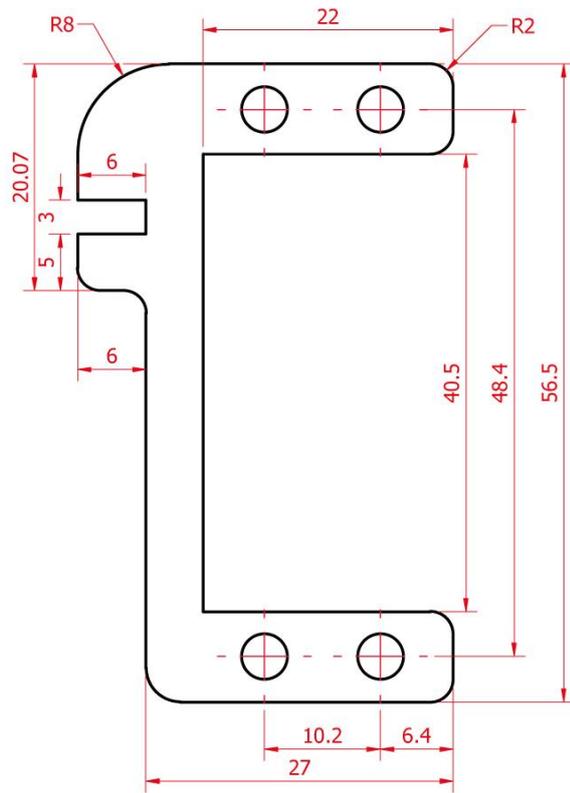
ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

	Escala 1.5:1	Material: Aluminio	04/09/14
	TITULO Anexo11: Adaptador rueda - motor Nema 17		



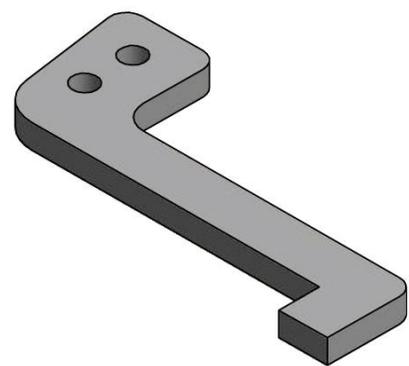
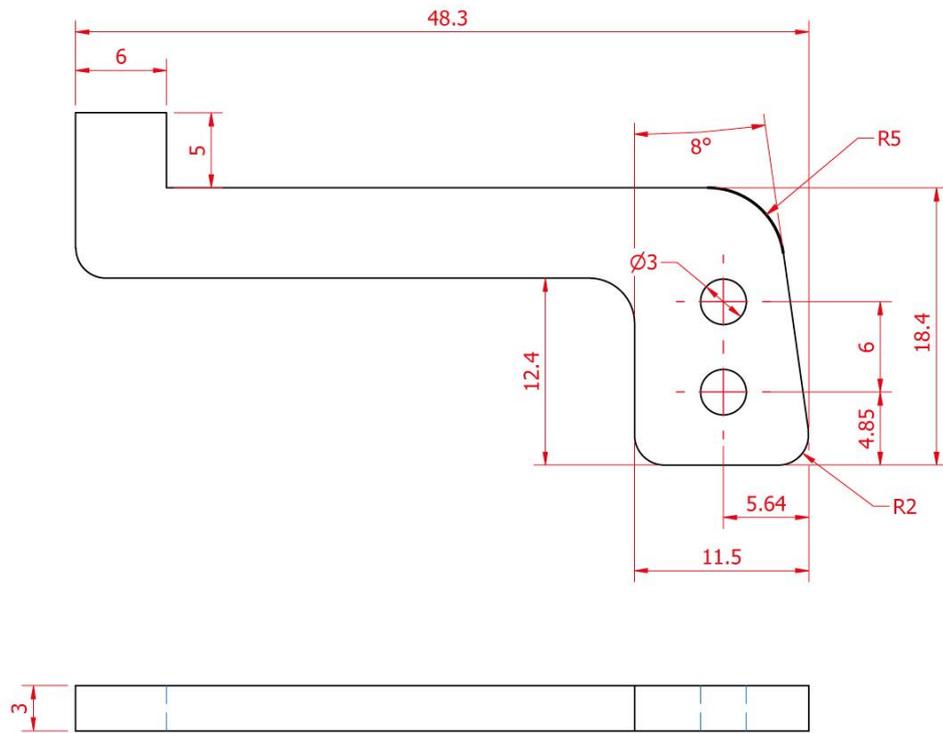
ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

	Escala 1.5:1	Material: Hierro	09/10/14
	TITULO Anexo 12: Porta servomotor 1		CANT. 1



ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

	Escala 1.5:1	Material: Hierro	09/10/14
	TITULO Anexo 13: Porta servomotor 2		CANT. 1



ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO



Escala 1.5:1

Material: Hierro

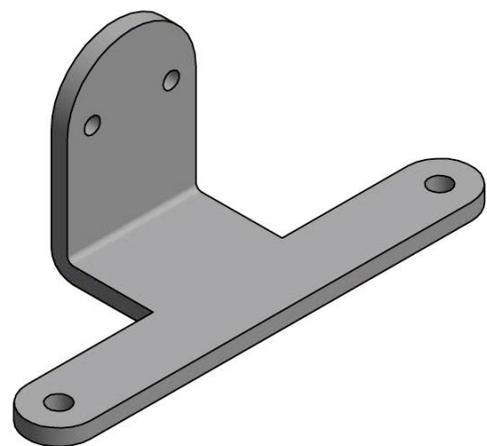
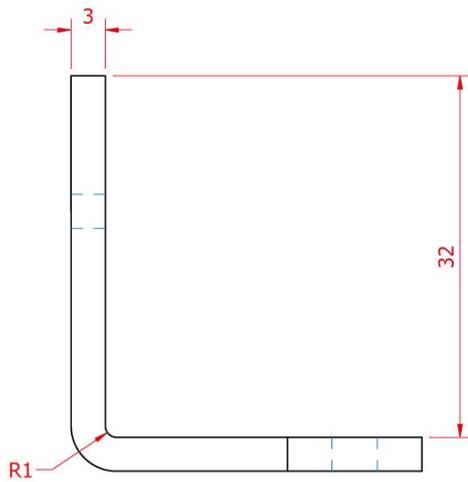
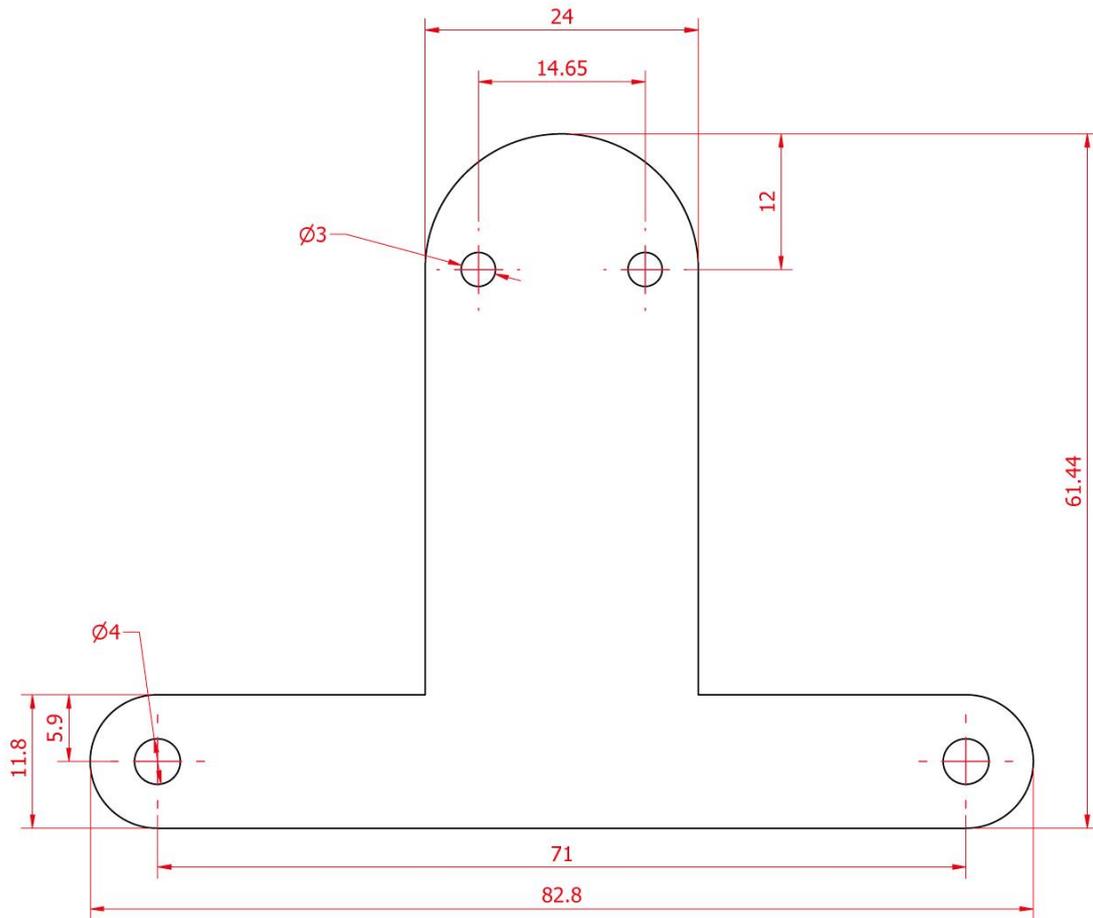
09/10/14

TITULO

Anexo 14: Unión servomotor 1

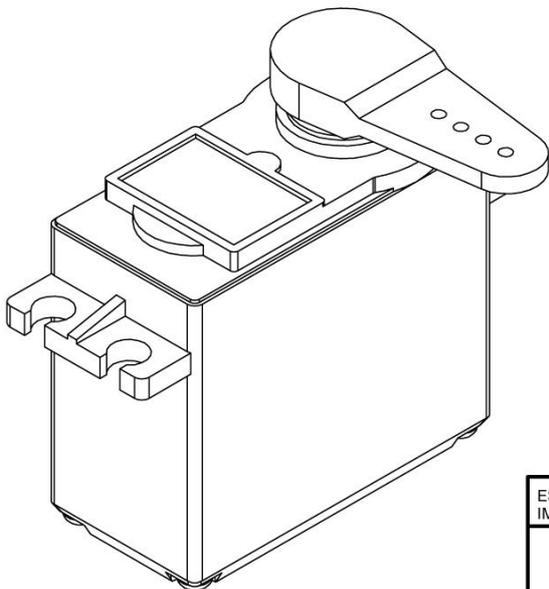
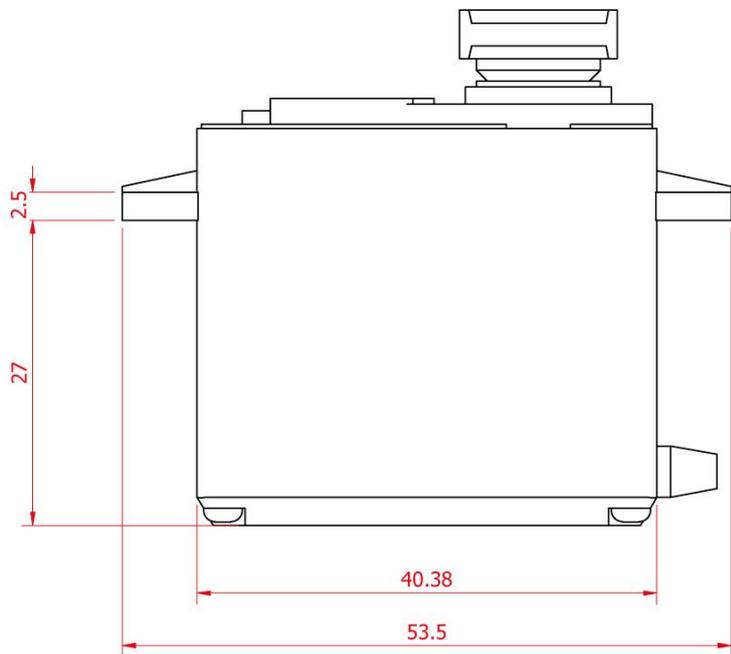
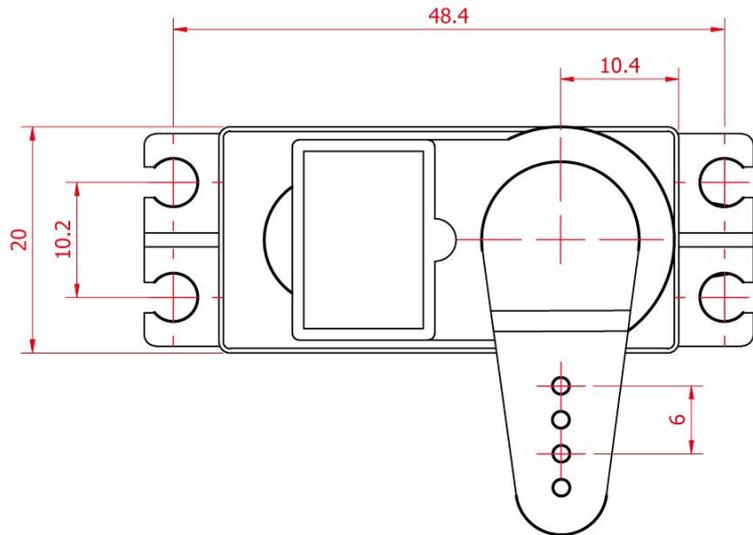
CANT.

