

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

**Facultad de Ingeniería en Mecánica y Ciencias de la
Producción**

"DISEÑO E IMPLEMENTACIÓN DE SISTEMA POR VOZ A UNA SILLA
DE RUEDAS ELÉCTRICA"

TRABAJO FINAL DE GRADUACIÓN

Materia Integradora

Previo la obtención del Título de:

INGENIERO MECÁNICO

Presentado por:

Álvaro Danilo Chévez Baquerizo

Israel Damián Silva Rosado

GUAYAQUIL - ECUADOR

Año: 2015

AGRADECIMIENTOS

Agradecemos a nuestros padres por el apoyo que nos brindado durante nuestro proceso estudiantil.

Agradecemos al CAMPRO por habernos permitido usar sus herramientas de trabajo e instalaciones al igual quedamos agradecidos con el Ing. Roca García por prestarnos las instalaciones del laboratorio de mecatrónica, que fue el lugar donde se realizó el proyecto.

DECLARACIÓN EXPRESA

“La responsabilidad del contenido desarrollado en la presente propuesta de la materia integradora corresponde exclusivamente al equipo conformado por:

Álvaro Danilo Chávez Baquerizo

Israel Damián Silva Rosado

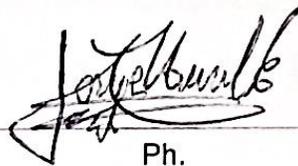
Jorge Luis Hurel Ezeta

Y el patrimonio intelectual del mismo a la Facultad de Ingeniería Mecánica y Ciencias de la Producción (FIMCP) de la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

Estamos también de acuerdo que el vídeo de la presentación oral es de plena propiedad de la FIMCP.



Israel Silva R.



Ph.

D. Jorge Hurel E.



Álvaro Chávez B.

RESUMEN

En este documento se presenta el desarrollo de un sistema controlado por comandos de voz para una silla de ruedas eléctricas. La implementación a la silla está diseñada para personas cuadripléjicas, que solo puedan tener movimiento de las cuerdas vocálicas. El objetivo principal es implantar el sistema en una silla de ruedas eléctricas para que una persona cuadripléjica pueda tener independencia al momento de desplazarse.

Los materiales utilizados para el desarrollo del producto fueron las tarjetas Arduino, EasyVR, un parlante, micrófono, relés, servomotores y eslabones. Los equipos utilizados fueron un computador portátil, un osciloscopio, multímetro. Se trabajó partiendo de un diseño conceptual en el cual se debió especificar el alcance, restricciones y requerimientos técnicos, para luego utilizar una matriz de decisión y escoger la mejor solución y de esa manera estimar los materiales, costos y el tiempo de construcción del sistema. El sistema lo dividimos en dos partes la parte mecánica y la parte electrónica.

Se obtuvo un sistema que permite a la silla de ruedas ser controlada por voz y los comandos que nos permitieron hacer esto fueron: adelante, atrás, izquierda, derecha, alto, encender, apagar, baja velocidad, media velocidad, alta velocidad, pitar. El tiempo de respuesta que le tomó a cada comando fue de un segundo y medio aproximadamente.

Palabras Clave: EasyVR, Arduino, Cuadripléjicas, Silla de Ruedas Eléctrica.

ABSTRACT

In this document we present the development of a voice controlled system for an electrical wheel chair, this adaptation is meant for quadriplegics whose only movements are from their vocal chords. The main objective of this work is to implement a system in the wheel chair that will make the people totally independent at the moment of moving.

The used materials for this product were Arduino cads, EasyVR, speaker, microphone, relays, actuators and links. The equipment we used were a laptop, an oscilloscope, multimeter, we worked using a theoretical design in which it was specified its scope, constraints and technical requirements for later being able to proceed with the decision matrix to choose the optimal solution. Thus, we estimated the materials, costs and construction time of the system which was divided into two parts the mechanical and the electrical part.

We obtained a system that allowed us to make the following movements: forward, backward, left, right, stop, on and off. The response time that took every command is of approximately a one second and a half.

Keywords: *EasyVR, Arduino, Quadriplegics, electrical wheel chair.*

ÍNDICE GENERAL

RESUMEN.....	I
ABSTRACT	II
ÍNDICE GENERAL	III
ABREVIATURAS.....	VI
SIMBOLOGÍA	VII
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS.....	X
ÍNDICE DE PLANOS.....	XI
CAPÍTULO 1.....	1
1. INTRODUCCIÓN	1
1.1. Descripción del problema.....	1
1.2. Objetivos	1
1.2.1. Objetivo General	1
1.2.2. Objetivos Específicos.....	1
1.3. Marco teórico	2
1.3.1. Cuadriplejía.....	2
1.3.2. Voz.....	3
1.3.3. Sillas de Rueda.....	3
1.3.4. Microcontroladores	4
1.3.5. La plataforma Arduino.....	5
1.3.6. La placa EasyVR	5
1.3.7. Micrófono	6
CAPÍTULO 2.....	8
2. METODOLOGÍA DE DISEÑO.....	8
2.1. Diseño Conceptual.....	8
2.1.1. Requerimientos Técnicos.....	8

2.1.2.	Restricciones	8
2.1.3.	Matriz de decisión	9
2.2.	Componentes para desarrollar el diseño	11
2.3.	Comandos por Voz	15
2.3.1.	Descripción de cada comando.....	15
2.3.2.	Comandos independientes del locutor.....	16
2.3.3.	Comandos dependientes del locutor.....	16
2.4.	Control de la parte Mecánica	18
2.4.1.	Movimientos básicos de la silla.....	18
2.4.2.	Control de los giros de los servomotores.....	21
2.5.	Control de la parte electrónica	22
2.5.1.	Control de encendido/apagado y velocidad.....	22
2.5.2.	Conexión de sensores de distancia	23
2.5.3.	Pantalla LCD.....	24
2.5.4.	Incorporación del micrófono y altavoz	25
2.5.5.	Adaptación del módulo bluetooth.....	26
2.6.	Conexión del Sistema con la tarjeta del Joystick	28
2.7.	Diagrama de Flujo.....	28
2.8.	Velocidades de la Silla	30
2.9.	Pruebas de efectividad del sistema.....	35
2.10.	Comparación entre Arduino y Bluetooth.....	35
2.11.	Costos	37
CAPÍTULO 3.....		38
3. RESULTADOS.....		38
CAPÍTULO 4.....		42
4. DISCUSIÓN Y CONCLUSIONES.....		42
4.1.	Conclusiones.....	44
4.2.	Recomendaciones	44

REFERENCIAS BIBLIOGRÁFICAS	46
APÉNDICES	48

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
FIMCP	Facultad de Ingeniería en Mecánica y Ciencias de la Producción
CAMPRO	Campo Avanzado de Mecanización y Prototipos
CPU	Unidad Central de Proceso
Max MSP	Lenguaje de Programación Visual
PWM	Modulación por ancho de pulso

SIMBOLOGÍA

m	Metro
cm	Centímetro
mV	Milivoltio
ms	Milisegundos
s	Segundos
°C	Grados Celsius
T	Temperatura
t	Tiempo
d	Distancia
V	Voltios
Ω	Ohmios
k Ω	kilo Ohmios
μ s	micro segundos
μ F	micro Faradios
ω	Velocidad angular
φ	Diámetro de la rueda
RPM	Revoluciones por minuto
R	Resistencia
V_s	Velocidad del sonido
C	Capacitancia
f	Frecuencia
Hz	Hercio
Kgf	Kilogramo – Fuerza
ΔV	Diferencia de Voltaje

ÍNDICE DE FIGURAS

Figura 1.1. Partes del Cuerpo afectadas por la enfermedad.....	2
Figura 1.2. Silla de Ruedas Convencional	3
Figura 1.3. Silla de Ruedas Eléctrica	4
Figura 1.4. Microcontrolador.....	5
Figura 1.5. Arduino UNO	5
Figura 1.6. Tarjeta EasyVR con micrófono y altavoz.....	6
Figura 1.7. Clasificación de los micrófonos	7
Figura 1.8. Características direccionales del micrófono	7
Figura 2.1. Etapas del diseño conceptual	8
Figura 2.2. Proceso para el movimiento de la silla	9
Figura 2.3. Proceso a seguir con Arduino	10
Figura 2.4. Proceso implementado un sistema mecánico	10
Figura 2.5. EasyVR Arduino Shield	11
Figura 2.6. Vista de las Conexiones del Shield	12
Figura 2.7 Esquema del Shield Con la tarjeta EasyVR	12
Figura 2.8. Modos de Operación del EasyVR	13
Figura 2.9. Modos de Operación del Shield	13
Figura 2.10. Pantalla inicial del programa EasyVR	14
Figura 2.11. Partes de un relé	14
Figura 2.12. Comandos principales del EasyVR Commander	17
Figura 2.13. Nombres asignados por el usuario.....	17
Figura 2.14. Dimensiones principales del Joystick	18
Figura 2.15. Dimensiones del servomotor	19
Figura 2.16. Ubicación de los servomotores respecto al joystick	19
Figura 2.17. Dimensiones de los eslabones.....	20
Figura 2.18. Características Técnicas del plástico estructural	21
Figura 2.19. Esquema de conexión de los servomotores con la EasyVR	22
Figura 2.20. Esquema de conexión del Relé.....	23
Figura 2.21. Esquema de conexión de los sensores de distancia.....	24
Figura 2.22. Esquema de conexión de Arduino nano y pantalla LCD	25
Figura 2.23. Esquema de conexión del micrófono y altavoz	25
Figura 2.24. Esquema de conexión del módulo bluetooth.....	26
Figura 2.25. Esquema Final de las conexiones del sistema por voz.....	27

Figura 2.26. Diagrama esquemático del funcionamiento del sistema acoplado a la silla de ruedas.....	28
Figura 2.27. Diagrama del proceso del sistema con los comandos	29
Figura 2.28. Relación matemática entre la Velocidad (RPM) y Voltaje	30
Figura 2.29. Relación Lineal entre Velocidad de la silla y voltaje.....	31
Figura 2.30. Separación de las ruedas de la silla.....	32
Figura 2.31. Relación cuadrática entre el giro de la silla y el voltaje	33
Figura 2.32. Calculadora de la Universidad de Tokio Okawa (Inglés).....	34
Figura 2.33. Tiempo de respuesta del sistema.....	34
Figura 3.1. Sistema de reconocimiento de voz que se acopla a joystick controlador.	38
Figura 3.2. Silla de ruedas Jazzy Jet 3 con el sistema por voz	39
Figura 5.1. Vista del Joystick Controlador	49
Figura 5.2. Joystick Controlador con sus principales componentes.....	50
Figura 5.3. Sensor de Distancia	51
Figura 5.4. Sensor de distancia con sus conexiones	51
Figura 5.5. Funcionamiento del sensor de distancia	52
Figura 5.6. Funcionamiento de un PWM	52
Figura 5.7. Ilustración de un micro servo	53
Figura 5.8. PWM servomotor.....	54

ÍNDICE DE TABLAS

Tabla 2.1. Matriz de Decisión	9
Tabla 2.2. Ángulos para cada servomotor.....	20
Tabla 2.3. Relación del Voltaje con los RPM	30
Tabla 2.4. Relación Voltaje vs. Velocidad Lineal de la Silla	31
Tabla 2.5. Relación entre el voltaje del potenciómetro y los giros de la silla.....	32
Tabla 2.6. Comandos y palabras designadas para el joystick	35
Tabla 2.7. Comandos y palabras designadas para el joystick controlador.....	35
Tabla 2.8. Comparación entre Arduino y Bluetooth.....	36
Tabla 2.9. Costos de los materiales y equipos utilizados.....	37
Tabla 3.1. Velocidades de la silla para cada comando	39
Tabla 3.2. Eficiencia para cada palabra utilizada	40
Tabla 3.3. Eficiencia para cada palabra corta utilizada	40
Tabla 3.4. Eficiencia utilizando reconocimiento de voz de Google.....	41
Tabla 3.5. Eficiencia utilizando reconocimiento de voz de Google en palabras cortas	41
Tabla 4.1. Tiempo de retorno en los sensores de distancia	43

ÍNDICE DE PLANOS

PLANO 1	Base de Madera
PLANO 2	Base lateral derecha
PLANO 3	Base inferior parte derecha
PLANO 4	Base inferior parte izquierda
PLANO 5	Base lateral izquierda
PLANO 6	Eslabón derecho del servomotor
PLANO 7	Eslabón izquierdo del servomotor
PLANO 8	Manivela del servomotor
PLANO 9	Plataforma de Madera para servomotores
PLANO 10	Tapa superior
PLANO 11	Ensamble del Sistema
PLANO 12	Vista en explosión del sistema con sus elementos principales

CAPÍTULO 1

1. INTRODUCCIÓN

1.1. Descripción del problema

Hoy en día existe una gran cantidad de personas con distintas discapacidades físicas que no les permiten desarrollarse plenamente dentro de la sociedad. Estas personas se encuentran bajo dicha circunstancia ya sea por condiciones congénitas o por causas accidentales y dependiendo del grado de su discapacidad, necesitan diferentes tipos de dispositivos que les faciliten el proceso de convivencia social de una forma más natural y sin sentirse excluidos de su ambiente inmediato. Estos dispositivos auxiliares tienen la característica en común de que con el paso del tiempo mejoran sus atributos tecnológicos y han permitido que estas personas se desarrollen de una manera incluyente, ayudándolas tanto física como moralmente.

Muchas de las nuevas tecnologías son muy caras y no cualquier persona puede recurrir a este tipo de ayuda. Por lo tanto, el desarrollar o implementar nuevos dispositivos a un costo menor es un objetivo muy importante y trascendental.

Con la finalidad de implementar herramientas más prácticas y accesibles, se llevó a cabo una investigación previa que sirvió de fundamento para utilizar un Arduino UNO y un módulo de reconocimiento de voz EasyVR que acoplados estos dispositivos permiten desarrollar un sistema controlado por comandos de voz.

1.2. Objetivos

1.2.1. **Objetivo General**

- ✓ Diseñar y construir un sistema que permita a una persona cuadripléjica, mediante comandos de voz, generar una orden que pueda ser utilizada para el control de una silla de ruedas eléctrica jazzy jet ultra.

1.2.2. **Objetivos Específicos**

- ✓ Que este sistema en construcción sea un elemento útil para nuestra sociedad.
- ✓ Utilizar un módulo de reconocimiento de voz en el sistema para controlar la silla de ruedas.

1.3. Marco teórico

1.3.1. Cuadriplejía

La cuadriplejía o tetraplejía, es la parálisis de los brazos, manos, tronco, piernas y órganos pélvicos. La cuadriplejía es causada por un daño a su médula espinal. Cuando la médula espinal se daña, se pierde la sensación y el movimiento. Los síntomas pueden depender del lugar y la severidad de la lesión en su médula espinal, en la figura 1.1 se aprecia los lugares del cuerpo que son afectados por esta enfermedad [1].

1.3.1.1. *Causas*

La cuadriplejía y la paraplejía con más frecuencia son causadas por lesiones en la columna vertebral pero también pueden ser causadas por enfermedades nerviosas como la esclerosis múltiple y la esclerosis lateral amiotrófica (Enfermedad de Lou Gehrig). La mayoría de las lesiones en la columna vertebral son causadas por accidentes (choques automovilísticos y lesiones deportivas).

El nivel del proceso de la lesión o la enfermedad determinará la presencia de cuadriplejía versus paraplejía. El daño a la columna vertebral en la base del cráneo o cuello conlleva a paraplejía. El daño debajo del cuello resulta en paraplejía [2].



Figura 1.1. Partes del Cuerpo afectadas por la enfermedad

1.3.2. Voz

La señal de voz es una onda de presión acústica que se genera voluntariamente a partir de movimientos de la estructura anatómica del sistema fonador humano. La producción de la voz comienza en el cerebro con la conceptualización de la idea que se desea transmitir, la cual se asocia a una estructura lingüística, seleccionando las palabras adecuadas y ordenándolas de acuerdo con unas reglas gramaticales. A continuación el cerebro produce los comandos nerviosos que mueven los órganos vocales para producir los sonidos. La unidad mínima de una cadena hablada es el fonema, el cual posee un carácter distintivo en la estructura de la lengua [3].

1.3.3. Sillas de Rueda

Una silla de rueda es una ayuda técnica que consiste en una silla adaptada con cuatro ruedas, como se muestra en la figura 1.2. Las sillas de ruedas están diseñadas para permitir el libre desplazamiento de personas, con problemas de locomoción o con movilidad reducida debido a una lesión o enfermedad física.



Figura 1.2. Silla de Ruedas Convencional

1.3.3.1. Clases de Sillas

Existen dos clases de sillas de ruedas:

- ✓ Manuales

Que son impulsadas por el propio ocupante que hace girar las ruedas traseras empujando los aros acoplados en el exterior de estas. Se las fabrican de materiales ligeros como aluminio y el titanio al carbono para darle mayor durabilidad.

✓ Eléctricas

Este tipo de sillas son impulsadas por motores que son accionados por baterías de 40 o 50 amperios recargables. La persona controla la silla por medio de un joystick y un pequeño panel de control que le da el acceso a configurar la velocidad, como se muestra en la figura 1.3. Para personas que ni puedan utilizar las manos existen dispositivos controlables por la voz, pero son muy costosos. El diseño de sillas de ruedas eléctricas puede ser clasificado por el sistema de accionamiento/chasis, batería, controlador, asiento, y el uso [4].



Figura 1.3. Silla de Ruedas Eléctrica

La mayoría de los modelos de sillas son altamente adaptables: tamaño y posición del asiento y respaldo, apoyabrazos y apoyapiés regulables y extraíbles.

1.3.4. Microcontroladores

Un micro controlador es un circuito integrado o chip, como se muestra en la figura 1.4, que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de una computadora completa en un solo circuito integrado programable y se destina a gobernar una sola tarea con el programa que reside en su memoria. Sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar [5].

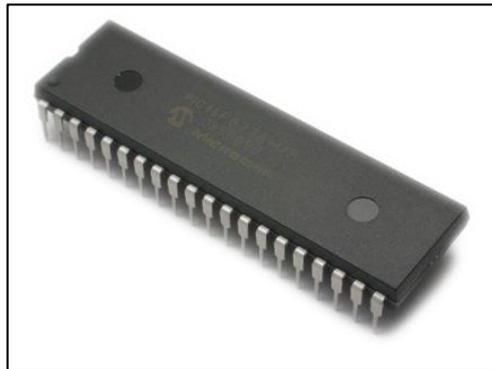


Figura 1.4. Microcontrolador

1.3.5. La plataforma Arduino

Es una plataforma de computación física de código abierto basado en una placa electrónica simple y un entorno de desarrollo para escribir software, como se muestra en la figura 1.5. Este se puede utilizar para desarrollar objetos interactivos, teniendo seis entradas analógicas y trece entradas digitales para una variedad de interruptores o sensores, y el control de una variedad de luces, motores, y otras salidas físicas.

Los proyectos Arduino pueden ser independientes o pueden comunicarse con el software que se ejecuta en el ordenador (por ejemplo, Flash, Processing, MaxMSP). Al ser una plataforma libre, tanto su diseño como su distribución, puede utilizarse para el desarrollo de cualquier tipo de proyecto sin necesidad de licencia [6].

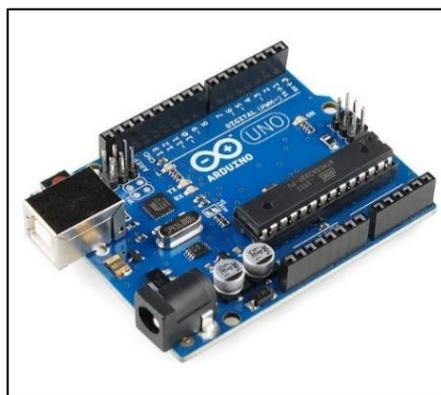


Figura 1.5. Arduino UNO

1.3.6. La placa EasyVR

La figura 1.6 muestra la placa EasyVR 3.0, la cual consiste en un módulo de reconocimiento de voz multiuso fabricado por ROBOTECH SRL, diseñado para añadir fácilmente capacidades de reconocimiento de voz versátil, confiable y rentable para cualquier aplicación.

El módulo EasyVR 3 se puede utilizar con cualquier host que tenga una interfaz UART alimentado de 3.3 – 5 V, tales como Arduino. Ejemplos de aplicaciones de uso son la automatización del hogar, tales como conmutadores de voz para controlar la luz, cerraduras, cortinas o aparatos de cocina [7].



Figura 1.6. Tarjeta EasyVR con micrófono y altavoz

1.3.7. Micrófono

Un micrófono es un elemento capaz de captar ondas sonoras convirtiendo la potencia acústica en eléctrica de similares características ondulatorias. Para ello se necesita la combinación escalonada de dos tipos de transductores. El primero de ellos consiste en una fina lámina, denominada diafragma. Su misión es transformar las variaciones de presión en vibraciones mecánicas, es por tanto un transductor mecano acústico. El segundo transforma las vibraciones mecánicas recibidas en magnitudes eléctricas, es por tanto un transductor electromecánico. El conjunto de los dos transductores puede considerarse como uno electro acústico [8].

1.3.7.1. Clasificación

Los micrófonos pueden clasificarse atendiendo a diversas características físicas tales como presión, magnetismo, como se muestra en la figura 1.7.

<u>Tipo de transductor mecanoacústico</u>	<u>Tipo de transductor electromecánico</u>
De presión: el diafragma está situado en uno de los extremos de una cámara cerrada; responde a los cambios de presión externa.	Electrostático
De gradiente: se obtiene abriendo la cámara y exponiendo ambos lados del diafragma a la presión sonora; responde a las diferencias de presión entre la parte anterior y posterior.	Piezoeléctrico
De presión y gradiente: se cierra parcialmente la cámara	Dinámico
De interferencia	Magnético
	De carbón o resistencia variable

Figura 1.7. Clasificación de los micrófonos

1.3.7.2. Características direccionales

La respuesta direccional de cualquier micrófono, llamada también directividad, es el cociente entre su sensibilidad en una dirección y su sensibilidad máxima, en la figura 1.8 se muestra las características de la respuesta direccional de un micrófono.

<u>Directividad</u>
De presión: son omnidireccionales, es decir tienen una respuesta similar para todas las direcciones del espacio
De gradiente: presentan respuestas de tipo bidireccional (en 8). Su máxima sensibilidad se presenta entre 0 y 180 grados.
De presión y gradiente: por combinación se obtienen diagramas polares intermedios (cardioides, hipercardioides y supercardioides).
De interferencia: muy direccionales, concebidos para enfocar un sonido determinado, consiguiendo eliminar todos los que se producen en su entorno.

Figura 1.8. Características direccionales del micrófono

CAPÍTULO 2

2. METODOLOGÍA DE DISEÑO

2.1. Diseño Conceptual

Se parte de los requerimientos, especificaciones técnicas y las restricciones. En la figura 2.1 se muestra las etapas que se sigue en el diseño conceptual.

Etapas del diseño conceptual

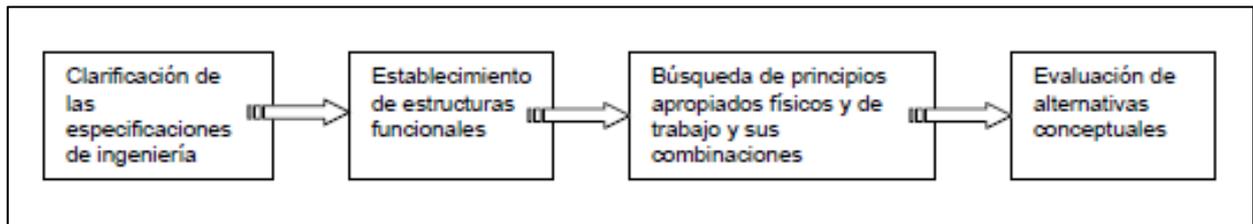


Figura 2.1. Etapas del diseño conceptual

2.1.1. **Requerimientos Técnicos**

- Adaptable a cualquier tipo de voz
- Módulo Reconocimiento de Voz
- Micrófono condensador unidireccional tipo cardioide
- Adaptable a una persona cuadripléjica
- Sistema Electro - Mecánico para accionar Joystick

2.1.2. **Restricciones**

- Micrófono no debe estar alejado de la persona
- El sistema debe ser de bajo costo
- Sistema mecánico para accionar el joystick
- No se puede alterar internamente la silla de ruedas
- Sistema a instalar debe ser adaptable
- La persona debe ser entrenada previo al manejo de la silla

Dada los requerimientos y especificaciones se tiene el proceso que se muestra en la figura 2.2.



Figura 2.2. Proceso para el movimiento de la silla

Para resolver el problema de la “X” se tiene que escoger una tarjeta para esto se utilizan tres alternativas.

Alternativas

Para definir las alternativas se utilizó la matriz de decisión, se tienen que basar en qué tipo de tarjeta se debe utilizar ya que es la que dará el control del joystick acoplado a la silla.

Arduino-EASY VR: Es una tarjeta factible de conseguir además posee una programación simple.

Arduino-Android Bluetooth: Para utilizar esta opción se necesita un dispositivo bluetooth y un celular para así poder mandar las órdenes al joystick.

Raspberry: La tarjeta Raspberry es complicada de conseguir en nuestro país y la programación no es simple.

2.1.3. Matriz de decisión

En la tabla 2.1 se muestra las alternativas y los factores con sus respectivos pesos.

Tabla 2.1. Matriz de Decisión

	SEGURIDAD (30%)	TIEMPO DE RESPUESTA (25%)	ESTETICA (15%)	FACILIDAD DE CONSTRUCCIÓN (30%)	
ARDUINO –EASY VR	27	20	12	21	80
ARDUINO-ANDROID BLUETOOH	24	20	10,5	22,5	77
RASPERRY	25,5	20	12	21	78,5

La tarjeta a utilizar será la Arduino, ya que es la que mejor puntuación presenta. Con la tarjeta Arduino se acoplará un módulo de voz llamado EasyVR, que permitirá darle las órdenes al Arduino mediante comando por voz. De esta forma se envían señales analógicas (desde el micrófono) para luego transformarlas en señales digitales con las tarjetas EasyVR – Arduino para luego esos pulsos acoplarlos con la tarjeta del joystick de la silla.

Algunos aspectos del hardware EasyVR:

- Espacio para 32 comandos de voz.
- Los comandos de voz se organizan en grupos. Para poder realizar cualquier operación sobre un comando de voz hay que indicar el grupo al que pertenece y la posición del comando de voz dentro del grupo.
- Los comandos de voz son dependientes del usuario y por tanto precisan ser entrenados para que puedan ser reconocidos. El sistema por tanto debe permitir entrenar los comandos de voz.
- Se recomienda no entrenar los comandos de voz más de 2 o 3 veces.

Interfaces hardware: La comunicación entre la placa de desarrollo Arduino y el PC se la debe llevar a cabo a través de un puerto USB.

Interfaces software: El control por voz utilizará el sistema propio del Arduino que es compatible con Windows.

Una vez tomada la decisión de que tarjeta se va a utilizar se tiene el siguiente proceso, que se muestra en la figura 2.3.

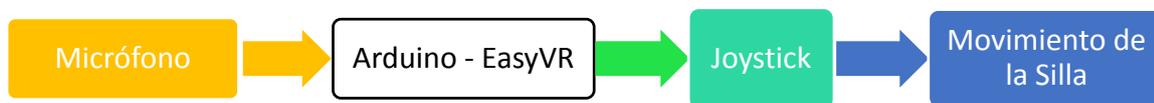


Figura 2.3. Proceso a seguir con Arduino

La forma de realizar el diseño será primero controlando el joystick para así poder realizar los cuatro movimientos básicos (adelante, atrás, izquierda, derecha).

Para accionar la palanca del joystick se utilizara un sistema mecánico. Se plantean dos opciones para poder hacer dicha acción.

- Servomotores acoplados con eslabones
- Sistema de engranes acoplados a un servomotor

En la figura 2.4, se muestra el proceso a seguir incluyendo el sistema mecánico.



Figura 2.4. Proceso implementado un sistema mecánico

Luego se enfocará en resolver el problema de encendido - apagado del joystick y los cambios de velocidad.

Adicionalmente se colocaran dos sensores de distancia uno en la parte posterior de la silla y otro en la parte frontal.

2.2. Componentes para desarrollar el diseño

• Micrófono

El micrófono que se utilizara es un micrófono omnidireccional electret marca Genius modelo HB – 02B con las siguientes características.

- Se adapta confortablemente a la oreja
- Conector del auricular estéreo 3.5mm
- Frecuencia de respuesta del auricular: 20~20KHz
- Impedancia del auricular: 32 OHM / 1KHz
- Sensibilidad del auricular: 76dB±4dB
- Conector del micrófono estéreo 3.5mm
- Frecuencia de respuesta del micrófono: 80Hz ~ 16KHz
- Impedancia del micrófono: 2.2KOhm / DC 4,5V

• Tarjeta Arduino

• Tarjeta de Reconocimiento de voz EasyVR

• EasyVR Arduino Shield

- El Shield es una placa que permite acoplar la tarjeta EasyVR con la tarjeta Arduino sin necesidad de usar algún cable. En la figura 2.5 se muestra el Shield (placa azul) acoplada con la tarjeta Arduino UNO y la EasyVR.

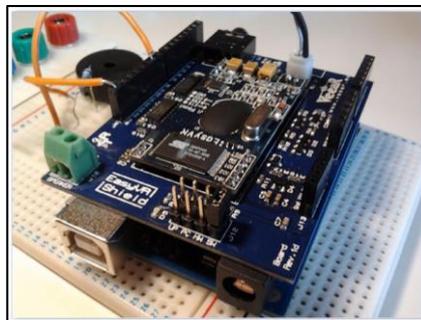


Figura 2.5. EasyVR Arduino Shield

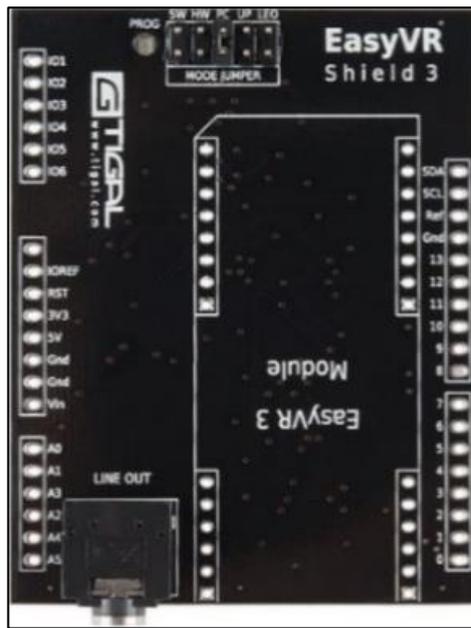


Figura 2.6. Vista de las Conexiones del Shield

- Mediante este acople es posible programar ambos dispositivos con la plataforma virtual de Arduino, haciendo uso de la librería EasyVR. En la figura 2.6, se muestra las conexiones del Shield. El Shield para realizar dicha relación interconecta internamente los dos dispositivos, conectando los puertos ERX y ETX de la EasyVR a cualquier par de pines multipropósitos del Arduino.

En la figura 2.7 se muestra el significado de cada uno de los pines del Shield.

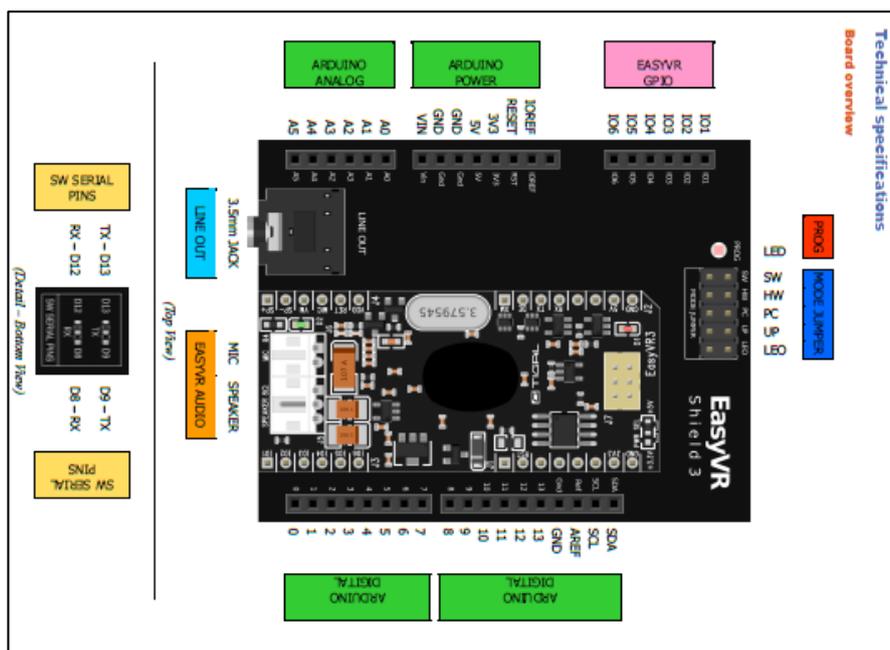


Figura 2.7 Esquema del Shield Con la tarjeta EasyVR

Los modos de operación que EasyVR acoplado al Shield puede entrar son UP mode, PC mode, HW mode y SW mode, configurables en la sección MODE JUMPER del Shield, en la figura 2.8 se muestra una tabla que muestra los modos de operación del EasyVR.

Pin assignment		
Group	Pin	Description
● ARDUINO HEADERS	-	Arduino UNO-R3 Shield interface, pass-through connectors (Pins 0-1 are in use when J12 is set to UP, PC, HW or LEO) (Pins 12-13 or 8-9 are in use when J12 is set to SW)
● EASYVR AUDIO	-	Audio cables connectors of the EasyVR 3 module (microphone and speaker)
● LINE OUT	-	3.5mm stereo/mono jack (16Ω - 32Ω headphones or line-level output)
● MODE JUMPER	SW	Arduino Software Serial (connected to pins 12-13 or 8-9)
	HW	Arduino Hardware Serial (connected to pins 0-1)
	PC	PC Mode (Arduino disabled, EasyVR in command mode)
	UP	Update Mode (Arduino disabled, EasyVR in boot mode)
	LEO	Leonardo Update (Arduino enabled, EasyVR in boot mode)
● PROG	-	Red light indicator for Flash programming modes (UP and LEO)
● SW SERIAL PINS	RX	Use resistor to select Software Serial RX pin: 12 or 8
	TX	Use resistor to select Software Serial TX pin: 13 or 9
● EASYVR GPIO	IO1	General purpose I/O as found on the embedded EasyVR 3 module (referenced at the internal VDD logic level – see note below)
	IO2	
	IO3	
	IO4	
	IO5	
	IO6	

Note: The General Purpose I/O lines (IO1-IO6) are at nominal 3.0VDC level. Do not connect higher voltages directly to these pins!

Figura 2.8. Modos de Operación del EasyVR

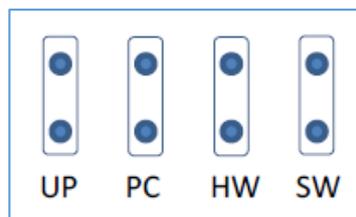


Figura 2.9. Modos de Operación del Shield

La figura 2.9 muestra los modos de operación del Shield que acoplado con la tarjeta EasyVR permiten trabajar en los siguientes modos:

PC Mode: En esta posición se permite la comunicación serial entre el EasyVR y el software EasyVR Commander, en este modo se permite realizar la fase de entrenamiento de comandos de voz que el usuario decida.

SW Mode: En este modo se puede programar y monitorear el módulo de reconocimiento de voz a través de la plataforma virtual de Arduino. Este modo es muy importante ya que permite cargar cualquier código que permita la operación del Arduino y el EasyVR.

UP Mode: Empleado para descargar archivos de audio en la memoria del dispositivo, a través del software Quick Synthesis 5.

EasyVR Commander

Este software es el que permite entrenar, configurar y ordenar los comandos de voz, para realizar esto se debe conectar el acople Shield – Arduino al computador por medio de un cable USB, colocando el MODE JUMPER en la posición PC, en la figura 2.10 se muestra la interfaz del software.

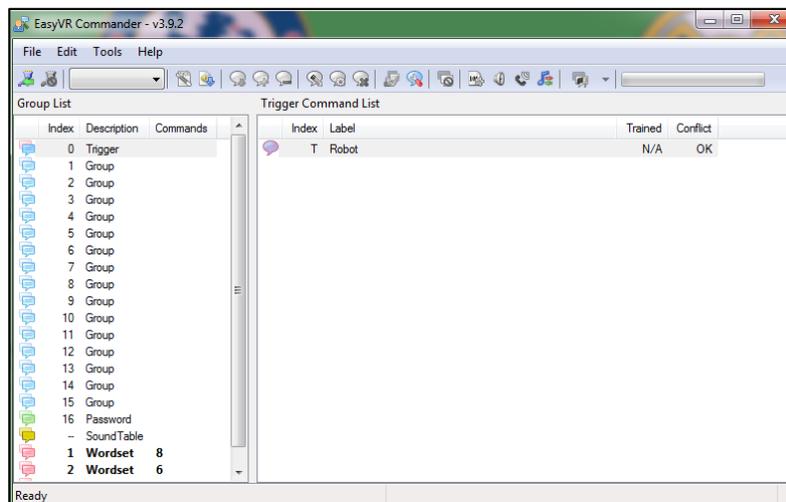


Figura 2.10. Pantalla inicial del programa EasyVR

- **Relé**

El relé permitirá controlar en encendido y apagado del Joystick Controlador ya que consta de dos circuitos diferentes: un circuito electromagnético y un circuito de contactos, al cual se aplica el circuito que quiere controlar, en la figura 2.11 se muestra las partes de un relé.

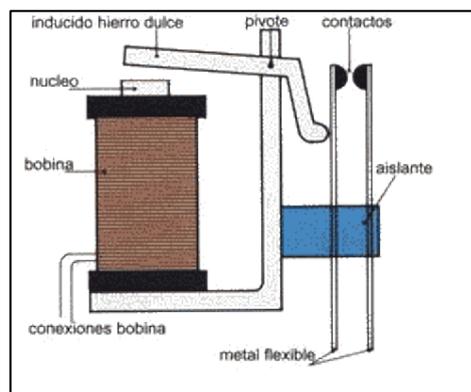


Figura 2.11. Partes de un relé

- **Cables**

Pequeños cables de diferentes colores que utilizares para hacer las distintas conexiones que debemos hacer para que funcione nuestro sistema.

- **Computador Portátil**

Se utilizara un computador portátil marca HP modelo Pavilion g4, que nos permitirá programar, generar el código en Arduino.

- **Batería 9 – 12 V**

Las baterías permiten que la tarjeta Arduino funcione la utilizaremos para pruebas ya que se pretende que la Arduino y el sistema en general funcione con las baterías de la silla.

- **Dos servomotores**

Las características de los servomotores se adjuntan en los apéndices

- **Base para ensamble de componentes**

Es una base de madera en donde estarán sujetos los servomotores junto con los eslabones.

- **Eslabones impresos**

Pequeñas piezas de plástico estructural realizadas en la impresora 3D del CAMPRO.

- **Dos sensores de distancia**

Pequeños sensores ultrasónicos compatibles con Arduino ideales para este tipo de proyectos.

- **Altavoz**

Bocina de 8 Ω , se la utilizará como pito.

- **Módulo Bluetooth**

Dispositivo que acoplado a la tarjeta Arduino, permite que se le pueda agregar controladores bluetooth, en nuestro caso será un celular con sistema Android.

- **Pantalla LCD**

Esta pantalla permitirá visualizar los comandos una vez dada la orden.

2.3. Comandos por Voz

2.3.1. Descripción de cada comando

Adelante.- Permite a la silla avanzar indefinidamente hacia adelante a una velocidad constante en línea recta.

Atrás.- Permite a la silla avanzar indefinidamente hacia atrás a una velocidad constante en línea recta.

Izquierda.- Hace que la silla pare e inmediatamente gira a la izquierda aproximadamente 5° a una velocidad baja.