1



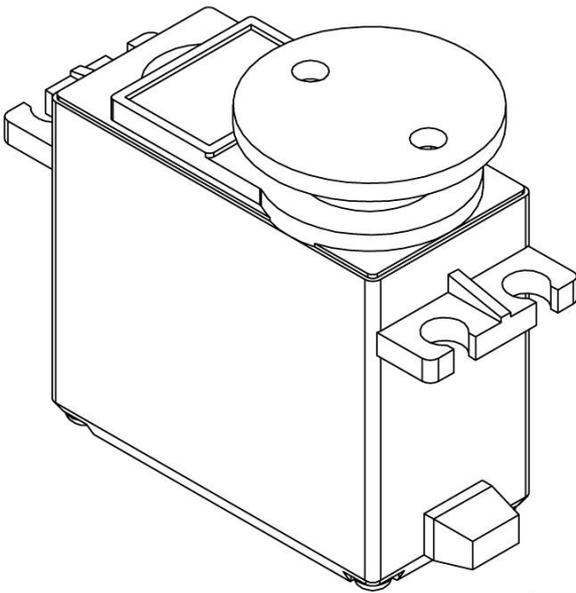
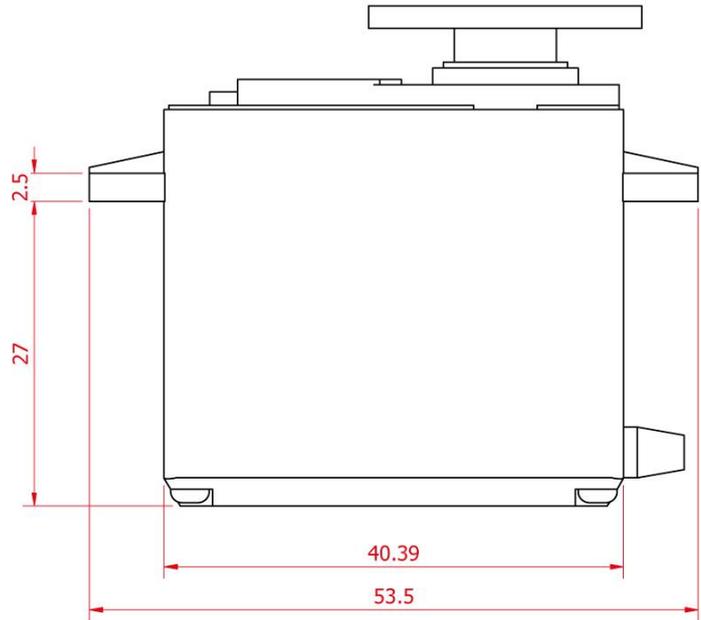
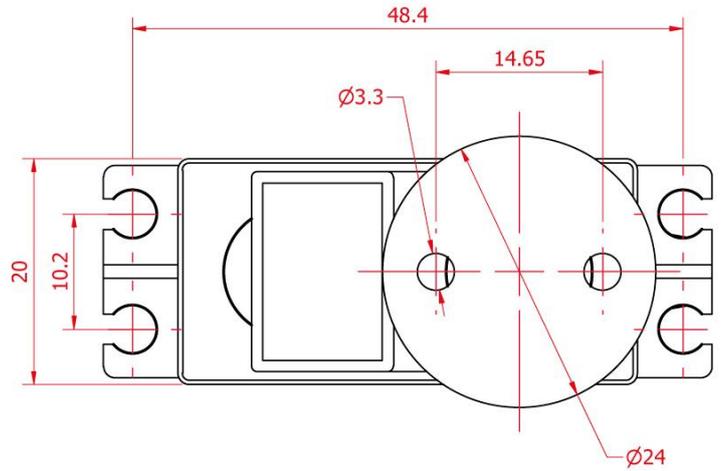
ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

 <p>UNIVERSIDAD DEL AZUAY</p>	<p>Escala 1.5:1</p>	<p>Material: Hierro</p>	<p>09/10/14</p>
	<p>TITULO</p> <p>Anexo 15: Unión servomotor 2</p>		<p>CANT.</p> <p>1</p>



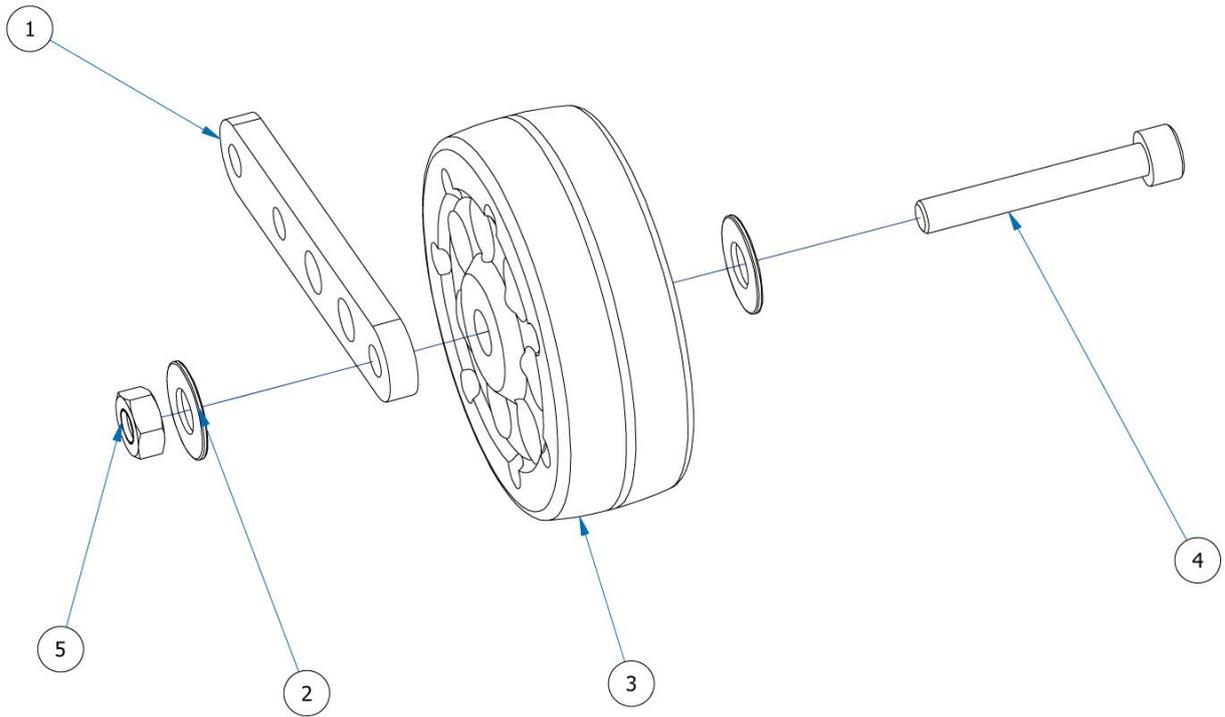
ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

 <p>UNIVERSIDAD DEL AZUAY</p>	<p>Escala 1.5:1</p>	<p>Material: -</p>	<p>09/10/14</p>
	<p>TITULO</p> <p>Anexo 16: Servo Motor HS-645MG 1</p>		<p>CANT.</p> <p>1</p>



ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

	Escala 1.5:1	Material: -	09/10/14
	TITULO Anexo 17: Servo Motor HS-645MG 2		CANT. 1



Lista de partes		
Item	Cantidad	Descripción
1	1	Link Simple
2	2	Arandela
3	1	Rueda
4	1	Perno allen M6X80
5	1	Tuerca M6 nylon

ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO



Escala 3:4

Material: -

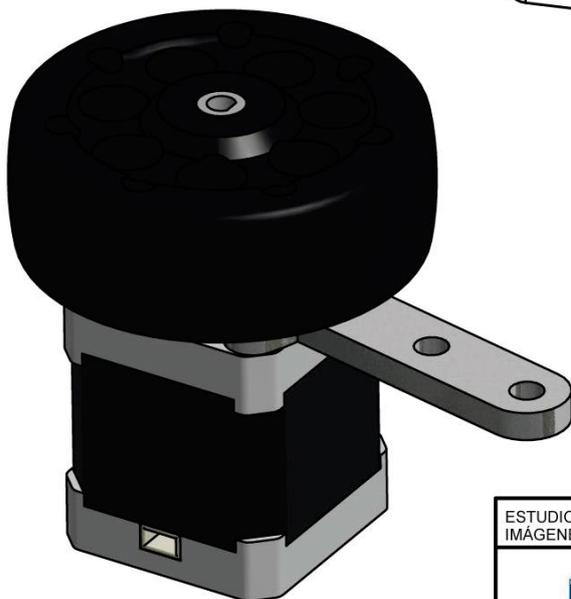
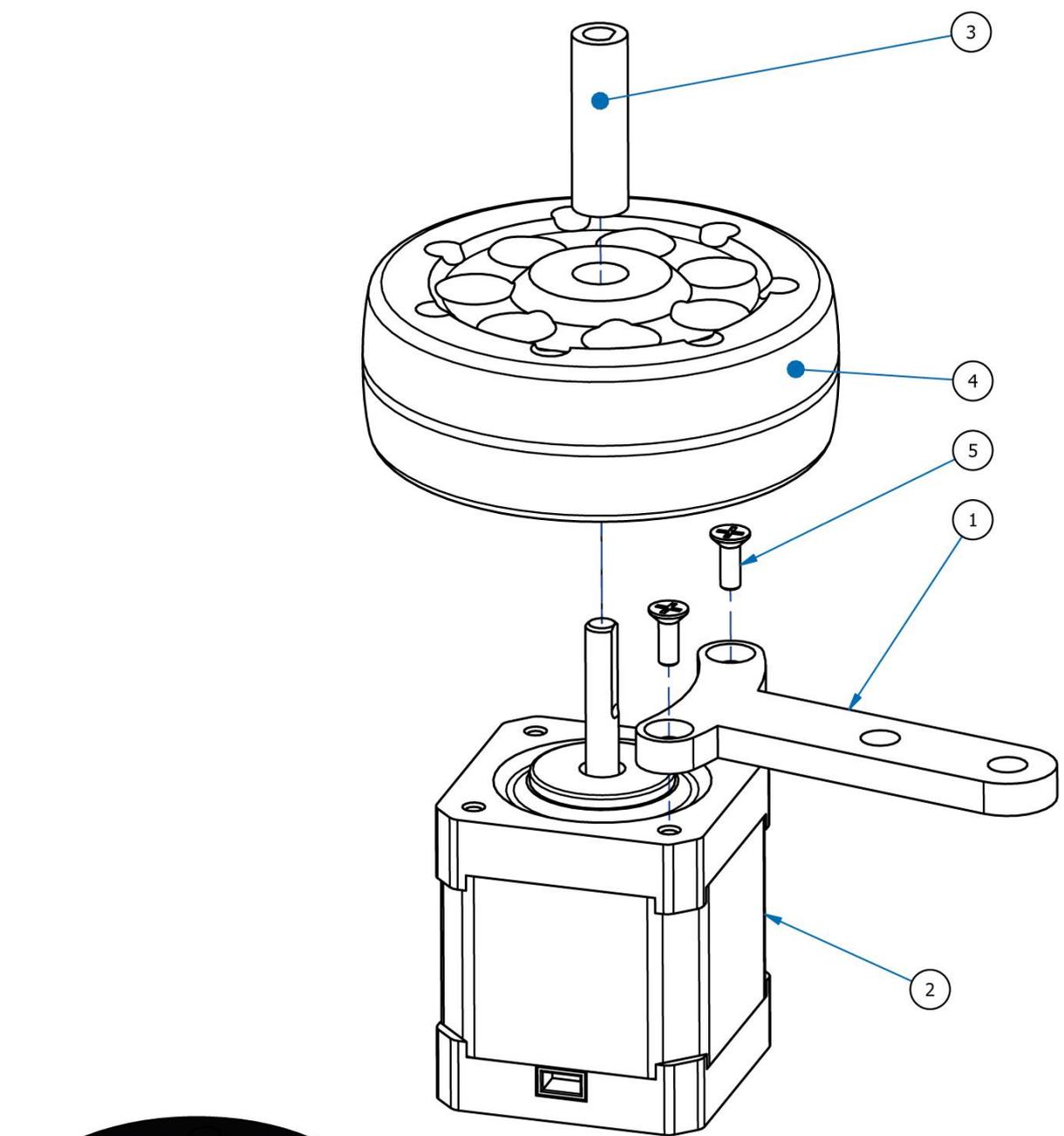
09/10/14

TITULO

**Anexo 18:
Ensamble rueda posterior**

CANT.