Derecha.- Hace que la silla pare e inmediatamente gire a la derecha aproximadamente 5° a una velocidad baja.

Encender.- Permite encender el joystick de la silla de ruedas y así comenzar a utilizar los comandos por voz.

Apagar.- Permite apagar el sistema y así poder utilizar el joystick de la silla manualmente.

Alto.- Hace que la silla se detenga de inmediato indistintamente a qué velocidad este.

Baja Velocidad.- Es la velocidad más baja a la que puede andar la silla. Al invocar este comando la silla regresa a esta velocidad inmediatamente.

Media Velocidad.- Es la segunda velocidad de la silla. Al invocar este comando la silla regresa a esta velocidad inmediatamente.

Alta Velocidad.- Es la máxima velocidad a la que puede andar la silla. Al invocar este comando la silla empieza a andar a esta velocidad.

Pito.- Permite al sistema mediante una bocina incorporada emitir un pitido que dura aproximadamente un segundo.

2.3.2. Comandos independientes del locutor

El módulo de reconocimiento de voz incorpora 26 comandos independientes del locutor predefinidos que no requieren una fase de entrenamiento, lo cual le permite a cualquier persona hacer uso de dichos comandos.

2.3.3. Comandos dependientes del locutor

Para estos comandos es necesario llevar a cabo una fase de entrenamiento para usarlos, para ello se debe establecer la comunicación entre el computador y el EasyVR a través del puerto USB del Arduino, se debe colocar el jumper J12 en la posición PC (PC MODE). Luego de esto se procede a entrenar los comandos por medio del software EasyVR Commander. En la figura 2.12 se muestra el significado de los iconos principales del software EasyVR Commander.

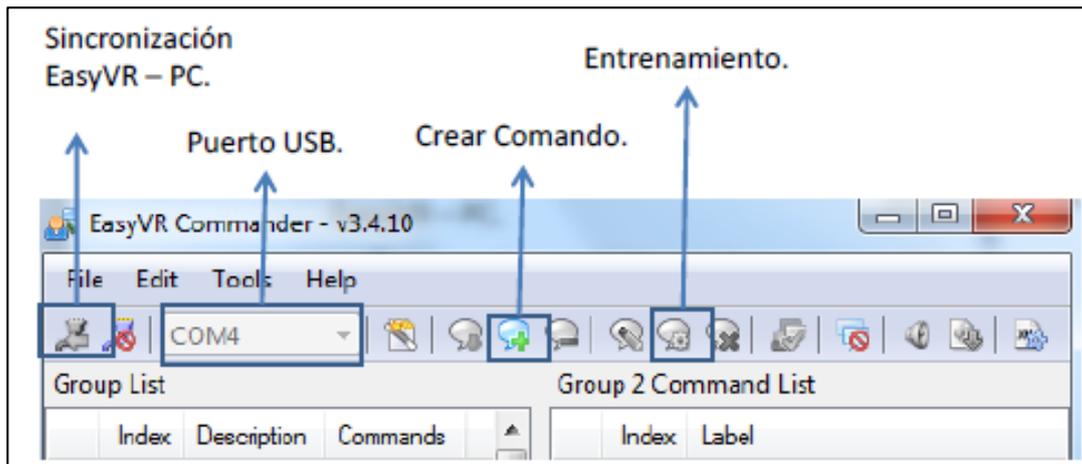


Figura 2.12. Comandos principales del EasyVR Commander

Una vez culminada la creación de los comandos por voz, el software muestra una lista con los nombres asignados por el usuario, en la figura 2.13 se muestra un ejemplo de comandos asignados.

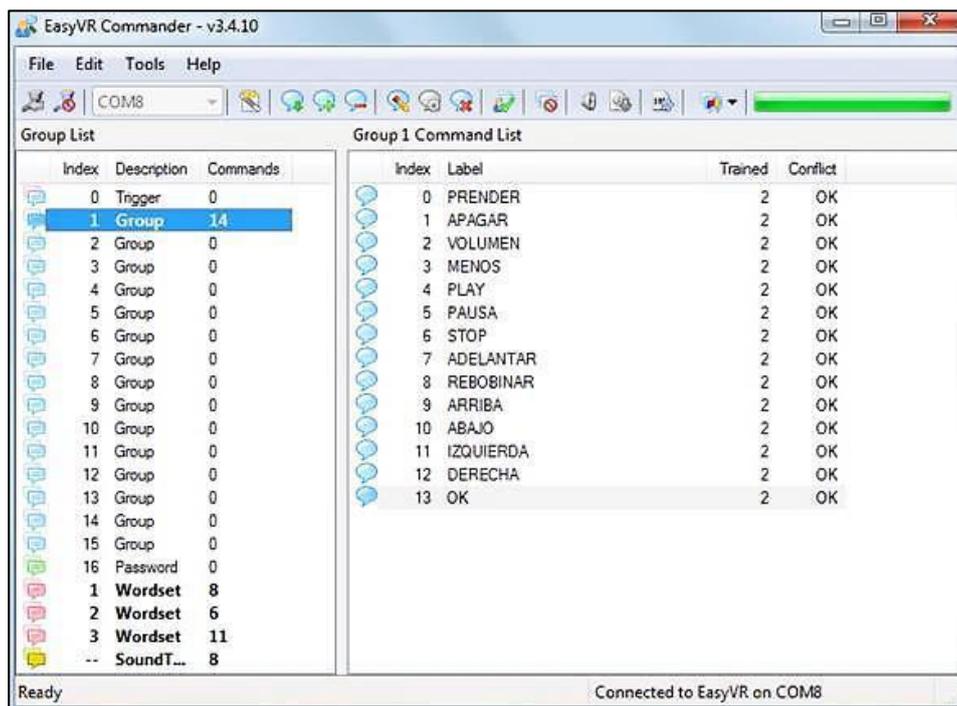


Figura 2.13. Nombres asignados por el usuario

Anteriormente se mencionó que se puede entrenar al módulo de reconocimiento de voz para que sea capaz de reconocer 32 comandos dependientes, los cuales tienen un grupo asignado que sirven de referencia para la realización del código de programación respectivo.

2.4. Control de la parte Mecánica

Esta parte se concentra en resolver el problema de los movimientos básicos de la silla que será la parte mecánica con eslabones o engranes.

2.4.1. Movimientos básicos de la silla

Para darle los movimientos al joystick de la silla se debe utilizar un sistema mecánico, entre las alternativas planteadas está utilizar servomotores junto con eslabones y calcular los ángulos para mover el joystick en la dirección deseada.

Otra alternativa planteada es utilizar engranes para simular el mecanismo de movimiento que se utiliza en una impresora convencional, pero el tiempo de respuesta aumentaría.

La alternativa seleccionada fue de los servomotores.

Ya con el joystick en la silla de ruedas se procede a definir el espacio que se va a utilizar, que debe ser el mínimo posible, en la figura 2.14 se muestran las dimensiones del joystick controlador de la silla de ruedas.

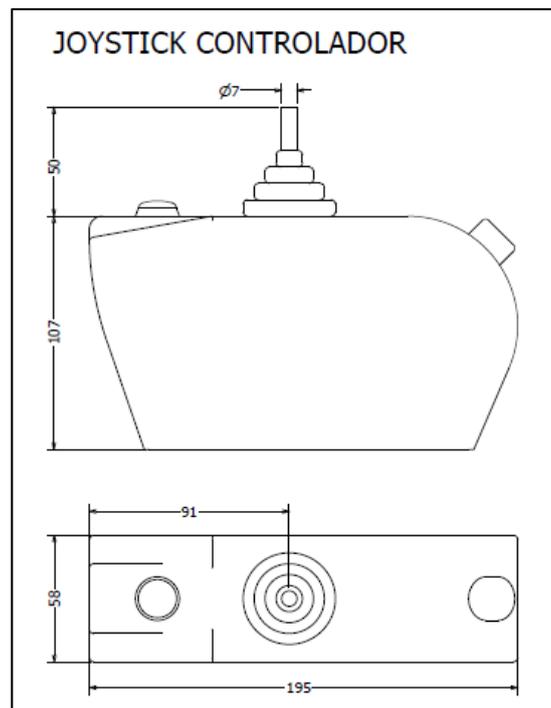


Figura 2.14. Dimensiones principales del Joystick

Con las medidas establecidas del joystick se procede a buscar la posición ideal de los servomotores y dimensionar los eslabones que acoplaran el joystick con los servomotores.

En la figura 2.15 se muestran las dimensiones de los servomotores para colocarlos en el lugar adecuado.

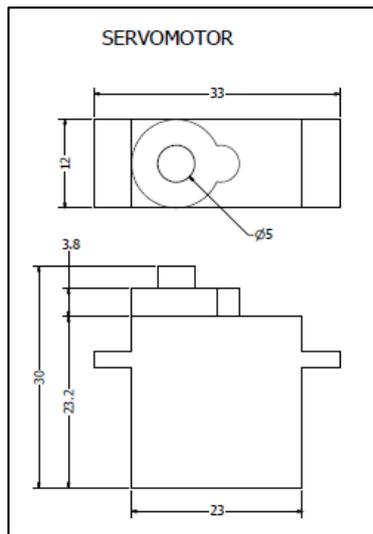


Figura 2.15. Dimensiones del servomotor

Con las dimensiones de los servomotores se procede a ubicarlos de la siguiente manera.

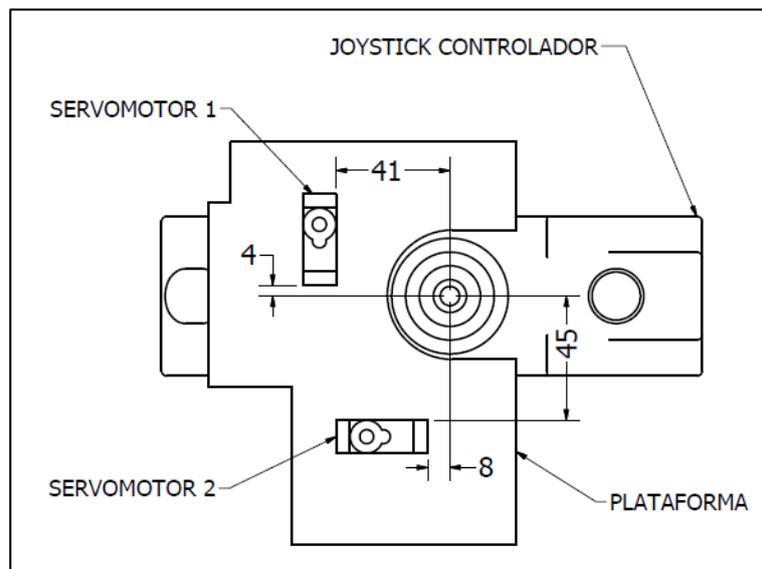


Figura 2.16. Ubicación de los servomotores respecto al joystick

Con las ubicaciones de los servomotores definidas como se muestra en la figura 2.16, se procede a dimensionar los eslabones que acoplaran cada servo con el joystick, se utilizara el mecanismo biela – manivela – corredera.

En la figura 2.17 se muestran las dimensiones de los eslabones son con la ubicación definitiva de cada servomotor.

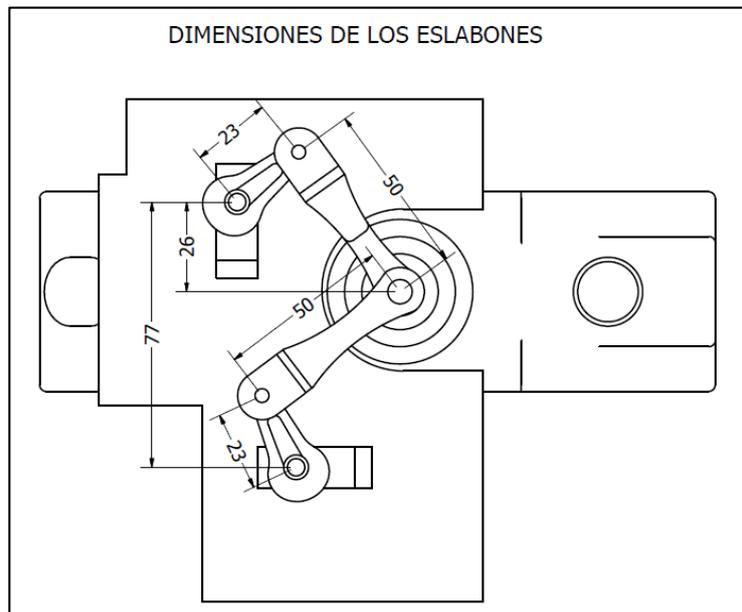


Figura 2.17. Dimensiones de los eslabones

Una vez colocado los eslabones se procede a calcular los ángulos máximos y mínimos que deben girar los servomotores para que el joystick se mueva en las cuatro direcciones. En la tabla 2.3 se muestran los ángulos a los que debe girar cada motor para hacer los movimientos (derecha, izquierda, adelante, atrás).

Tabla 2.2. Ángulos para cada servomotor

Derecha	
Servo 1:	50°
Servo 2:	100°
Izquierda	
Servo 1:	95°
Servo 2:	145°
Atrás	
Servo 1:	40°
Servo 2:	85°
Adelante	
Servo 1:	135°
Servo 2:	105°

Los ángulos fueron determinados de forma experimental y con la ayuda del CAD Inventor. El material utilizado para hacer los eslabones fue un plástico estructural, en la figura 2.18 se muestran algunas características.

Properties	Condition	VisiJet M3 X
Composition		-----
Color		White
Bottle Quantity		2 kg
Density @ 80 °C (liquid)	ASTM D4164	1.04 g/cm ³
Tensile Strength	ASTM D638	49 MPa
Tensile Modulus	ASTM D638	2168 MPa
Elongation at Break	ASTM D638	8.3 %
Flexural Strength	ASTM D638	65 MPa
Heat Distortion Temperature @ 0.45 MPa	ASTM D648	88 °C
Ash Content		N/A
Melting Point		N/A
Softening Point		N/A
USP Class VI Certified*		No
ProJet Compatibility		SD, HD
Description		ABS-like Plastic

Figura 2.18. Características Técnicas del plástico estructural

Los eslabones fueron hechos en una impresora 3D del CAMPRO. Las dimensiones, forma final de cada eslabón se encuentran en la sección 5 apéndices, planos esquemáticos.

Con los ángulos de cada eslabón se procede a crear un código que va a permitir generar la orden para que cada comando se lo pueda controlar con la voz. El código es creado en el software propio de Arduino y los comandos por voz en el software EasyVR Commander.

2.4.2. Control de los giros de los servomotores

La conexión de los servomotores con la tarjeta EasyVR Arduino Shield se muestra en la figura 2.19.

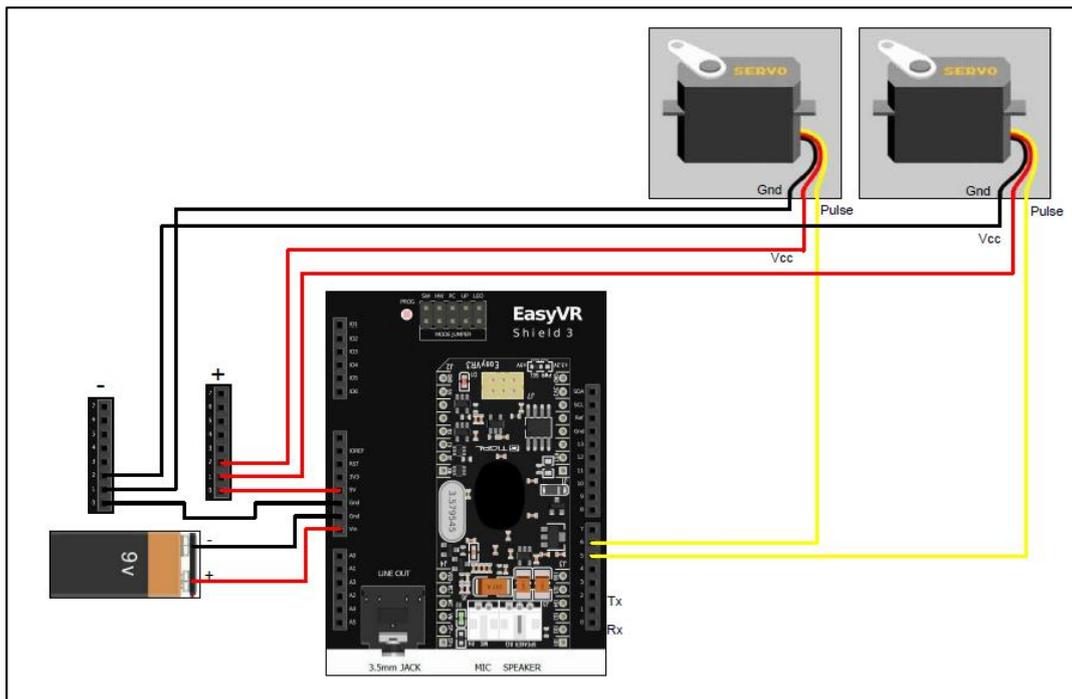


Figura 2.19. Esquema de conexión de los servomotores con la EasyVR

Los servo motores tienen tres entradas: dos analógicas y una digital. Las entradas analógicas son las de color negro y rojo, y se conectan en la parte analógica del Shield y son alimentadas por una batería de 9V. La entrada digital PWM es de color amarillo y se conecta con la parte digital del Shield en los pines 5 y 6 como se muestra en la figura 2.19.

2.5. Control de la parte electrónica

Esta parte se concentra en resolver el problema de encendido y apagado del sistema, colocar los sensores de distancia, incorporar la pantalla LCD, micrófono, altavoz y el módulo bluetooth.

2.5.1. Control de encendido/apagado y velocidad

Para lograr controlar el encendido y apagado del joystick con la tarjeta Arduino se utiliza un relé, en la figura 2.20 se muestra en esquema de conexiones del relé con el Shield.

También se utiliza un comando de voz "ENCENDIDO/APAGADO" el cual envía al pin "3" un pulso de 5V durante un segundo, el cual activa un relé y este a su vez cierra el circuito de encendido y apagado del Joystick Controlador.

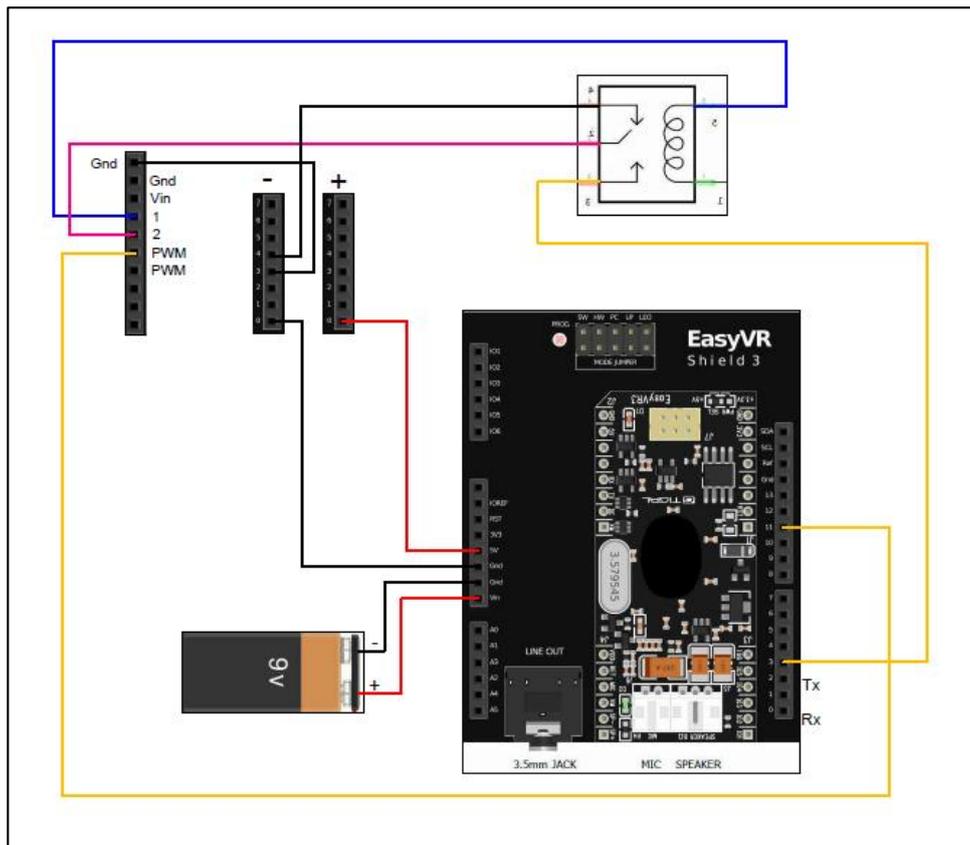


Figura 2.20. Esquema de conexión del Relé

2.5.2. Conexión de sensores de distancia

Los sensores de distancia son del tipo ultrasonido, emiten ondas y mediante su reflexión en obstáculos censan la distancia a que se encuentran. La ubicación de los sensores será uno en la parte delantera y otro en la parte posterior, la seguridad es muy esencial para la realización del proyecto.

Esta función es independiente del módulo de voz ya que este actúa cuando encuentra un obstáculo para asegurar la integridad de la persona en la silla. Este sensor se activa mediante un código o librería especial para Arduino, mediante los pines Trig 7 y 10 mandamos un pulso de 0v durante dos microsegundos y uno de 5v durante diez microsegundos, esto genera una onda ultrasónica el cual al momento de encontrar un obstáculo es reflejada y es censado por el "Echo" del sensor y manda una señal digital que ingresa al pin 8 y 9 del Arduino, como se muestra en la figura 2.21. Mediante un pequeño cálculo matemático se calcula la distancia a la que se encuentra el obstáculo del sensor.

$$d_{obstaculo} = T \cdot V_s / 2$$

Ec. 2.1

V_s : Velocidad del sonido

T: Tiempo que le tomo a la onda en regresar al sensor.

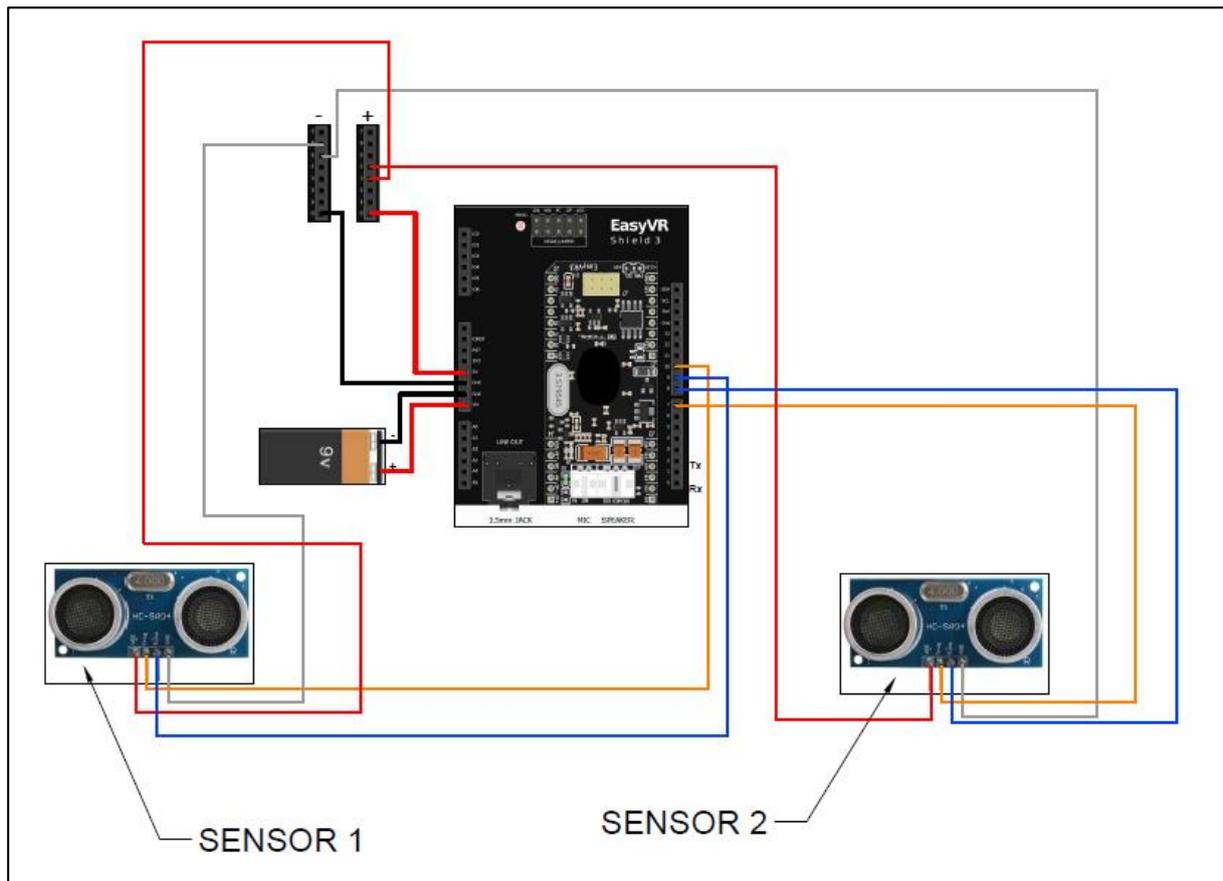


Figura 2.21. Esquema de conexión de los sensores de distancia

2.5.3. Pantalla LCD

La pantalla LCD es una herramienta útil al momento de calibrar la voz y los comandos, para poder incorporarla en el sistema se debe adquirir un Arduino nano y otra batería.

En la figura 2.22 se muestra el esquema de conexión con la tarjeta Arduino nano el cual se conecta en serie con el Arduino uno. Por medio de este Arduino nano se puede saber que comandos de voz están siendo activados para luego mostrarlos en una pantalla LCD al usuario.

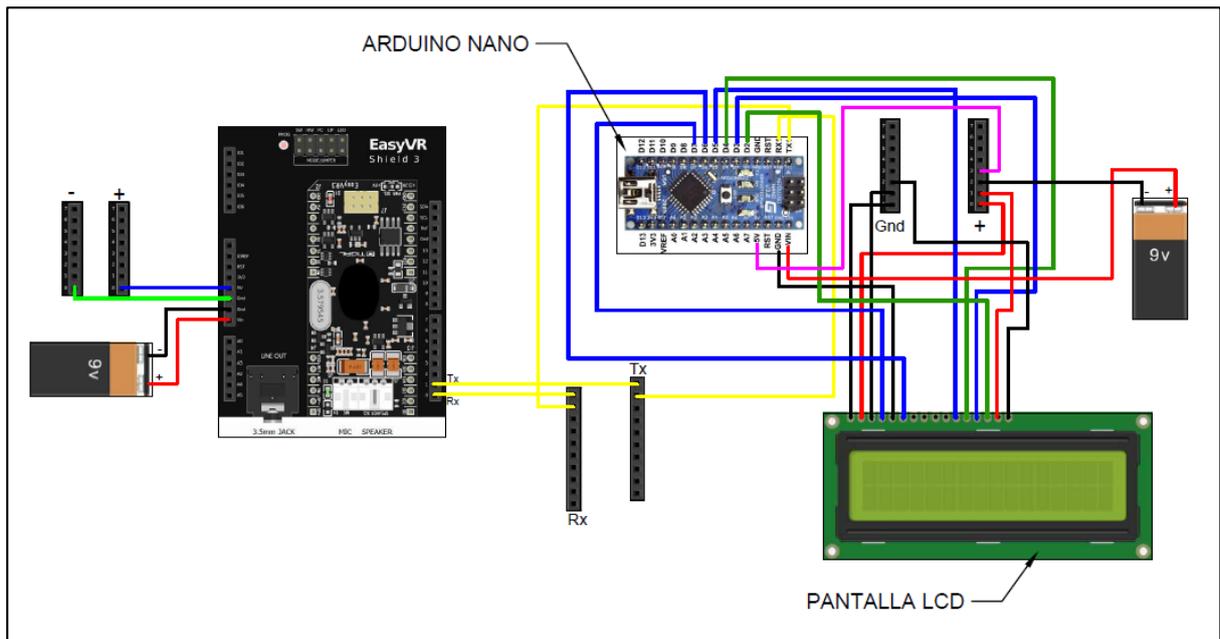


Figura 2.22. Esquema de conexión de Arduino nano y pantalla LCD

2.5.4. Incorporación del micrófono y altavoz

Para agregar el micrófono se tuvo que hacer una conversión de un Jack estéreo 3.5 mm a un Jack mono (hembra) 3.5 mm este último Jack va conectado a los pines del micrófono de la EasyVR, como se muestra en la figura 2.23.

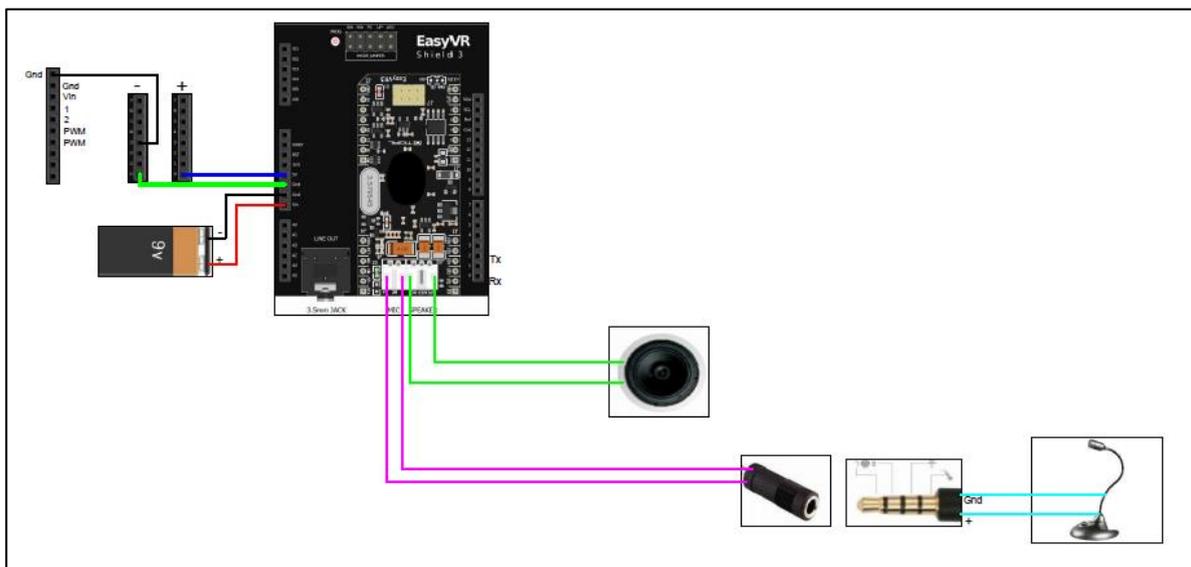


Figura 2.23. Esquema de conexión del micrófono y altavoz

2.5.5. Adaptación del módulo bluetooth

El módulo Bluetooth fue una adaptación para comparar los tiempos de respuesta entre el micrófono y mediante un celular con tecnología Android y con conexión Bluetooth.

Para llevar a cabo esto se requiere crear una aplicación que permita enlazar el celular con el módulo bluetooth. Este módulo también se conecta en comunicación serial o serie con el Arduino Uno mediante los pines RX (0) y TX (1).

El esquema se muestra en la figura 2.24.

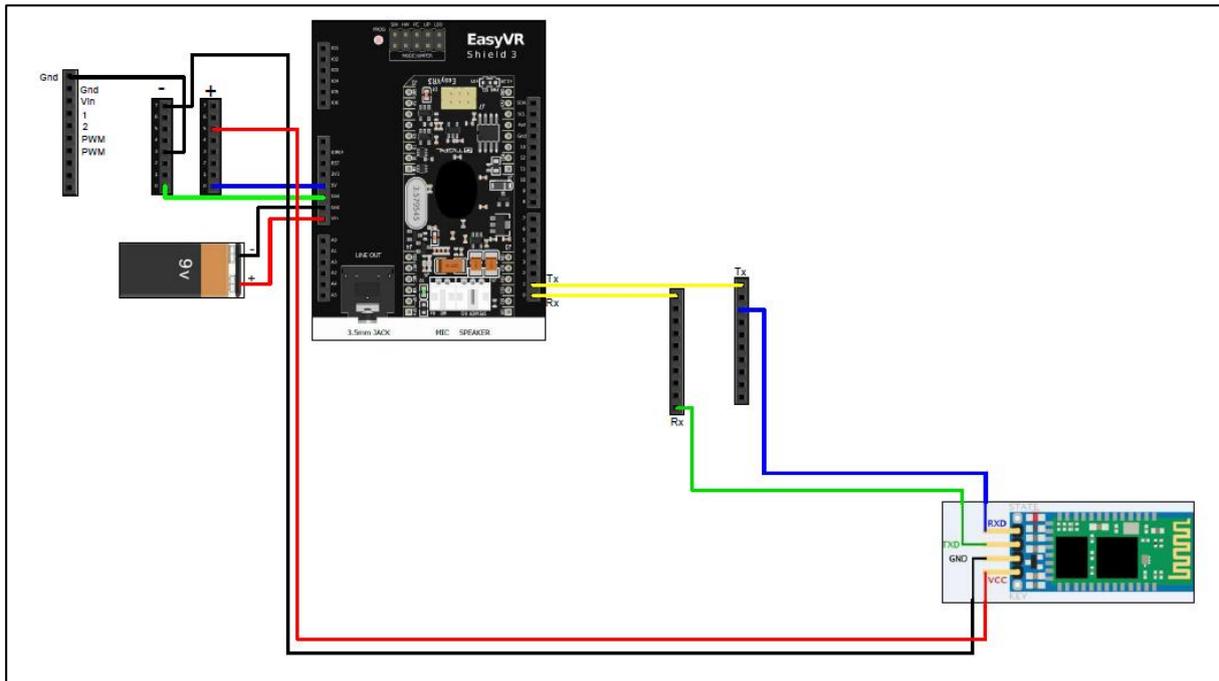
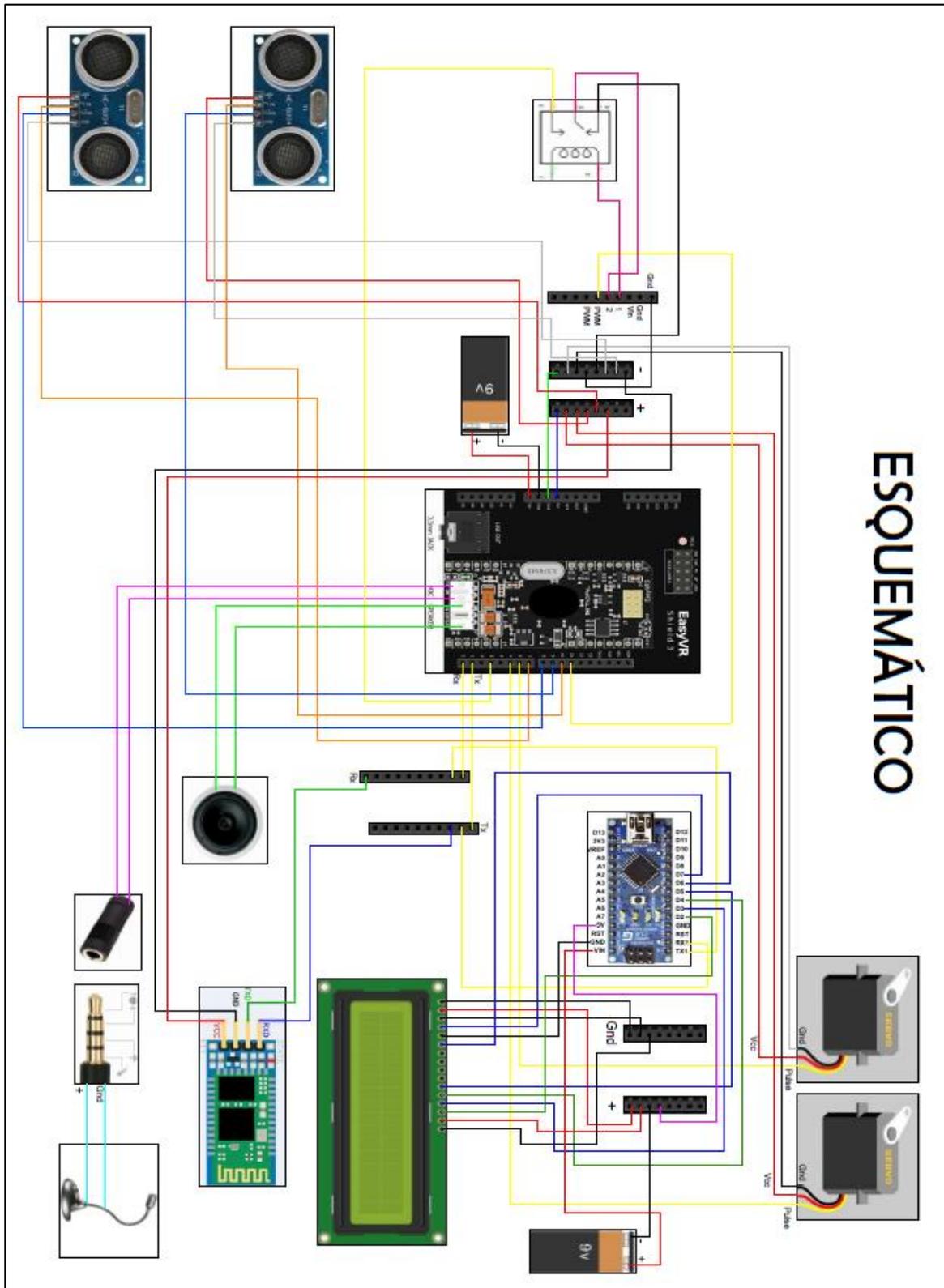


Figura 2.24. Esquema de conexión del módulo bluetooth

Una vez hecho un conjunto todas las incorporaciones el esquema final es el que se muestra en la figura 2.25.



ESQUEMÁTICO

Figura 2.25. Esquema Final de las conexiones del sistema por voz

2.6. Conexión del Sistema con la tarjeta del Joystick

El sistema se describe en la figura 2.25 donde se muestran los pines de salida del joystick

2.7. Diagrama de Flujo

El diagrama que se describe en la figura 2.26, donde se muestra gráficamente el proceso de funcionamiento de todo el sistema de la silla de rueda eléctrica. Teniendo como variable de entrada la señal adquirida por el micrófono que luego es procesada por la Easy VR junto con el Arduino para efectuar las ordenes de control al joystick y este a su vez controle los motores de la silla por medio de un puente H.

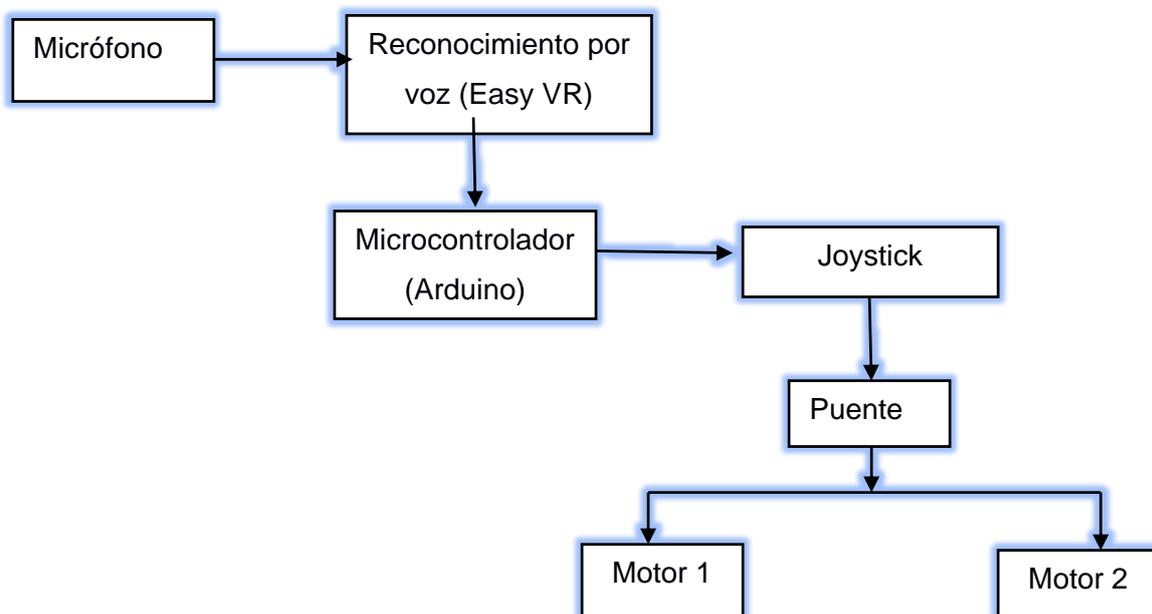


Figura 2.26. Diagrama esquemático del funcionamiento del sistema acoplado a la silla de ruedas

En el diagrama presentado en la figura 2.27, se muestra el orden secuencial de cómo se efectúa internamente el algoritmo que se elaboró para efectuar las acciones de control. Donde se inicializan las variables de entrada luego de eso se presenta un ciclo repetitivo en donde si el sistema reconoce el comando de voz guardado en la tarjeta efectúa la acción correspondiente al comando de voz. En caso de que no reconozca el ciclo se repite desde el inicio.