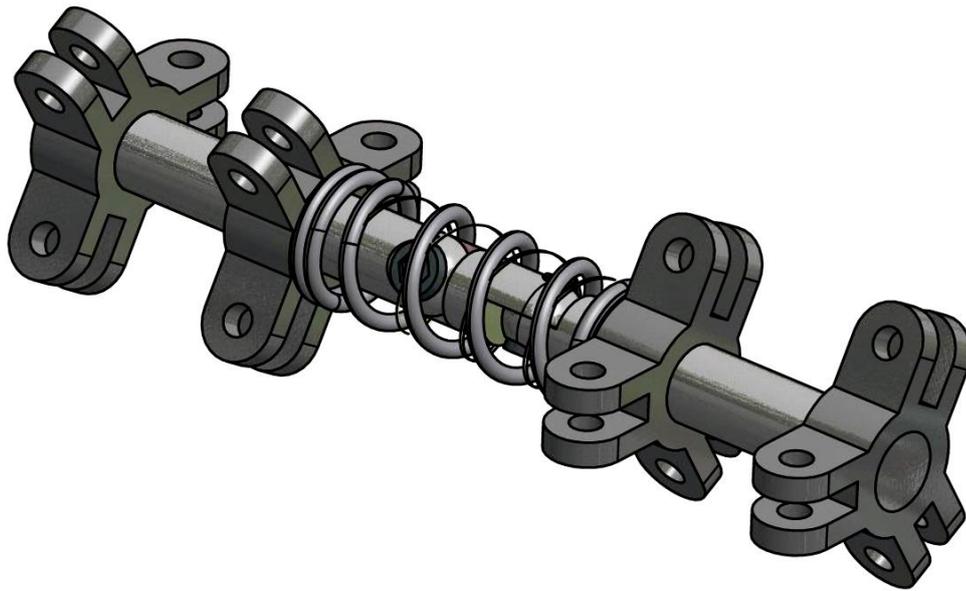
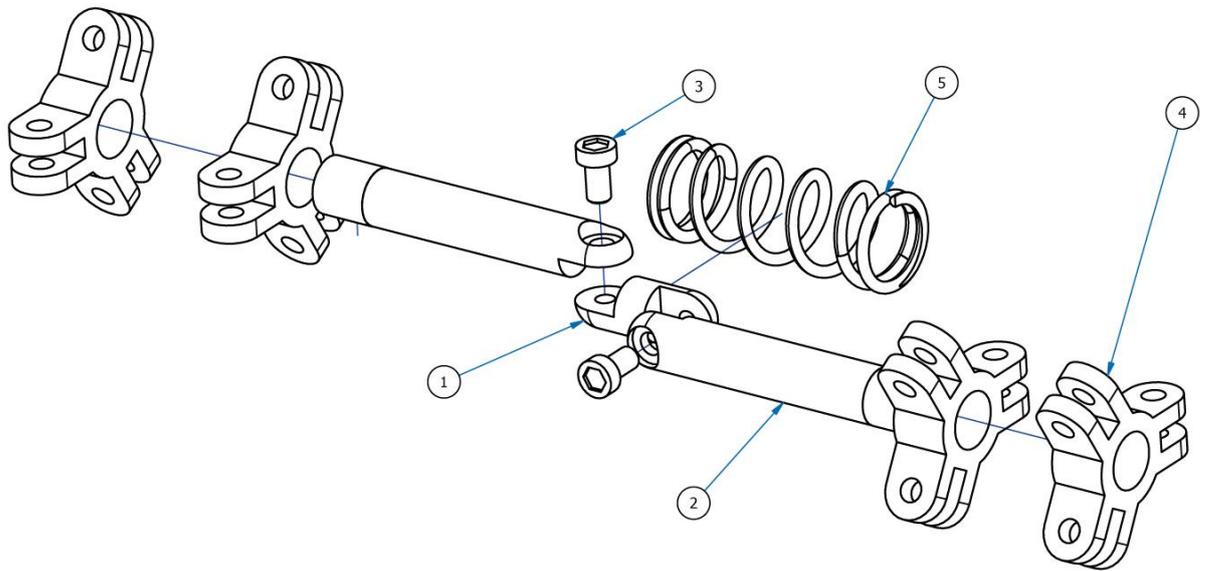
3



Lista de partes		
Item	Cantidad	Descripción
1	1	Link Motor
2	1	Motor NEMA 17
3	1	Adaptador Rueda - motor NEMA 17
4	1	Rueda
5	2	Tornillo avellanado M3

ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

	Escala 1:1	Material: -	04/09/14
	TITULO Anexo 19: Ensamble ruedas motor		



Lista de partes		
Item	Cantidad	Descripción
1	1	Union eje
2	2	Eje extremo
3	2	Perno allen M6 Corto
4	4	Union Triple
5	1	Resorte

ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO



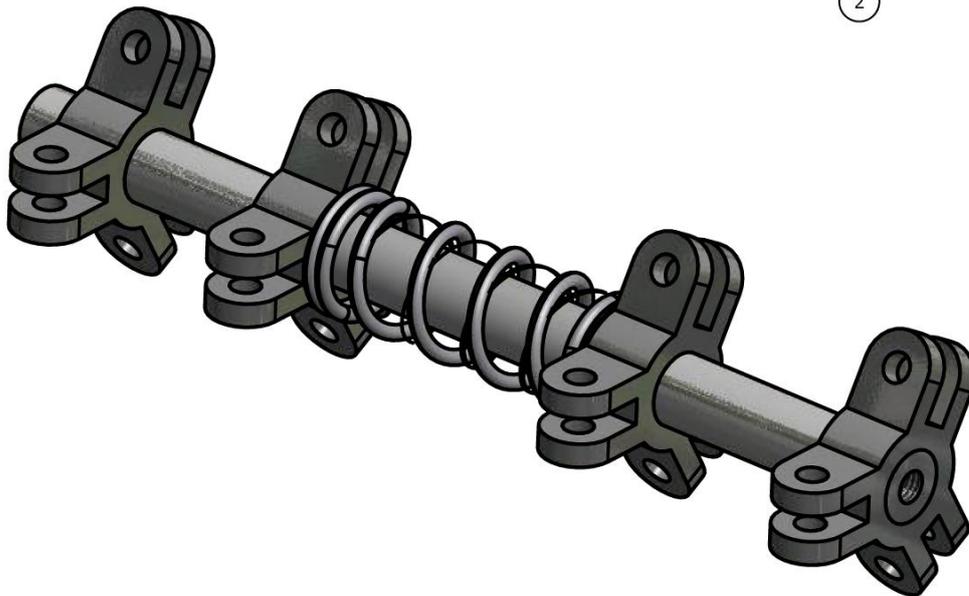
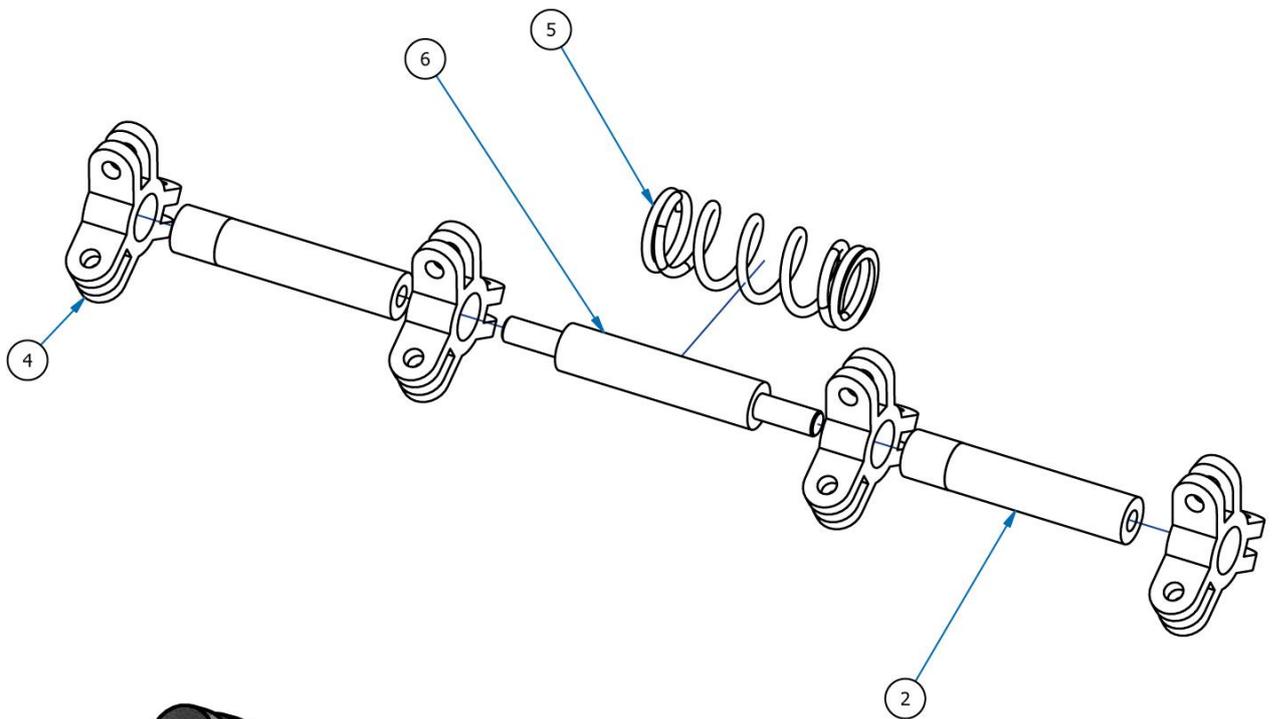
Escala 0.75:1