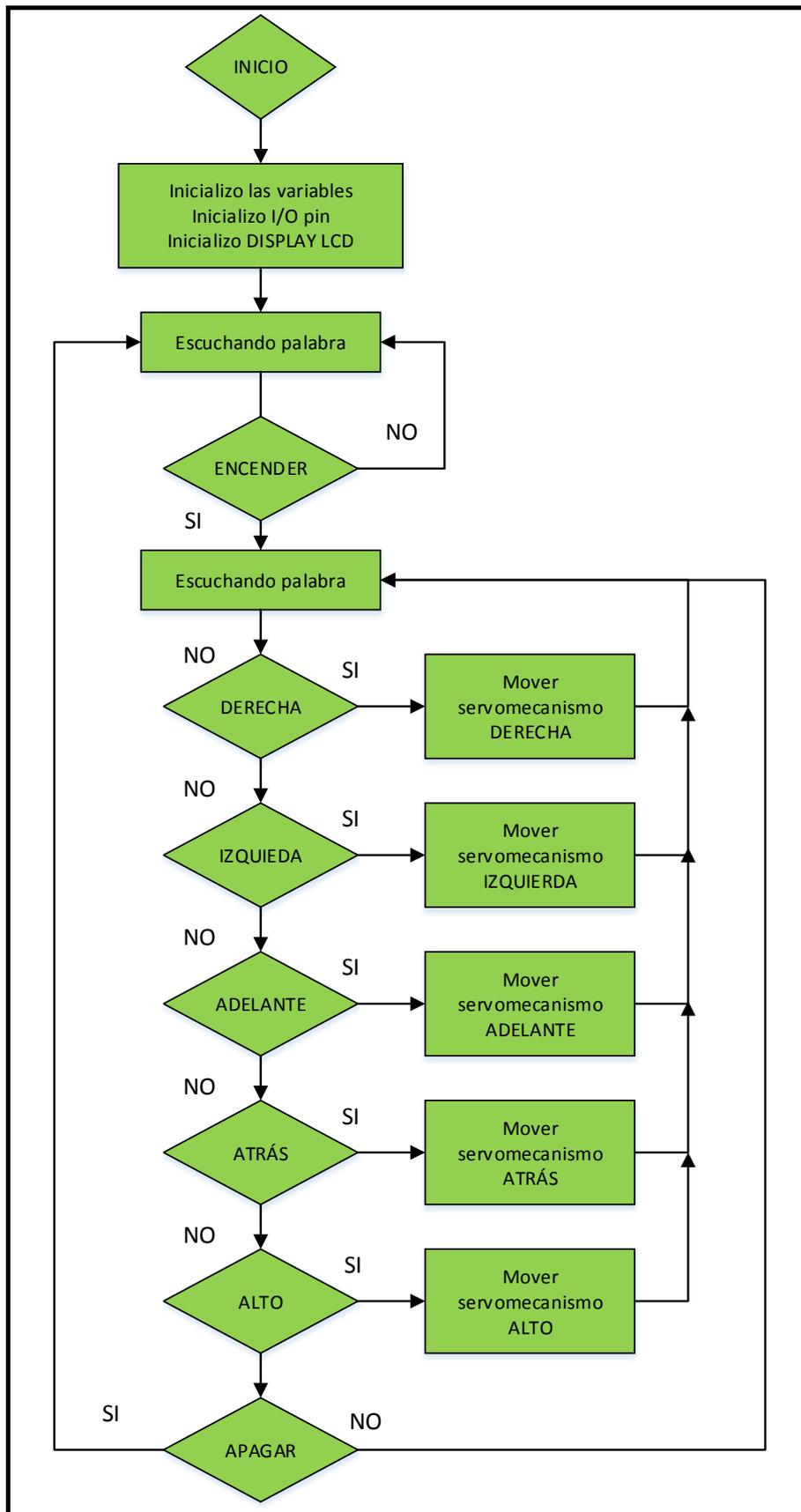


Figura 2.27. Diagrama del proceso del sistema con los comandos

2.8. Velocidades de la Silla

Para determinar las velocidades de la silla se utiliza un tacómetro que nos permite determinar la velocidad angular de cada rueda, y con el diámetro de la rueda se podrá determinar la velocidad lineal de la silla, utilizando la ecuación 2.2.

$$V_{silla} = \omega \cdot \left(\frac{\varphi}{2}\right)$$

Ec. 2.2

V_{silla} : Velocidad en cm/s

ω : Velocidad angular en rpm

φ : Diámetro de la rueda cm

En la tabla 2.4 se muestran los datos obtenidos de la velocidad de las ruedas en RPM de la silla de rueda.

Tabla 2.3. Relación del Voltaje con los RPM

Voltaje(V)	Velocidad de las llantas RPM			
	izquierda	derecha	adelante	atrás
0	16	15	30,5	24
2,5	27	22	73	58
4,7	48	43	110	89,5

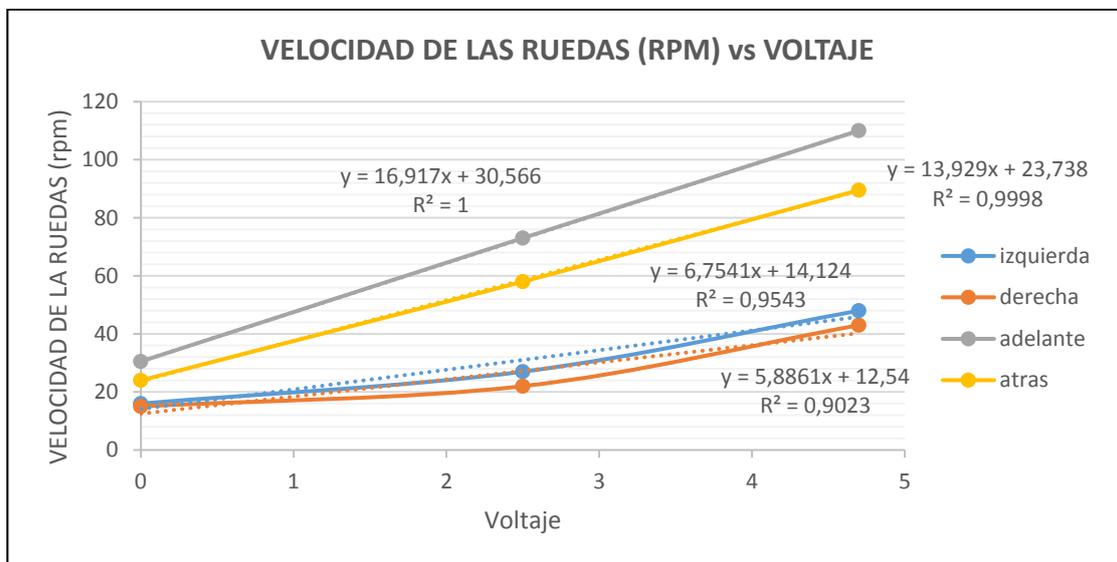


Figura 2.28. Relación matemática entre la Velocidad (RPM) y Voltaje

En la figura 2.28 se muestra la velocidad de las ruedas vs el voltaje que ingresa al controlador del joystick, se observa que para los diferentes movimientos de la silla como "ADELANTE", "ATRÁS", "DERECHA", "IZQUIERDA", se tiene diferentes valores de velocidad

angular de las ruedas tomando como observación que cuando la silla se mueve a la izquierda solo gira la rueda izquierda y cuando gira a la derecha solo gira la llanta derecha y la rueda izquierda permanece estática. Se tiene en cuenta que los valores tienen una tendencia lineal.

Para calcular la velocidad de la silla en los movimientos hacia ADELANTE y ATRÁS se usa la ecuación 2.3:

$$r_{rueda} = 128.5 \text{ mm}$$

$$V_{silla} = \omega_{rueda} \times \left(\frac{2\pi}{60}\right) \times r_{rueda} \tag{Ec. 2.3}$$

$$V_{silla}[\text{cm/seg}] = \omega_{rueda} \times \left(\frac{2\pi}{60}\right) \times (12.85)$$

Con esta ecuación se calcula la velocidad de la silla, y los valores obtenidos se muestran en la tabla 2.5:

Tabla 2.4. Relación Voltaje vs. Velocidad Lineal de la Silla

Voltaje(V)	Velocidad de la silla (cm/seg)	
	adelante	atrás
0	40,86	32,15
2,5	97,80	77,70
4,7	147,37	119,91

En la figura 2.29 se muestra que los valores tienen una tendencia lineal por la cual se expresa una ecuación de la velocidad de la silla vs el voltaje.

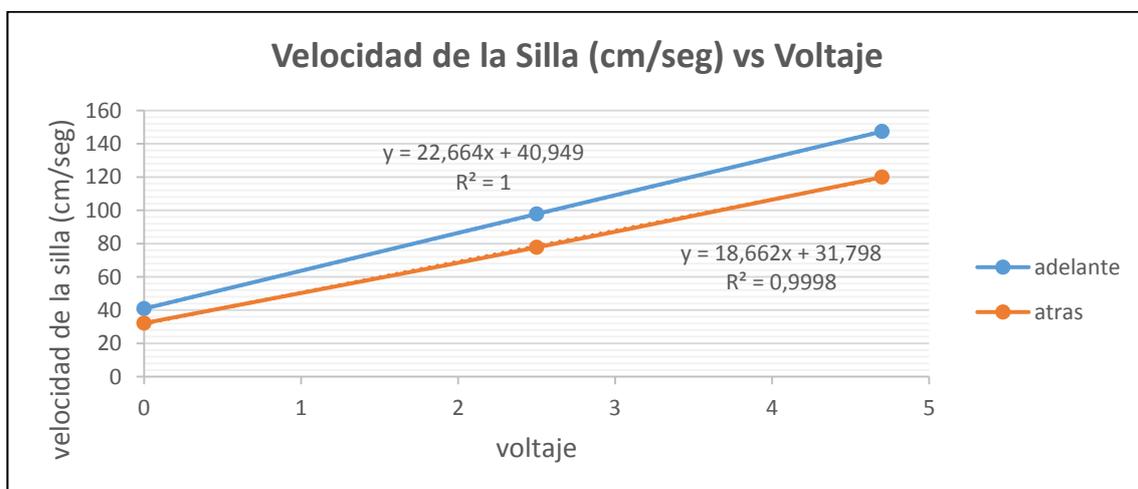


Figura 2.29. Relación Lineal entre Velocidad de la silla y voltaje

Movimiento “Adelante”:

La ecuación 2.4 se obtuvo como resultado de la figura 2.29 y se muestra a continuación.

$$V_{silla} \left[\frac{cm}{seg} \right] = 22,664V + 40.9 ; \text{ Donde } V \text{ es el voltaje de } 0V \text{ a } 5V. \quad \text{Ec. 2.4}$$

Movimiento “Atrás”:

$$V_{silla} \left[\frac{cm}{seg} \right] = 18.662V + 31.8 ; \text{ Donde } V \text{ es el voltaje de } 0V \text{ a } 5V.$$

Para calcular la velocidad de giro de la silla en los movimientos: DERECHA e IZQUIERDA se utiliza la ecuación 2.5:

$$\omega \left[\frac{grados}{seg} \right] = \frac{V_{silla}}{r_{silla}} \times \left(\frac{180}{\pi} \right) \quad \text{Ec. 2.5}$$

$$r_{rueda} = 128.5 \text{ mm}$$

$r_{silla} = 500 \text{ mm}$; Este radio es la longitud medida de los extremos de cada rueda como se muestra en la figura 2.30.



Figura 2.30. Separación de las ruedas de la silla

$$V_{silla} = \omega_{rueda} \times \left(\frac{2\pi}{60} \right) \times r_{rueda}$$

$$V_{silla} \left[\frac{cm}{seg} \right] = \omega_{rueda} \times \left(\frac{2\pi}{60} \right) \times (12.85)$$

Con esta ecuación se calcula la velocidad de giro de la silla y se muestra los valores obtenidos en la tabla 2.6.

Tabla 2.5. Relación entre el voltaje del potenciómetro y los giros de la silla

Voltaje(V)	giro de la silla (grados/seg)	
	izquierda	derecha
0	24,58	23,04
2,5	41,47	38,40
4,7	73,73	66,05

En la figura 2.31 se muestra que los valores tienen una tendencia lineal por la cual se puede expresar una ecuación de la velocidad de giro de la silla vs el voltaje.

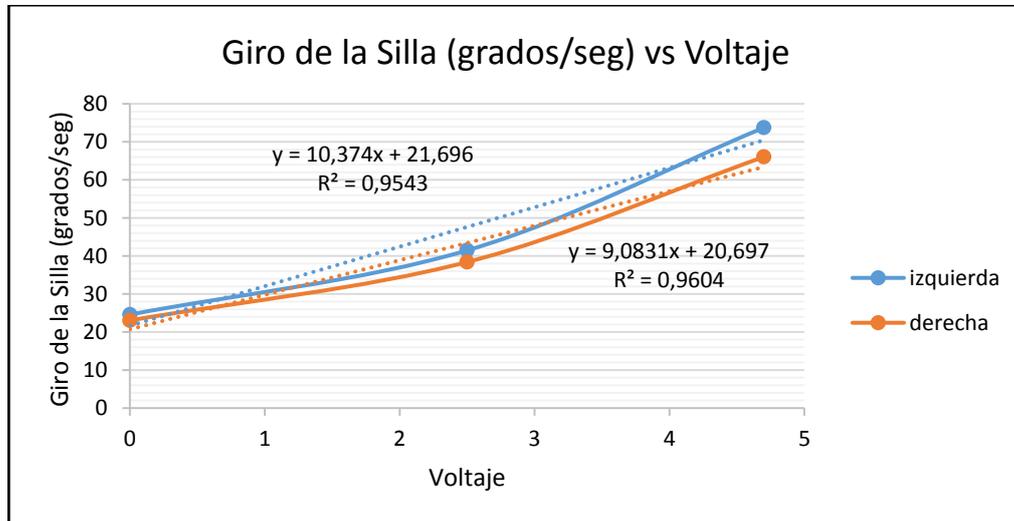


Figura 2.31. Relación cuadrática entre el giro de la silla y el voltaje

La ecuación 2.6 se obtuvo como resultado de la figura 2.31 y se muestra a continuación.

Movimiento “Izquierda”:

$$\omega_{silla} \left[\frac{\text{grados}}{\text{seg}} \right] = 10.4V + 21.7 ; \text{Donde } V \text{ es el voltaje de } 0V \text{ a } 5V. \quad \text{Ec. 2.6}$$

Movimiento “Derecha”:

$$\omega_{silla} \left[\frac{\text{grados}}{\text{seg}} \right] = 9.08V + 20.7 ; \text{Donde } V \text{ es el voltaje de } 0V \text{ a } 5V.$$

Con las ecuaciones que se calcularon, se puede usarla para controlar la velocidad de la silla de ruedas, ya que con el microcontrolador ARDUINO se puede mandar señales de voltaje PWM en uno de sus pines digitales hacia el controlador del joystick. Para mandar esta señal de voltaje primero tiene que ir por un filtro pasabanda, este filtro lo que hace es convertir la señal PWM (DIGITAL) a una señal analógica.

Para eso se usa la calculadora de la universidad de Tokio OKAWA, que se muestra en la figura 2.31, el cual se ingresan los parámetros de entrada como:

f_{pwm} (Frecuencia de la señal PWM generada por ARDUINO)= 500Hz

V_L (Rango de voltaje de los pines de salidas del ARDUINO)=5V – 0V

R: Resistencia del circuito

C: Capacitor del circuito

Se puede observar que cuando la Resistencia es de $10\text{K}\Omega$ y el capacitor de $10\mu\text{F}$ el valor de pico o gradiente de voltaje (ΔV) es de 0.025 , por lo tanto se escoge estos valores de resistencia y voltaje para el circuito pasa bajo.

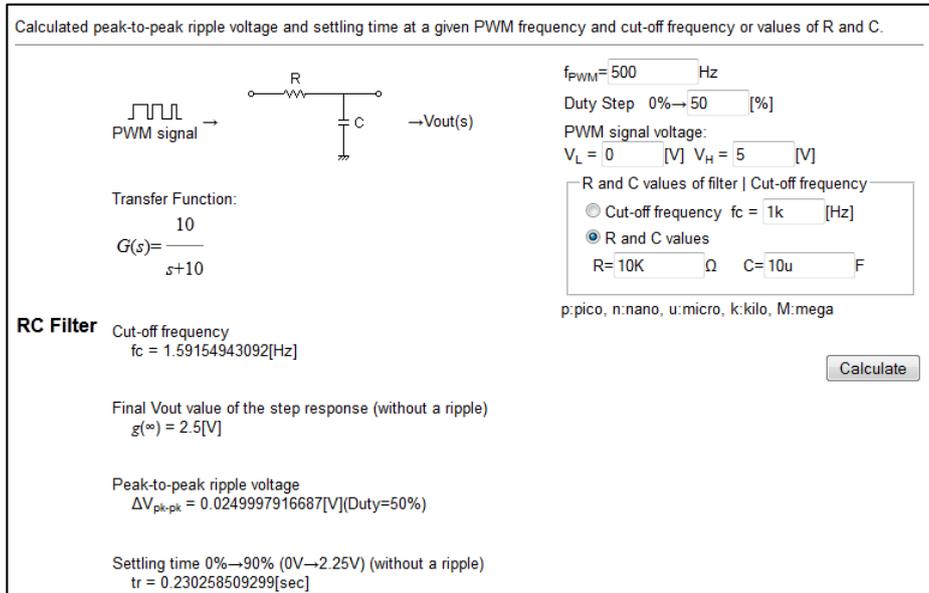


Figura 2.32. Calculadora de la Universidad de Tokio Okawa (Inglés)

En la figura 2.33 se muestra el tiempo de respuesta que le toma a este sistema a obtener el valor deseado es bueno con un tiempo de estabilización de 0.3seg y con un pico de gradiente de voltaje de 0.024V

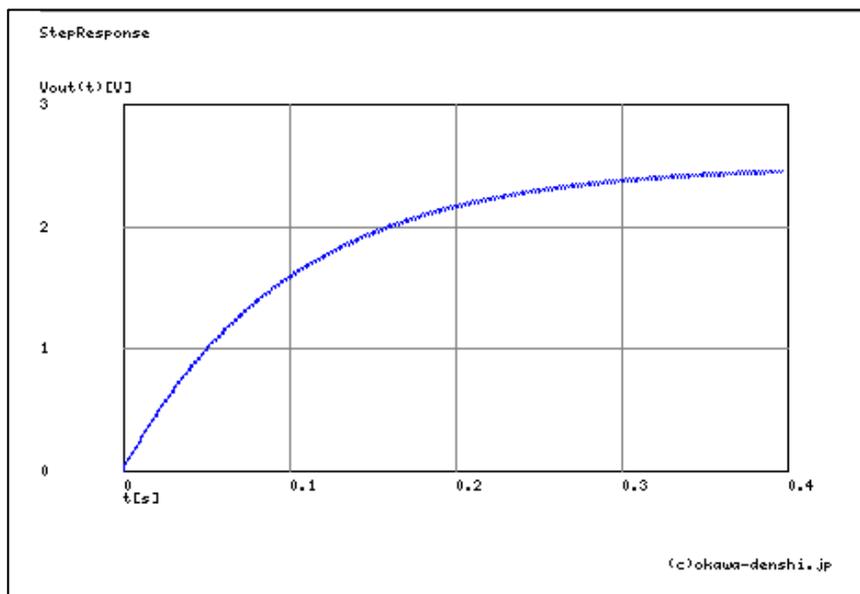


Figura 2.33. Tiempo de respuesta del sistema

2.9. Pruebas de efectividad del sistema

Las pruebas que se realizan tienen el objetivo principal de verificar que el módulo de reconocimiento de voz tuviera la misma efectividad para cualquier comando de voz, para lo cual se realizan unas grabaciones previas donde se registra al locutor pronunciando los comandos de voz para tal fin.

Las pruebas se las realiza en el laboratorio de mecatrónica de la ESPOL, los comandos para controlar la silla fueron:

Adelante, Atrás, Izquierda, Derecha, Encender, Apagar, Baja, Media, Alta, Pita, en las tablas 2.7 y 2.8 se muestran los comandos y la palabra utilizada para dicho comando.

Tabla 2.6. Comandos y palabras designadas para el joystick

		PALABRA UTILIZADA
COMANDOS	ADELANTE	"ADELANTE"
	ATRÁS	"ATRÁS"
	IZQUIERDA	"IZQUIERDA"
	DERECHA	"DERECHA"
	ALTO	"ALTO"

Tabla 2.7. Comandos y palabras designadas para el joystick controlador

		PALABRA UTILIZADA
COMANDOS	ENCENDER	"ENCIENDE"
	APAGAR	"APAGAR"
	VELOCIDAD BAJA	"BAJA"
	VELOCIDAD MEDIA	"MEDIA"
	VELOCIDAD ALTA	"ALTA"
	PITA	"PITA"

Las palabras utilizadas son ejemplos, al final el usuario o locutor escoge la palabra adecuada para cada comando.

La eficiencia se mide por las veces que se tiene que repetir la palabra para que funcione el comando u orden.

2.10. Comparación entre Arduino y Bluetooth

Para realizar la comparación entre Arduino y Bluetooth se consideran los siguientes aspectos:

- Tiempo de respuesta

- Disponibilidad de un teléfono celular
- Disponibilidad de un plan de datos
- Micrófono especial
- Grabación previa de comandos

En la tabla 2.9 se muestra datos importantes y comparativos para tener en cuenta al momento de elegir entre uno de ellos.

Tabla 2.8. Comparación entre Arduino y Bluetooth

	Arduino	Bluetooth
Tiempo de respuesta	~ 3 seg	~ 1.5 seg
Disponibilidad de un teléfono celular	No	Si
Disponibilidad de un plan de datos	No	Si
Micrófono con filtro de ruido exterior	Si	No
Grabación previa de comandos	Si	Si
Eficiencia	94	97

2.11. Costos

Los costos de cada pieza se muestran a continuación en la tabla 2.10.

Tabla 2.9. Costos de los materiales y equipos utilizados

	CANTIDAD	COSTO	VALOR TOTAL
ARDUINO UNO	1	28	28
EASYVR	1	65	65
LCD	1	8	8
BLUETOOTH	1	12	12
SERVOMOTOR	2	8	16
RELE	2	1	2
SENSORES	2	10	20
CAJA METALICA	1	10	10
FORRO	1	10	10
ARDUINO NANO	1	17	17
IMPRESIÓN 3D	1	30	30
PERNOS Y TUERCAS	30	0,15	4,5
PINES	1	3	3
MICROFONO	1	50	50
CABLES, ETC	1	4	4
			279,5

El costo de los elementos es de \$280 dólares.

CAPÍTULO 3

3. RESULTADOS

- Como resultado de nuestro diseño tal como se muestra en la figura 3.1. Se obtuvo un sistema que permite a la silla de rueda controlarla por voz, por medio de comandos de voz que son programables para cada tipo de voz. El diseño final de la silla con el sistema de voz acoplado al joystick se muestra en la figura 3.2.

Los comandos por voz designados fueron:

1. Adelante.
2. Atrás.
3. Izquierda.
4. Derecha.
5. Encender.
6. Apagar.
7. Velocidad baja.
8. Velocidad Media.
9. Velocidad Alta.
10. Alto.
11. Pitar.



Figura 3.1. Sistema de reconocimiento de voz que se acopla a joystick controlador.



Figura 3.2. Silla de ruedas Jazzy Jet 3 con el sistema por voz

- El tiempo de respuesta que toma en realizar cada uno de los comando es de aproximadamente un segundo y medio.
- El tiempo de respuesta utilizando el módulo bluetooth fue de tres segundos para cada comando.
- El tiempo de respuesta de los sensores de distancia fue de medio segundo para una distancia de 100 cm.
- Las Velocidades de la Silla para cada movimiento se muestran en la tabla 3.1.

Tabla 3.1. Velocidades de la silla para cada comando

Velocidad (cm/s)	Adelante	Atrás
Alta	148	120
Media	87	69
Baja	53.8	43

- La eficiencia de los comandos por voz usando el módulo de voz EasyVR y con palabras largas se obtuvo un promedio de 76.4 % de eficacia; tal como se muestra en la tabla 3.2.

Tabla 3.2. Eficiencia para cada palabra utilizada

Comandos	Palabra utilizada	Intentos Reconocidos	# Intentos
Adelante	"Adelante"	4	5
Atrás	"Atrás"	4	5
Izquierda	"Izquierda"	3	5
Derecha	"Derecha"	3	5
Encender	"Enciende"	4	5
Apagar	"Apagar"	4	5
Velocidad Media	"Media"	4	5
Velocidad Baja	"Baja"	4	5
Velocidad Alta	"Alta"	4	5
Alto	"Alto"	4	5
Pitar	"Pita"	4	5
		76.4	Eficiencia (%)

Tabla 3.3. Eficiencia para cada palabra corta utilizada

Comandos	Palabra utilizada	Intentos Reconocidos	# Intentos
Adelante	"Adel"	5	5
Atrás	"Atrás"	4	5
Izquierda	"Iz"	5	5
Derecha	"Dere"	5	5
Encender	"ON"	4	5
Apagar	"OFF"	4	5
Velocidad Media	"Media"	4	5
Velocidad Baja	"Baja"	4	5
Velocidad Alta	"Alta"	4	5
Alto	"STOP"	4	5
Pitar	"Pi"	5	5
		87.3	Eficiencia (%)

En la tabla 3.3 se muestra que la eficiencia promedio aumento a 87.3% utilizando palabras cortas.

Utilizando el módulo bluetooth y el reconocimiento de voz de Google se obtuvo una eficiencia del 92.7% usando palabras largas tal como se muestra en la tabla 3.4.

Tabla 3.4. Eficiencia utilizando reconocimiento de voz de Google

Comandos	Palabra utilizada	Intentos Reconocidos	# Intentos
Adelante	"Adelante"	5	5
Atrás	"Atrás"	5	5
Izquierda	"Izquierda"	5	5
Derecha	"Derecha"	4	5
Encender	"Enciende"	5	5
Apagar	"Apagar"	5	5
Velocidad Media	"Media"	4	5
Velocidad Baja	"Baja"	5	5
Velocidad Alta	"Alta"	4	5
Alto	"Alto"	5	5
Pitar	"Pita"	4	5
		92.7	Eficiencia (%)

En la tabla 3.5 se muestra la eficiencia de los comando de voz con palabras cortas, obteniendo una eficiencia del 92.7%

Tabla 3.5. Eficiencia utilizando reconocimiento de voz de Google en palabras cortas

Comandos	Palabra utilizada	Intentos Reconocidos	# Intentos
Adelante	"Adel"	5	5
Atrás	"Atrás"	5	5
Izquierda	"Iz"	4	5
Derecha	"Dere"	5	5
Encender	"ON"	5	5
Apagar	"OFF"	4	5
Velocidad Media	"Media"	5	5
Velocidad Baja	"Baja"	5	5
Velocidad Alta	"Alta"	4	5
Alto	"STOP"	5	5
Pitar	"Pi"	4	5
		92.7	Eficiencia (%)

CAPÍTULO 4

4. DISCUSIÓN Y CONCLUSIONES

- El módulo de reconocimiento EasyVR, al acoplarse con la tarjeta Arduino UNO, resultaron ser buenas opciones para el desarrollo de prototipos que requieren el reconocimiento por voz.
- El tiempo de respuesta se ve afectado por la longitud de cada comando es decir si se utiliza la palabra “izquierda” para el comando izquierda, al decir la palabra toma más tiempo que decir una palabra más corta como por ejemplo “iz”, eso ayuda a mejorar el tiempo de respuesta en cada comando.
- El tiempo de respuesta de los sensores de distancia se ve afectado por la distancia que se necesite que la silla se detenga, ya que al ser sensor de distancia ultrasónico emite ondas y se rige bajo la siguiente ecuación:

$$d = txV_{sonido}$$

Ec. 4.1

t: tiempo entre T_{rig} y Echo

La velocidad del sonido en función de la temperatura está dada por la siguiente ecuación [9].

$$V_{sonido} = 331.4 + 0.61T,$$

Ec. 4.2

Donde T es la temperatura ambiente en °C

En la Ciudad de Guayaquil la temperatura ambiente promedio es de 30 °C [10].

$$\text{Por tanto: } V_{sonido} = 331.4 + (0.61)(30) = 349.7 \text{ m/s}$$

En la tabla 4.1. Se muestra los resultados de los tiempos de retorno de los pulsos a distintas distancias.

Tabla 4.1. Tiempo de retorno en los sensores de distancia

tiempo (ms)	distancia (cm)	velocidad sonido (m/s)
0,00	0	349,7
0,57	10	
1,14	20	
1,72	30	
2,29	40	
2,86	50	
3,43	60	
4,00	70	
4,58	80	
5,15	90	
5,72	100	
6,86	120	
7,43	130	
8,01	140	
8,58	150	

El tiempo de retorno aumenta a medida que aumenta la distancia, es de mucha consideración escoger un valor adecuado de distancia ya que el tiempo para que pare la silla a dicha distancia depende del tiempo “t” y de la velocidad de la silla.

Se notó que para la distancia de paro de la silla (120 cm) el tiempo de retorno es de 7 ms aproximadamente.

- Un factor importante por el cual el tiempo de respuesta utilizando el módulo bluetooth fue de tres segundos es la conexión a internet, ya que al momento de hacer las pruebas se utilizó conexión inalámbrica, es preferible utilizar conexión de datos ya sea 3G o si es posible 4G, pero esto incluiría costos adicionales dependiendo de los megas y la operadora telefónica.
- La eficiencia aumento a 92.7% debido a que el reconocimiento de voz de Google es uno de los mejores del mundo, y está en constante mejoras con sus productos “Google Voice Search” [12].
- Al momento de decir un comando, el reconocimiento de voz de Google busca encontrar una relación directa entre una onda de sonido y un texto, y no entre dos palabras o frases de distinto idioma, en el apéndice se encontrará información completa del reconocimiento de voz de Google [12].

- Mover el Joystick con los servomotores resultó ser una solución válida y muy mecánica pero no necesariamente es la mejor ya que se requiere tener precisión al momento de acoplar el sistema al Joystick Controlador, ya que un mal acople hará que el sistema no funcione, una mejor solución podría haber sido hacerlo netamente electrónico mediante un PWM.

4.1. Conclusiones

- Al utilizar palabras más cortas es decir con menos sílabas el reconocimiento es mejor con esto aumento la eficiencia del sistema.
- Comparando el sistema mediante el micrófono acoplado a la EasyVR con el del bluetooth acoplado a la EasyVR, mejor eficiencia en cuanto a reconocimiento de voz tuvo el reconocimiento con bluetooth.
- La eficiencia en cuanto al tiempo de respuesta entre el micrófono acoplado a la EasyVR con el del bluetooth acoplado a la EasyVR es mejor en el micrófono acoplado a la EasyVR.
- Al utilizar los sensores de distancia en las posiciones indicadas se obtuvo muy buena respuesta de tiempo ya que no hay ningún elemento que este delante de los sensores.
- La pantalla LCD nos permite visualizar los comandos al momento de decirlos. Es importante tener la pantalla sobre todo para ver la distancia que hay a los obstáculos más cercanos.

4.2. Recomendaciones

Al momento de realizar las fases de entrenamiento y de reconocimiento de los comandos de voz, se recomienda tener en cuenta las siguientes condiciones.

- Realizar la fase de entrenamiento de los comandos por voz ubicando el micrófono en su posición final.
- Se recomienda grabar los comandos en un lugar con poco ruido exterior para evitar problemas de reconocimiento de voz.
- Se recomienda utilizar palabras cortas que tengan máximo tres sílabas.
- Se recomienda colocar mínimo cuatro sensores de distancia, por lo menos uno para cada dirección, (adelante, atrás, izquierda, derecha).
- Es muy importante revisar el manual del módulo de reconocimiento de voz EasyVR, donde se puede encontrar el listado completo de funciones y programación del mismo que permiten aprovechar al máximo la tarjeta y evitar estropearla con alguna mala manipulación.

- La persona que vaya a ser beneficiada de este sistema debe recibir un entrenamiento previo, es decir, para que pueda modular su voz a un mismo tono, para dar las ordenes al reconocimiento por voz.
- La distancia a la que debe colocarse el micrófono no debe ser ni muy cerca ni muy lejos, se recomienda usarlo a una distancia de 10 cm.
- Los sensores de distancia deben ser ubicados en la silla de tal manera que no tengan obstáculos de la propia silla en frente.
- Los movimientos para los giros deben ser lentos, para que el usuario no sufra movimientos que le hagan perder su posición ideal en la silla.
- Al utilizar los comandos (izquierda y derecha) estos solo permiten hacer girar la silla cinco grados. Lo que resultó ser una buena opción para giros cortos, pero para giros largos no, ya que si se quiere hacer un giro de 180 grados (media vuelta) se requiere decir muchas veces el mismo comando. Es conveniente agregar otro comando que permita hacer girar la silla un ángulo mayor, como por ejemplo 90 grados.

REFERENCIAS BIBLIOGRÁFICAS

[1]. Health, A. (2014). Allina Health. Obtenido de Allina Health: http://www.allinahealth.org/mdex_sp/SD7168G.HTM

[2]. Adolfoneda. (2005). Paraplejia y Cuadriplejia. Obtenido de Paraplejia y Cuadriplejia: <http://paraplejia2.blogspot.com/>

[3]. Bellesi, F., & Ortiz, F. (2008). Reconocimiento de Voz para Aplicación en Domótica.

[4]. Traviezo, L. y. (2015). ACADEMIA. Obtenido de ACADEMIA: http://www.academia.edu/9921461/Loaiza_y_Traviezo

[5]. Samira Grayeb, J. A. (5 de Diciembre de 2008). MICROCONTROLADORES. Obtenido de MICROCONTROLADORES: <http://losmicrocontroladores.blogspot.com/>

[6]. Amangandi, J. (5 de Octubre de 2012). ARDUINO. Obtenido de ARDUINO: <http://jamangandi2012.blogspot.com/2012/10/que-es-arduino-te-lo-mostramos-en-un.html>

[7]. VEER. (n.d.). EasyVR Module. In VEER, EasyVR User Manual Release 3 (p. 46).

[8]. Vasco, U. d. (11 de Abril de 2007). Universidad del País Vasco. Obtenido de <http://www.ehu.eus/acustica/espanol/electricidad/micres/micres.html>

[9]. Movimiento Ondulatorio. (2004). Obtenido de Variación de la velocidad del sonido con la temperatura: http://www.sc.ehu.es/sbweb/fisica_/ondas/acustica/sonido/sonido1.html

[10]. Ecuador, S. (03 de 09 de 2015). StarMedia Ecuador. Obtenido de Pronóstico del clima del tiempo en Guayaquil: <http://clima.starmedia.com/sudamerica/ecuador/guayaquil/>

[11]. COMPUTERWORLD. (24 de Julio de 2013). COMPUTERWORLD España. Obtenido de COMPUTERWORLD España: <http://www.computerworld.es/innovacion/google-adquiere-patentes-de-reconocimiento-de-voz>

[12]. Norvig, P. (11 de Noviembre de 2011). Tecnología y Anuncios de Google. Obtenido de Una mirada dentro de la tecnología de reconocimiento de voz de Google: http://tecnologiayproductosgoogle.blogspot.com/2011/11/una-mirada-dentro-de-la-tecnologia-de_23.html

APÉNDICES

APÉNDICE A

FICHAS TÉCNICA Y CÓDIGO ARDUINO

Joystick Controlador

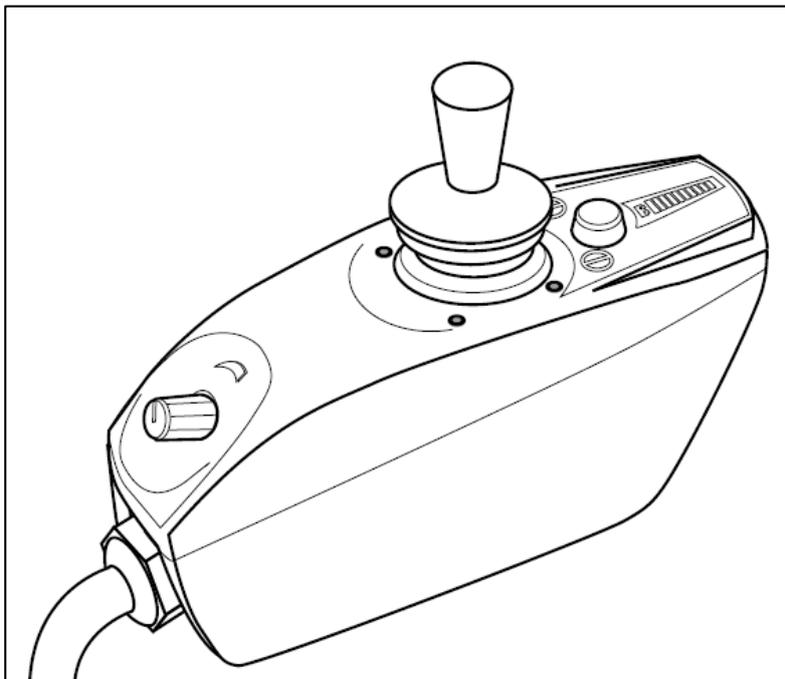


Figura 4.1. Vista del Joystick Controlador

El Joystick controlador como se muestra en la figura 5.1 es un sistema de control electrónico totalmente programable que permite hacer funcionar la silla de ruedas. Está diseñado para que el usuario tenga completo control del movimiento de la silla y la velocidad de la misma. El Joystick ha sido pre – programado para satisfacer las necesidades de un usuario típico.

Características del Joystick Controlador

La figura a continuación proporciona información sobre los componentes y conexiones. Es conveniente utilizar el diagrama para familiarizarse con las funciones y ubicación de cada componente antes de utilizar el Joystick Controlador.

Funciones disponibles para el Joystick Controlador

Joystick.- Se utiliza para controlar la dirección y la velocidad de la silla.

Ajuste de la velocidad.- El usuario puede aumentar o disminuir la velocidad máxima de la silla eléctrica.

Control de batería.- Permite ver el nivel de batería de la silla de ruedas.

Joystick Controlador y conexiones

En la figura 5.2 se muestra un esquema sencillo de sus componentes y su funcionamiento.