Material: -

04/09/14

TITULO
Anexo 20:
Ensamble Eje

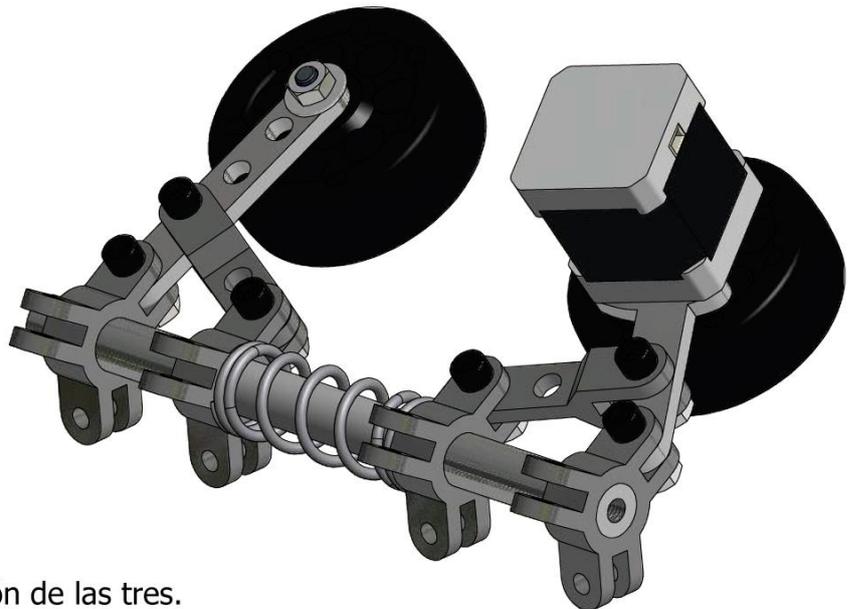
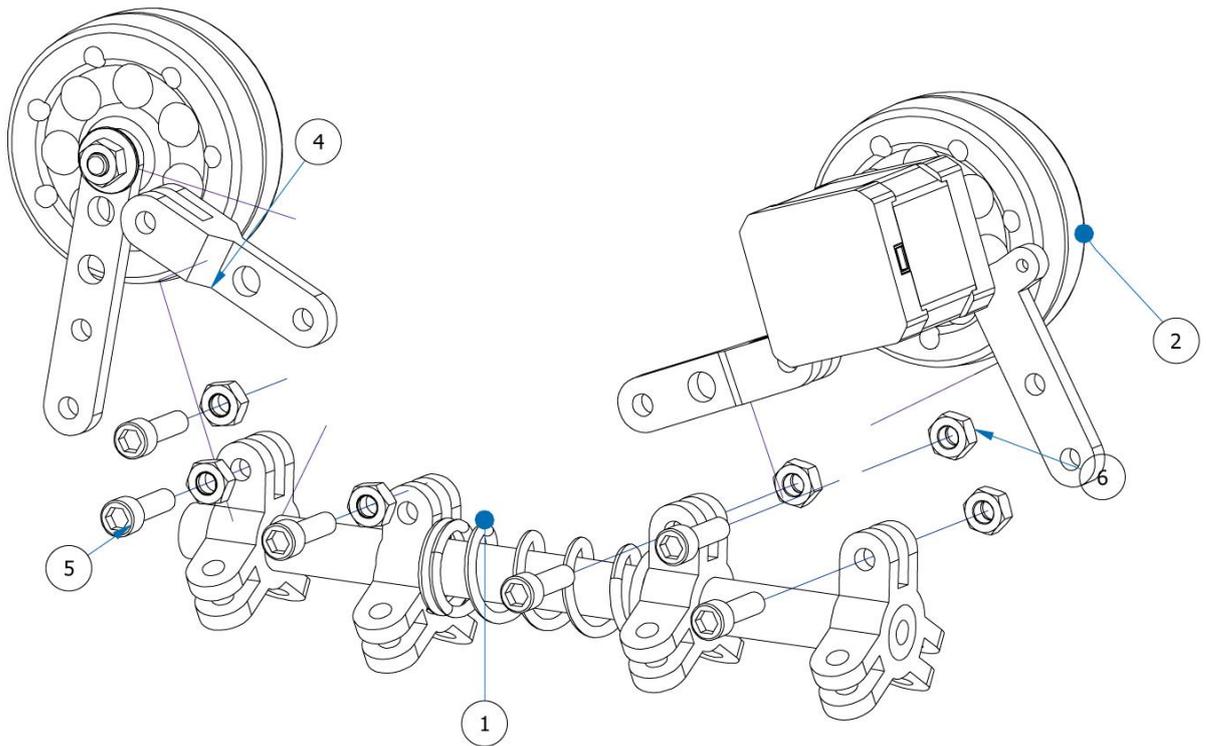
CANT.
1



Lista de partes		
Item	Cantidad	Descripción
2	2	Eje extremo V2
4	4	Union Triple
5	1	Resorte
6	1	Unión ejes rectos

ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

 UNIVERSIDAD DEL AZUAY	Escala 1:2	Material: -	09/10/14
	TITULO Anexo 21: Ensamble Eje V2		

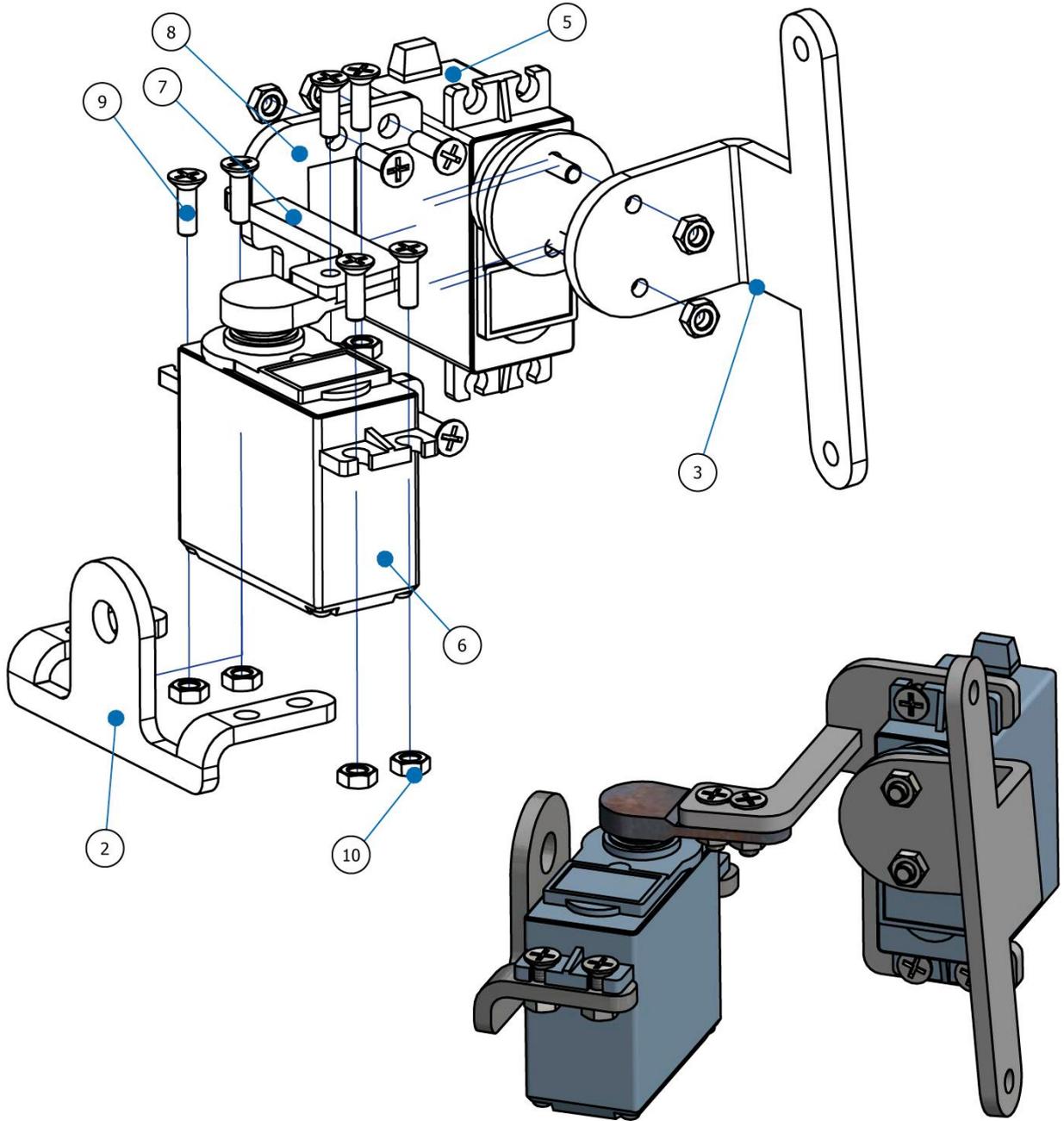


*Se muestra solo una sección de las tres.

Lista de partes		
Item	Cantidad	Descripción
1	1	Ensamble eje V2
2	3	Ensamble Motor
3	3	Ensamble ruedas
4	6	Link Lateral
5	18	Perno allen M6X30
6	18	Tuerca M6 nylon

ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO

	Escala 3:4	Material: -	09/10/14
	TITULO Anexo 22: Ensamble Cuerpo		



Lista de partes

Item	Cantidad	Descripción
2	1	Porta servomotor 1
3	1	Unión servomotor 2
5	2	Servo Motor HS-645MG 2
6	1	Servo Motor HS-645MG 1
7	1	Unión servomotor 1
8	1	Porta servomotor 2
9	14	Tornillo avellanado M3
10	14	Tuerca M6 nylon

ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO



Escala 1:1

Material: -

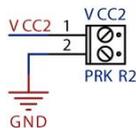
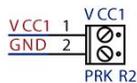
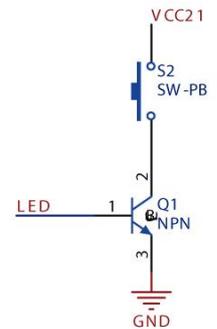
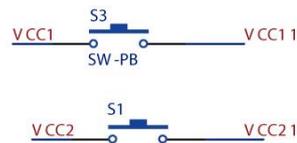
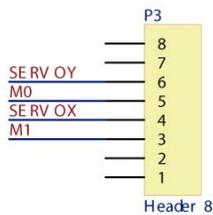
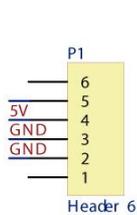
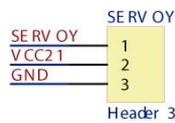
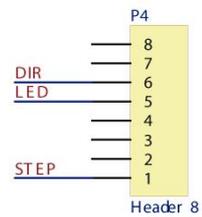
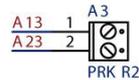
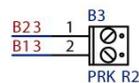
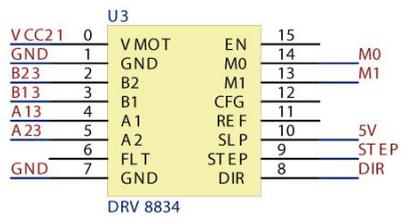
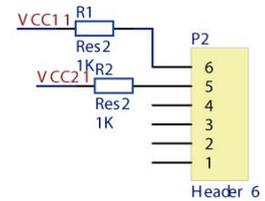
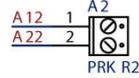
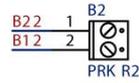
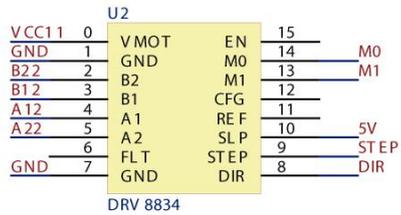
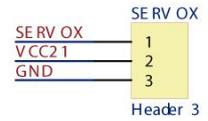
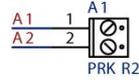
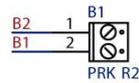
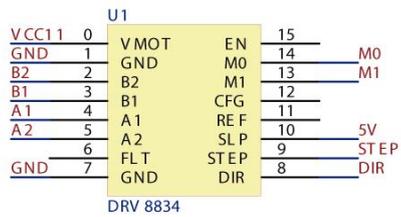
09/10/14

TITULO

Anexo 23:
Ensamble porta cámara

CANT.