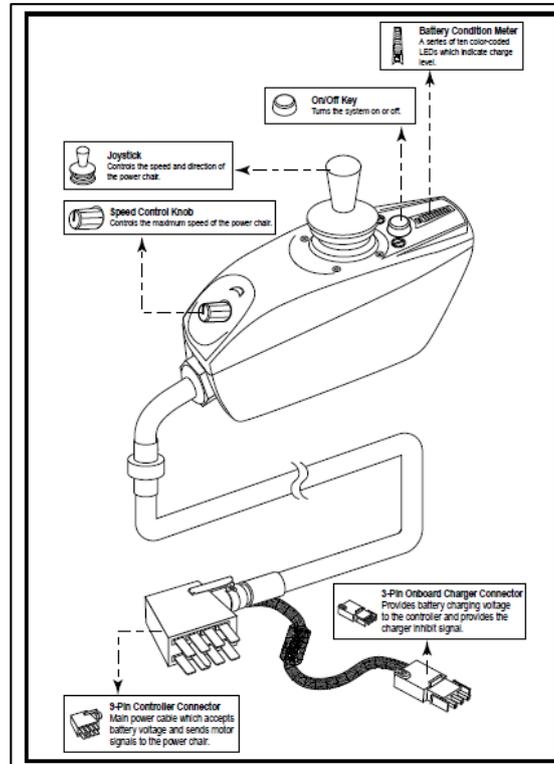


Figura 4.2. Joystick Controlador con sus principales componentes

SENSORES DE DISTANCIA HC – SR204

Características

- Dimensiones del circuito: 43 x 20 x 17 mm
- Tensión de alimentación: 5 Vcc
- Frecuencia de trabajo: 40 KHz
- Rango máximo: 4.5 m
- Rango mínimo: 1.7 cm
- Duración mínima del pulso de disparo (nivel TTL): 10 μ S.
- Duración del pulso eco de salida (nivel TTL): 100-25000 μ S.
- Tiempo mínimo de espera entre una medida y el inicio de otra 20 mS.

Pines de conexión

- VCC
- Trig (Disparo del ultrasonido)
- Echo (Recepción del ultrasonido)
- GND



Figura 4.3. Sensor de Distancia

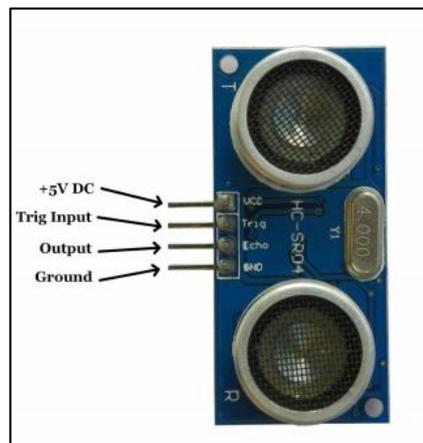


Figura 4.4. Sensor de distancia con sus conexiones

Funcionamiento

El sensor HC – SR04 tal como se muestra en las figura 5.5 consta de emisor de ultrasonidos, el receptor y los circuitos de control. Cuando el Trig envía una serie de pulsos ultrasónicos de 40Khz y recibe el eco (Echo), se calcula la distancia entre la unidad y el objeto midiendo el tiempo de viaje del sonido y la salida como un ancho de pulso TTL.

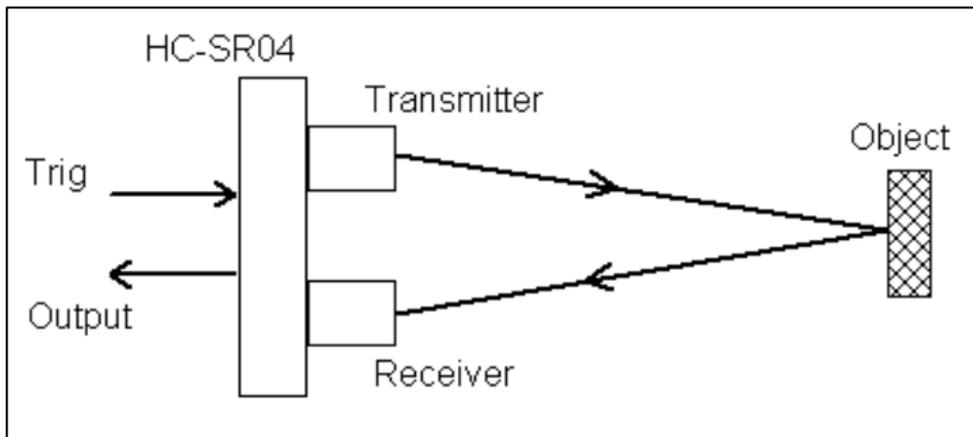


Figura 4.5. Funcionamiento del sensor de distancia

Como usarlo

Para medir la distancia que se necesita para generar una señal Trig y conducirla por la entrada del pin Trig. Los niveles deben estar de acuerdo con los requisitos de nivel TTL (es decir, alto nivel > 2.4V, nivel bajo < 0.8V) y su ancho debe ser mayor que 10µs como se establece en la figura 5.6. Al mismo tiempo se necesita controlar el pin de salida midiendo el ancho de la señal de salida. La distancia detectada se la calcula con la fórmula:

$$Distancia = \frac{Ancho\ de\ Pulso * Velocidad\ del\ Sonido}{2}$$

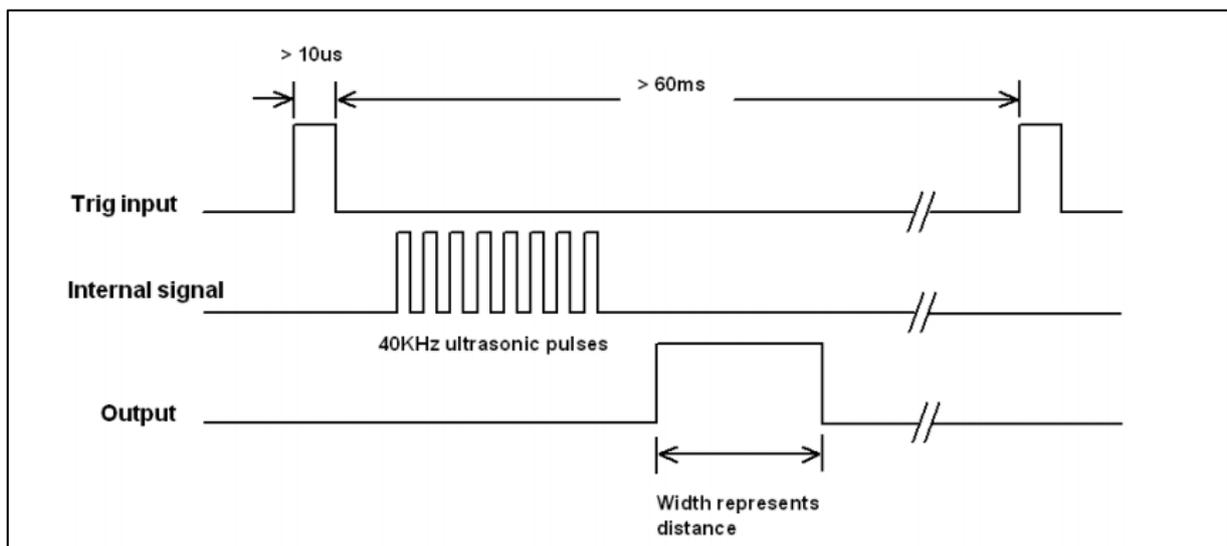


Figura 4.6. Funcionamiento de un PWM

Micro Servo SG90

Pequeño y ligero, con alta potencia de salida, este tipo de servo puede girar aproximadamente 180 grados (90 en cada dirección), y funciona igual que los tipos estándar, pero de menor tamaño como se muestra en la figura 5.7.

Se puede utilizar cualquier código de servo, hardware o una biblioteca para el control de estos servos, hardware o una biblioteca para el control de estos servos.

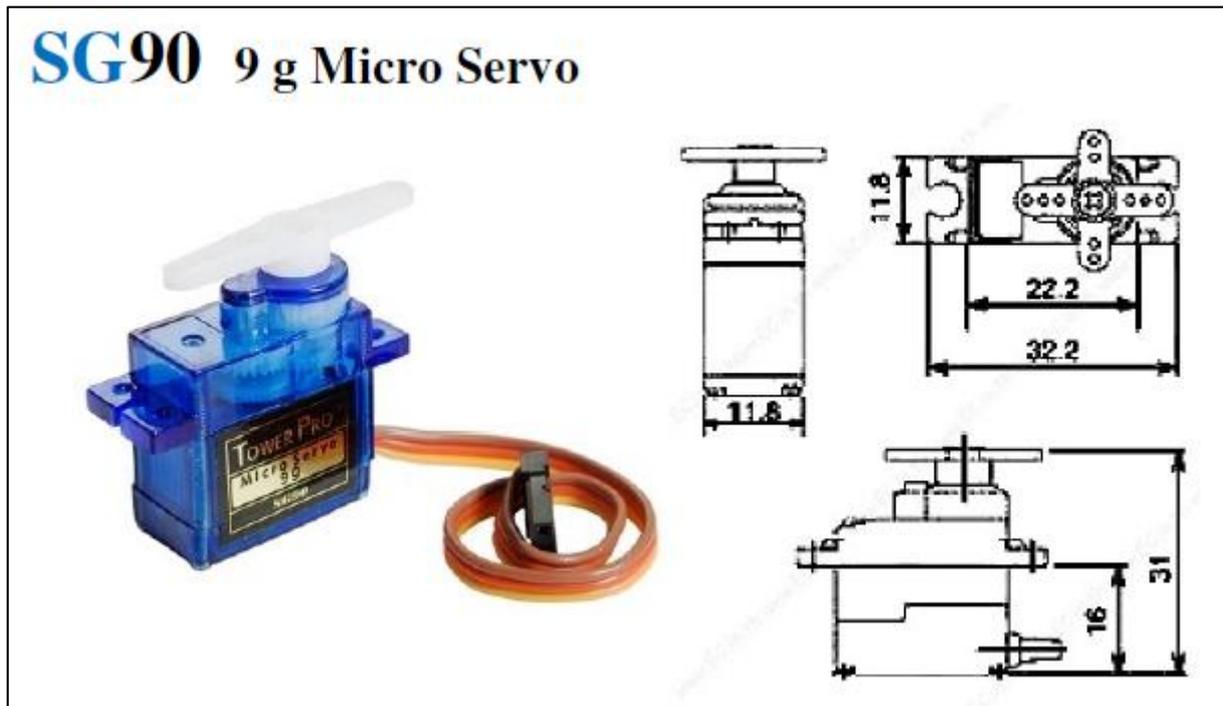


Figura 4.7. Ilustración de un micro servo

Especificaciones

- Peso: 9g
- Dimensiones: 22.2x11.8x31 mm aprox.
- Torque: 1.8 Kgfxcm
- Velocidad de funcionamiento: 0,1 s /60 grados
- Voltaje de Funcionamiento: 4.8 V (5 V)
- Ancho de Banda en caídas: 10 μ
- Rango de temperatura: 0°C – 55°C

El funcionamiento del servomotor y el esquema de conexión para el micro servomotor se muestra en la figura 5.8.

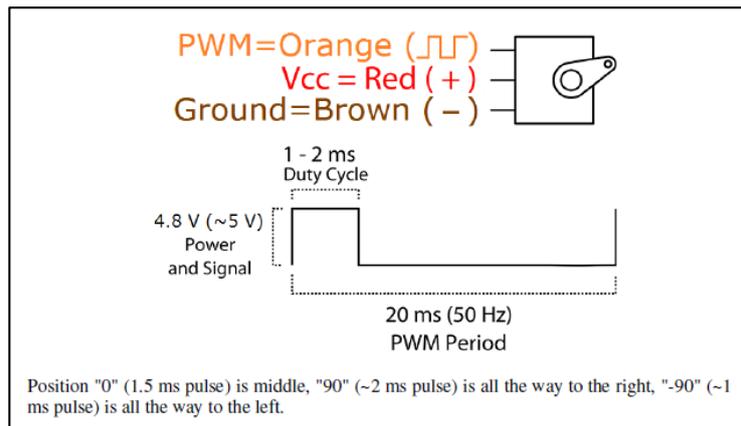


Figura 4.8. PWM servomotor

Reconocimiento de Voz de Google

En el caso del reconocimiento de voz, a diferencia de la traducción automática, se busca encontrar una relación directa entre una onda de sonido y un texto, y no entre dos palabras o frases en distintos idiomas – aunque en la traducción en modo conversación, se requieren ambas cosas funcionando a la par.

En los sistemas del pasado, había que sentarse y entrenar al sistema por horas antes de que pudiera reconocer algunas palabras – y sólo para el usuario que entrenó al sistema, sin tomar en cuenta otros acentos o formas de pronunciar, ni siquiera otros tonos de voz. Por eso quisimos desarrollar algo que cualquiera pudiera usar de forma inmediata, que sólo haya que hablar para que el sistema entienda enseguida. Entonces, en lugar de depender de una persona hablando por un largo tiempo para entrenar al sistema, nos apoyamos en mucha gente diciendo muchas cosas. De este modo, son nuestros usuarios los que están entrenando al sistema en conjunto.

Me tomaré un momento para explicarles a detalle cómo es que en realidad funciona. Son tres partes las que conforman nuestro modelo de reconocimiento de voz. Primero, está el modelo acústico que mapea todas las maneras posibles en que las ondas sonoras pueden formar fonemas, tales como “ah”, “mm” o “bah”. Esto resulta complicado debido a que la acústica varía mucho dependiendo de qué micrófono se utilice, el sonido ambiental, y cómo se esté sosteniendo el dispositivo, el género y edad de quien habla, e incluso los sonidos previos o posteriores a los que se quieren captar. También existen muchas versiones debido a la gran variedad de acentos y dialectos. Sin embargo, con suficientes ejemplos de habla, podemos modelar cuáles son las formas más probables para formar fonemas. Luego los fonemas se reúnen en nuestro modelo léxico, que básicamente es un diccionario sobre cómo pronunciar todas las palabras en un idioma. Esto también toma en cuenta la gran cantidad de diferencias

entre acentos – ya que el modelo sabe que existen múltiples maneras de pronunciar las cosas, y sabe cuáles son las más probables. Tanto “sep-ti-embre” como “set-iembre” darán como resultado “septiembre”, porque el modelo sabe que ambas formas son comunes.

Finalmente, las palabras se entretajan como parte de un modelo de lenguaje que indica qué palabras suelen venir después o antes de otras. Podrán existir dos ondas sonoras que suenen muy similar, como “horas” y “moras”, pero si le antecede la frase “pastel de...” el modelo de lenguaje determinará que “moras” es la palabra correcta. Tenemos mucho texto para entrenar al sistema – para Búsqueda por voz, en donde los usuarios le dicen su búsqueda a Google, entrenamos al modelo con más de 230 mil millones de palabras incluidas en consultas reales hechas al buscador. Adicionalmente, cuando el sistema no reconoce correctamente una frase y el usuario lo corrige, se da un proceso de aprendizaje automático que mejora el modelo de lenguaje.

Por supuesto que todo esto se da de forma anónima – nosotros no conservamos los ejemplos de entrenamiento que pudieran ser relacionados con algún hablante en particular, todo se combina dentro de nuestro modelo general. Sin embargo podemos darles la posibilidad a algunos usuarios de tomar la decisión de participar voluntariamente en el proceso de aprendizaje automático. Pueden habilitar esto y nuestro modelo empezará a aprender cómo es que su voz varía en relación a un modelo base – como por ejemplo algún acento marcado o una voz muy grave. El modelo funciona bien incluso sin que tengan que entrenarlo ustedes mismos, pero tienen la opción de ayudarnos a mejorarlo todavía más.

Pueden probar nuestro sistema de búsquedas por voz en un teléfono Android o en la aplicación de Google Search para iOS o BlackBerry. En Android, además de realizar búsquedas, pueden redactar correos con la voz, o hablar dentro de una aplicación en lugar de usar el teclado – lo hemos agregado al teclado de Android para que puedan hablar en casi cualquier lugar en el que podrían escribir. La Búsqueda por voz también está disponible para los usuarios de Google Chrome desde la computadora de escritorio – por el momento solamente en inglés.

Esta tecnología nos permite entregarles los resultados de búsqueda en un menor tiempo, pero también trabajamos en la integración con otros de nuestros productos, como Google Translate, para ofrecerles la posibilidad de utilizar otros servicios de una forma natural mediante el uso de su voz y desde cualquier dispositivo.

Por Peter Norvig, Director de Investigación en Google [12].

CODIGO ARDUINO

```
#if !defined(SERIAL_PORT_MONITOR)
  #error "Arduino version not supported. Please update your IDE to the latest version."
#endif

#if defined(SERIAL_PORT_USBVIRTUAL)
  // Shield Jumper on HW (for Leonardo and Due)
  #define port SERIAL_PORT_HARDWARE
  #define pcSerial SERIAL_PORT_USBVIRTUAL
#else
  // Shield Jumper on SW (using pins 12/13 or 8/9 as RX/TX)
  #include "SoftwareSerial.h"
  SoftwareSerial port(12, 13);
  #define pcSerial SERIAL_PORT_MONITOR
#endif

#include "EasyVR.h"
#include <Servo.h>          // Incluye la libreria Servo

Servo servo1;
Servo servo2;

#define Pecho1 10
#define Ptrig1 4
long duracion1, distancia1;

#define Pecho2 8
#define Ptrig2 7
long duracion2, distancia2;
int cont=0;
int pinzumbador = 9; // pin del zumbador
int frecuencia = 130;
```

```

EasyVR easyvr(port);

//Groups and Commands
enum Groups
{
    GROUP_0 = 0,
    GROUP_1 = 1,
};

enum Group0
{
    G0_ENCIENDE = 0,
};

enum Group1
{
    G1_MEDIA_VELOCIDAD_ = 0,
    G1_APAGAR_ = 1,
    G1_ALTO_ = 2,
    G1_SIGUE_ = 3,
    G1_ATRAS_ = 4,
    G1_DERECHA_ = 5,
    G1_IZQUIERDA_ = 6,
    G1_ALTA_VELOCIDAD_ = 7,
    G1_BAJA_VELOCIDAD_ = 8,
    G1_ADELANTE = 9,
    G1_IZQUIERDA = 10,
    G1_DERECHA = 11,
    G1_ATRAS = 12,
    G1_ALTO = 13,
    G1_MEDIA_VELOCIDAD = 14,
    G1_ALTA_VELOCIDAD = 15,
    G1_BAJA_VELOCIDAD = 16,
    G1_DERECHA2 = 17,
    G1_IZQUIERDA2 = 18,
    G1_PITA = 19,
};

```

```

G1_MEDIA_DERECHA = 20,
G1_MEDIA_IZQUIERDA = 21,
G1_ADELANTE_KARLA = 22,
G1_ATRAS_KARLA = 23,
G1_DERECHA_KARLA = 24,
G1_IZQUIERDA_KARLA = 25,
G1_ALTO_KARLA = 26,
G1_MEDIA_VELOCIDAD_KARLA = 27,

};

int8_t group, idx;

void setup()
{
  // setup PC serial port
  pcSerial.begin(9600);
  servo1.attach(6,600,1500);
  servo2.attach(5,600,1500);

  pinMode(Pecho1, INPUT); // define el pin 2 omo entrada (echo)
  pinMode(Ptrig1, OUTPUT); // define el pin 3 como salida (trigger)
  pinMode(Pecho2, INPUT); // define el pin 2 omo entrada (echo)
  pinMode(Ptrig2, OUTPUT); // define el pin 3 como salida (trigger)

  pinMode(11,OUTPUT);
  pinMode(3,OUTPUT);

  // bridge mode?
  int mode = easyvr.bridgeRequested(pcSerial);
  switch (mode)
  {
  case EasyVR::BRIDGE_NONE:
    // setup EasyVR serial port
    port.begin(9600);

```

```

// run normally
pcSerial.println(F("---"));
pcSerial.println(F("Bridge not started!"));
break;

case EasyVR::BRIDGE_NORMAL:
// setup EasyVR serial port (low speed)
port.begin(9600);
// soft-connect the two serial ports (PC and EasyVR)
easyvr.bridgeLoop(pcSerial);
// resume normally if aborted
pcSerial.println(F("---"));
pcSerial.println(F("Bridge connection aborted!"));
break;

case EasyVR::BRIDGE_BOOT:
// setup EasyVR serial port (high speed)
port.begin(115200);
// soft-connect the two serial ports (PC and EasyVR)
easyvr.bridgeLoop(pcSerial);
// resume normally if aborted
pcSerial.println(F("---"));
pcSerial.println(F("Bridge connection aborted!"));
break;
}

while (!easyvr.detect())
{
Serial.println("EasyVR not detected!");
delay(1000);
}

easyvr.setPinOutput(EasyVR::IO1, LOW);
Serial.println("EasyVR detected!");
easyvr.setTimeout(5);
easyvr.setLanguage(0);