1



ESTUDIO Y DISEÑO DE UN ROBOT EXPLORADOR DE DUCTOS MEDIANTE LA CAPTURA DE IMÁGENES EN TIEMPO REAL UTILIZANDO UN INTERFACE GRÁFICO REMOTO



Escala

Material:

09/10/14

TITULO
Anexo 24:
Driver motor

CANT.
1

Anexo 25 - Programación Arduino

```
1  /* TESIS: Control de un robot explorador de ductos*/
2
3  #include <SoftwareSerial.h> //Libreria para conexion serial
4  #include <Stepper.h> //Libreria para motor de pasos
5  #include <string.h>
6  #include <stdio.h>
7  #include <stdlib.h>
8  #include <Servo.h> //Libreria para Servomotores
9
10 #define RxD 7
11 #define TxD 6
12
13 int ejex = 0;
14 int ejey = 0;
15
16 #define DEBUG_ENABLED 1
17
18 SoftwareSerial blueToothSerial(RxD,TxD);
19
20 Servo servox;
21 Servo servoy;
22
23 // Variables para almacenar datos de posicion de camara tanto en x como en y
24
25 int vBateria = 0;
26
27 // Variables para almacenar la direccion
28 int direccion = 0;
29
30 // Variables para almacenar datos de velocidad
31 int velocidad = 0;
32
```

```
33 // Variables para led
34 int led = 0;
35
36 //Variable para controlar el sentido de giro
37 int dir = 13;
38
39 //Variable para controlar iluminacion
40 int luz = 12;
41
42 char inData[20];
43 byte index = 0;
44
45 //Matriz en donde se guardaran los valores tras recibirlos
46 //y separarlos para la posicion de la camara, velocidad,
47 //direccion y estado de los Leds
48 long values []= {0,0,0,0} ;
49
50 //Inicializacion variables motores de pasos
51 const int stepsPerRevolution = 200; // numero de pasos del motor
52 // Puertos de salida para el control del motor de pasos
53 Stepper myStepper(stepsPerRevolution, 8,9,10,11);
54
55 =====
56         SETUP
57 =====
58
59 void setup () {
60   Serial.begin(9600);
61   pinMode(RxD, INPUT);
62   pinMode(TxD, OUTPUT);
63
64   Serial.println("LISTO");
65   pinMode (luz , OUTPUT);
```

```
66 digitalWrite (luz , LOW);
67 servox.attach(3);
68 servoy.attach(5);
69 delay(100);
70 setupBluetoothConnection();
71
72 }
73
74 =====
75     LOOP
76 =====
77
78 void loop () {
79     //Subrutina para leer los datos y separar los valores enviados
80     //desde el computador
81     leer_separar();
82
83     if (direccion == 1)
84     {
85         digitalWrite (dir , HIGH);
86     }
87
88     else if (direccion == 0)
89     {
90         digitalWrite (dir , LOW);
91     }
92
93
94     if (led == 1)
95     {
96         digitalWrite (luz , HIGH);
97     }
98
```

```
99  else if (led == 0)
100 {
101  digitalWrite (luz , LOW);
102 }
103
104 switch (velocidad){
105  case 1:
106
107  lento();  //Subrutina para velocidad lenta
108  break;
109
110  case 2:
111
112  medio();  //Subrutina para velocidad media
113  break;
114
115  case 3:
116
117  rapido();  //Subrutina para velocidad rapida
118  break;
119
120  case 0:
121
122  paro();  //Subrutina para paro
123  break;}
124
125  lectura();  //Subrutina para leer valores de las baterias
126
127  Serial.flush();
128 }
129
130 //RUTINA PARA LEER VALORES DE LAS BATERIAS
131 void lectura ()
```

```
132 {
133  vBateria = analogRead (A0);
134  int voltaje = map (vBateria, 0, 1023, 0, 5);
135  Serial.print (voltaje);
136 }
137
138 //RUTINA PARA MOVER MOTORES A VELOCIDAD LENTA
139 void lento()
140 {
141  myStepper.setSpeed(70);
142  myStepper.step(20);
143  return;
144 }
145
146 //RUTINA PARA MOVER MOTORES A VELOCIDAD MEDIA
147 void medio()
148 {
149  myStepper.setSpeed(150);
150  myStepper.step(20);
151  return;
152 }
153
154 //RUTINA PARA MOVER MOTORES A VELOCIDAD RAPIDA
155 void rapido()
156 {
157  myStepper.setSpeed(200);
158  myStepper.step(20);
159  return;
160 }
161
162 //RUTINA PARA DETENER LOS MOTORES
163 void paro()
164 {
```

```
165 myStepper.setSpeed(0);
166 return;
167 }
168
169 //RUTINA PARA SEPARAR LOS DATOS RECIBIDOS DESDE EL COMPUTADOR
170 void leer_separar()
171 {
172   while(blueToothSerial.available() > 0)
173   {
174     char aChar = blueToothSerial.read();
175     //Lee el puerto serial para recibir los datos
176
177     if(aChar == 'h')
178     {
179       // El caracter h indica que es el fin de la cadena.
180       // Se pasa a Parse (analisis)
181       parse(); //SUBRUTINA PARA ARMAR LA MATRIZ
182       index = 0;
183       inData[index] = NULL;
184     }
185     else //En caso de que no se ha detectado el caracter de fin
186       //de cadena se debe seguir separando los datos
187     {
188       inData[index] = aChar;
189       index++;
190       inData[index] = '\0'; // Keep the string NULL terminated
191     }
192   }
193
194 }
195
196 //RUTINA DE ANALISIS,
197 void parse()
```

```
198 {
199   char separator[] = ",";      // el caracter que uso para separar los valores
200
201   char *result = NULL;
202   int index = 0;
203
204   result = strtok( inData, separator ); // aqui se separan los caracteres
205   while( (result != NULL) && (index < 6) ) {
206     values[index++] = atol(result); // aqui se convierten en valores
207
208     result = strtok( NULL, separator ); // aqui vacia result
209   }
210
211   // Almacenamos los valores en las variables correspondientes
212   direccion = values[0];
213   velocidad = values[1];
214   ejex  = values[2];
215   ejey  = values[3];
216   led   = values[4];
217
218   //Escribimos el angulo en cada uno de los ejes segun los que se recibio
219   servox.write (ejex);
220   servoy.write (ejey);
221 }
222
223 //RUTINA DE CONEXION BLUETOOTH
224 void setupBlueToothConnection()
225 {
226   blueToothSerial.begin(38400);
227   // Setear BaudRate al valor por defecto: 38400
228   blueToothSerial.print("\r\n+STWMOD=0\r\n");
229   // Configurar al Bluetooth para trabajr en modo Slave
230   blueToothSerial.print("\r\n+STNA=TESIS SARMIENTO DIAZ\r\n");
```

```
231 // Configurar el nombre del Bluetooth como "TESIS SARMIENTO DIAZ"
232 blueToothSerial.print("\r\n+STOAUT=1\r\n");
233 // Permitir dispositivos pareados que se conecten
234 blueToothSerial.print("\r\n+STAUTO=0\r\n");
235 // Permitir-Negar (1-0)autoconexion
236 delay(2000);
237 // Este delay es requerido para la conexion
238 blueToothSerial.print("\r\n+INQ=1\r\n");
239 // Permitir localizar el Bluetooth
240 Serial.println("Bluetooth Localizable");
241 delay(2000);
242 // Este delay es requerido para la conexion.
243 blueToothSerial.flush();
244 }
```