```

```

    group = EasyVR::TRIGGER; //<-- start group (customize)
}

void action();

void loop()
{
    easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED on (listening)

    Serial.print("Say a command in Group ");
    Serial.println(group);
    easyvr.recognizeCommand(group);

    do
    {
        // can do some processing while waiting for a spoken command
        digitalWrite(Ptrig1, LOW);
        delayMicroseconds(2);
        digitalWrite(Ptrig1, HIGH); // genera el pulso de trigger por 10ms
        delayMicroseconds(10);
        digitalWrite(Ptrig1, LOW);

        duracion1 = pulseIn(Pecho1, HIGH);
        distancia1 = (duracion1/2) / 29;

        digitalWrite(Ptrig2, LOW);
        delayMicroseconds(2);
        digitalWrite(Ptrig2, HIGH); // genera el pulso de trigger por 10ms
        delayMicroseconds(10);
        digitalWrite(Ptrig2, LOW);

        duracion2 = pulseIn(Pecho2, HIGH);
        distancia2 = (duracion2/2) / 29;
    }
}

```

```

//Serial.print(distancia2);
Serial.println("cm");

Serial.print(distancia2);
Serial.println("cm");

if (distancia2 < 20 ){
  cont=cont+1;
  // if (cont>2){
    servo2.write(90);
    servo1.write(90);
    //cont=0;}
  // if (distancia2 < 5 ){
    // cont=0;}
  }
}
while (!easyvr.hasFinished());

easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off

idx = easyvr.getWord();
if (idx >= 0)
{
  // built-in trigger (ROBOT)
  // group = GROUP_X; <-- jump to another group X
  return;
}
idx = easyvr.getCommand();
if (idx >= 0)
{
  // print debug message
  uint8_t train = 0;
  char name[32];
  Serial.print("Command: ");
  Serial.print(idx);
  if (easyvr.dumpCommand(group, idx, name, train))

```

```

{
  Serial.print(" = ");
  Serial.println(name);
}
else
  Serial.println();
easyvr.playSound(0, EasyVR::VOL_FULL);
// perform some action
action();
}
else // errors or timeout
{
  if (easyvr.isTimeout())
    Serial.println("Timed out, try again...");
  int16_t err = easyvr.getError();
  if (err >= 0)
  {
    Serial.print("Error ");
    Serial.println(err, HEX);
  }
}
}

void action()
{
  switch (group)
  {
  case GROUP_0:
    switch (idx)
    {
    case G0_ENCLENDE:
      // write your action code here
      digitalWrite(11,HIGH);
      delay(1000);
      digitalWrite(11,LOW);
      Serial.println('1');
    }
  }
}

```

```

servo2.write(90);
servo1.write(90);
group = GROUP_1;// <-- or jump to another group X for composite commands
break;
}
break;
case GROUP_1:
switch (idx)
{
case G1_MEDIA_VELOCIDAD_:
analogWrite(3,100);
group = GROUP_1;// <-- or jump to another group X for composite commands
break;
case G1_APAGAR_:
// write your action code here
Serial.println('3');
digitalWrite(11,HIGH);
delay(1000);
digitalWrite(11,LOW);
servo2.write(90);
servo1.write(90);
group = GROUP_0; //<-- or jump to another group X for composite commands
break;
case G1_ALTO_:
// write your action code here
Serial.println('7');
servo2.write(90);
servo1.write(90);
group = GROUP_1;// <-- or jump to another group X for composite commands
break;
case G1_SIGUE_:
Serial.println('2');
servo1.write(135); //adelante
servo2.write(105);
group = GROUP_1;// <-- or jump to another group X for composite commands

```

```

break;
case G1_ATRAS_:
  Serial.println('4');
  servo1.write(40); //atras
  servo2.write(85);
  group = GROUP_1;// <-- or jump to another group X for composite commands
  break;
case G1_DERECHA_:
// write your action code here
  Serial.println('5');
  analogWrite(3,100);
  servo1.write(100); //derecha
  servo2.write(50);
  delay(700);
  servo2.write(90);
  servo1.write(90);
  group = GROUP_1;// <-- or jump to another group X for composite commands
  break;
case G1_IZQUIERDA_:
  Serial.println('6');
  analogWrite(3,100);
  servo1.write(95); //izquierda      //Gira el servo a 180 grados
  servo2.write(145);
  delay(700);
  servo2.write(90);
  servo1.write(90);
  group = GROUP_1; //<-- or jump to another group X for composite commands
  break;
case G1_ALTA_VELOCIDAD_:
  analogWrite(3,240);
  group = GROUP_1;// <-- or jump to another group X for composite commands
  break;
case G1_BAJA_VELOCIDAD_:
// write your action code here
  analogWrite(3,30);
  group = GROUP_1;// <-- or jump to another group X for composite commands

```

```

    break;
case G1_ADELANTE:
    Serial.println('2');
    servo1.write(135); //adelante
    servo2.write(105);
    group = GROUP_1; // <-- or jump to another group X for composite commands
    break;
case G1_IZQUIERDA:
Serial.println('6');
    analogWrite(3,100);
    servo1.write(95); //izquierda //Gira el servo a 180 grados
    servo2.write(145);
    delay(700);
    servo2.write(90);
    servo1.write(90);
    group = GROUP_1; //<-- or jump to another group X for composite commands
    break;
case G1_DERECHA:
    // write your action code here
    Serial.println('5');
    analogWrite(3,100);
    servo1.write(100); //derecha
    servo2.write(50);
    delay(700);
    servo2.write(90);
    servo1.write(90);
    group = GROUP_1; // <-- or jump to another group X for composite commands
    break;
case G1_ATRAS:
    Serial.println('4');
    servo1.write(40); //atras
    servo2.write(85);
    group = GROUP_1; // <-- or jump to another group X for composite commands
    break;
case G1_ALTO:
    // write your action code here

```

```

Serial.println('7');
servo2.write(90);
servo1.write(90);
group = GROUP_1;// <-- or jump to another group X for composite commands
break;
case G1_MEDIA_VELOCIDAD:
analogWrite(3,100);
group = GROUP_1;// <-- or jump to another group X for composite commands
break;
case G1_ALTA_VELOCIDAD:
analogWrite(3,240);
group = GROUP_1;// <-- or jump to another group X for composite commands
break;
case G1_BAJA_VELOCIDAD:
// write your action code here
analogWrite(3,30);
group = GROUP_1;// <-- or jump to another group X for composite commands
break;
case G1_DERECHA2:
// write your action code here
Serial.println('5');
analogWrite(3,100);
servo1.write(100); //derecha
servo2.write(50);
delay(700);
servo2.write(90);
servo1.write(90);
group = GROUP_1;// <-- or jump to another group X for composite commands
break;
case G1_IZQUIERDA2:
Serial.println('6');
analogWrite(3,100);
servo1.write(95); //izquierda //Gira el servo a 180 grados
servo2.write(145);
delay(700);
servo2.write(90);

```

```

servo1.write(90);
group = GROUP_1; //<-- or jump to another group X for composite commands
break;
case G1_PITA:
tone(pinzumbador,frecuencia); // inicia el zumbido
delay(3000);
noTone(pinzumbador); // lo detiene a los dos segundos
delay(2000);
analogWrite(9,LOW);
group = GROUP_1; // <-- or jump to another group X for composite commands
break;

case G1_MEDIA_DERECHA:
Serial.println('5');
analogWrite(3,100);
servo1.write(100); //derecha
servo2.write(50);
delay(2000);
servo2.write(90);
servo1.write(90);

group = GROUP_1; // <-- or jump to another group X for composite commands
break;

case G1_MEDIA_IZQUIERDA:
Serial.println('6');
analogWrite(3,100);
servo1.write(95); //izquierda //Gira el servo a 180 grados
servo2.write(145);
delay(2000);
servo2.write(90);
servo1.write(90);

group = GROUP_1; // <-- or jump to another group X for composite commands
break;

```

```

case G1_ADELANTE_3 :
  Serial.println('2');
  servo1.write(135); //adelante
  servo2.write(105);
  group = GROUP_1; // <-- or jump to another group X for composite commands
  break;

case G1_ATRAS_3 :
  Serial.println('4');
  servo1.write(40); //atras
  servo2.write(85);
  group = GROUP_1; // <-- or jump to another group X for composite commands
  break;

case G1_DERECHA_3 :
  // write your action code here
  Serial.println('5');
  analogWrite(3,100);
  servo1.write(100); //derecha
  servo2.write(50);
  delay(700);
  servo2.write(90);
  servo1.write(90);
  group = GROUP_1; // <-- or jump to another group X for composite commands
  break;

case G1_IZQUIERDA_3 :
  Serial.println('6');
  analogWrite(3,100);
  servo1.write(95); //izquierda //Gira el servo a 180 grados
  servo2.write(145);
  delay(700);
  servo2.write(90);
  servo1.write(90);
  group = GROUP_1; //<-- or jump to another group X for composite commands
  break;

```

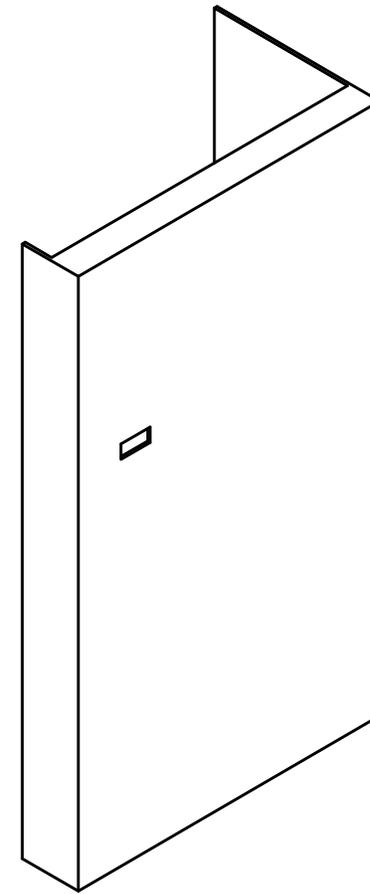
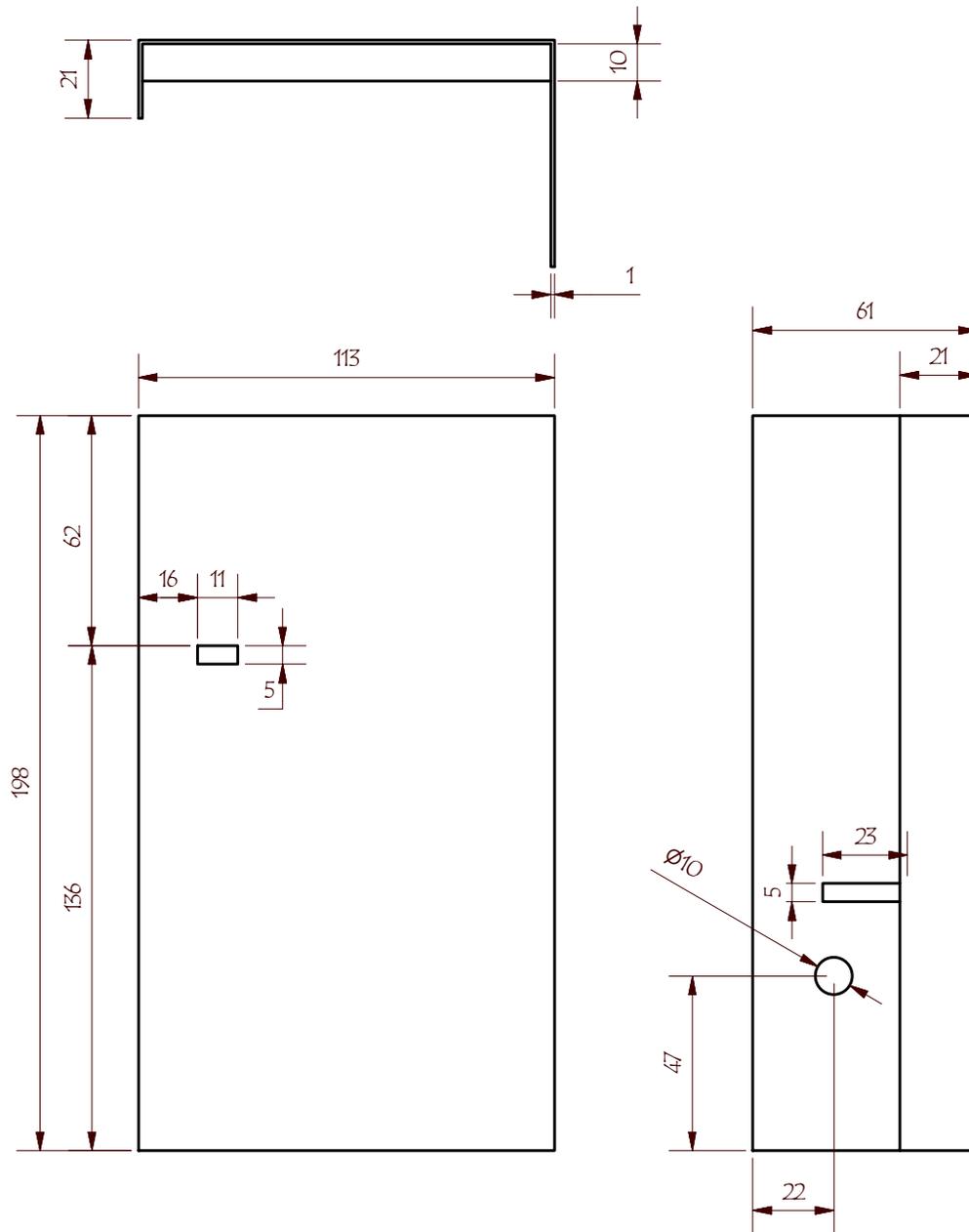
```
case G1_ALTO_3 :
  Serial.println('7');
  servo2.write(90);
  servo1.write(90);
  group = GROUP_1;// <-- or jump to another group X for composite commands
  break;

case G1_MEDIA_VELOCIDAD_3 :
  analogWrite(3,100);
  group = GROUP_1;// <-- or jump to another group X for composite commands
  break;

}
break;
}
}
```

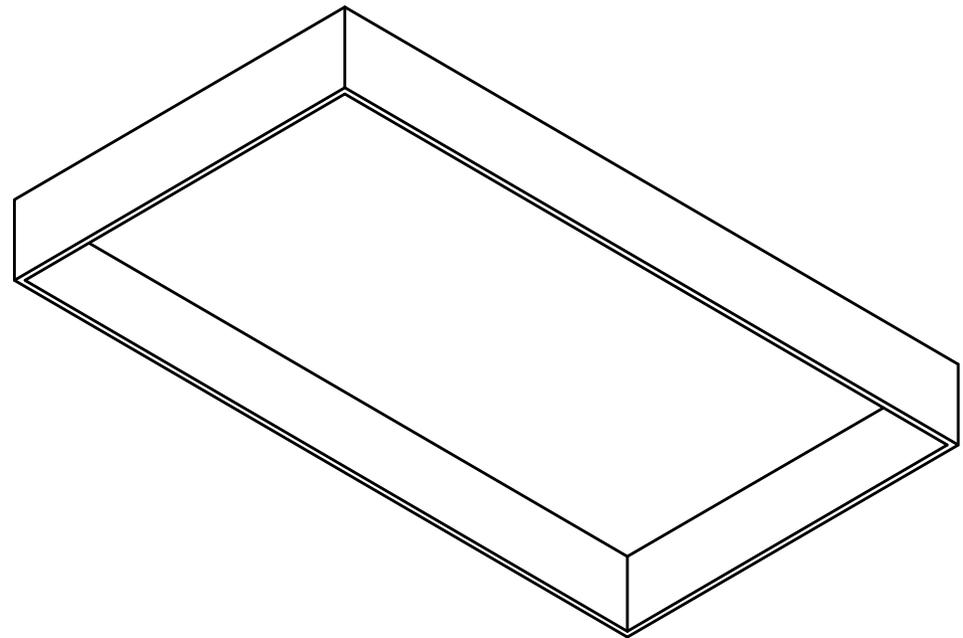
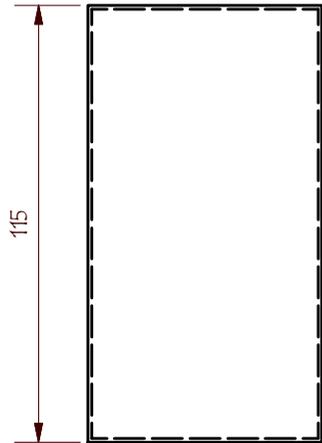
APÉNDICE B

Planos Esquemáticos

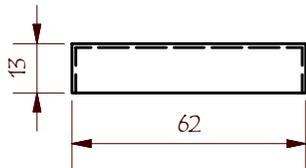


Isométrico

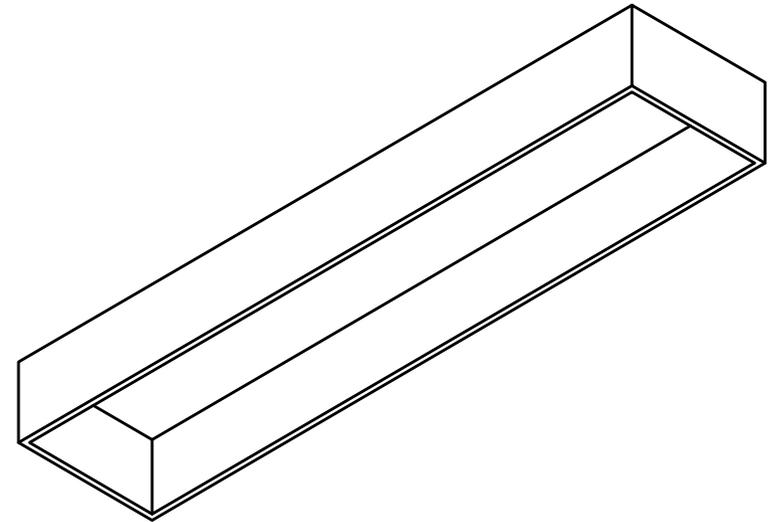
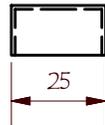
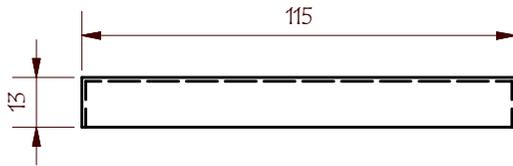
Nombre del Archivo		Base metálica lateral derecha	
Nombre de los estudiantes		Álvaro Chévez Baquerizo Israel Silva Rosado	
Descripción		Parte	
Plancha metálica de dimensiones 198x195 y espesor de 1 mm		2/10	
Proyección	Escala:	Material	Fecha
	1:2	Aluminio	08/09/2015



Isométrico

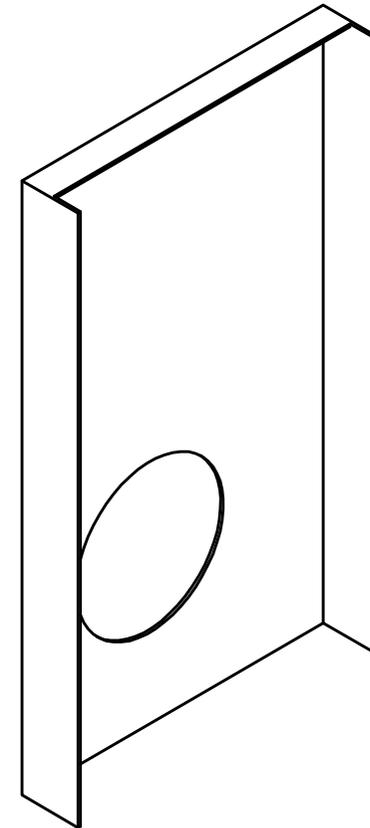
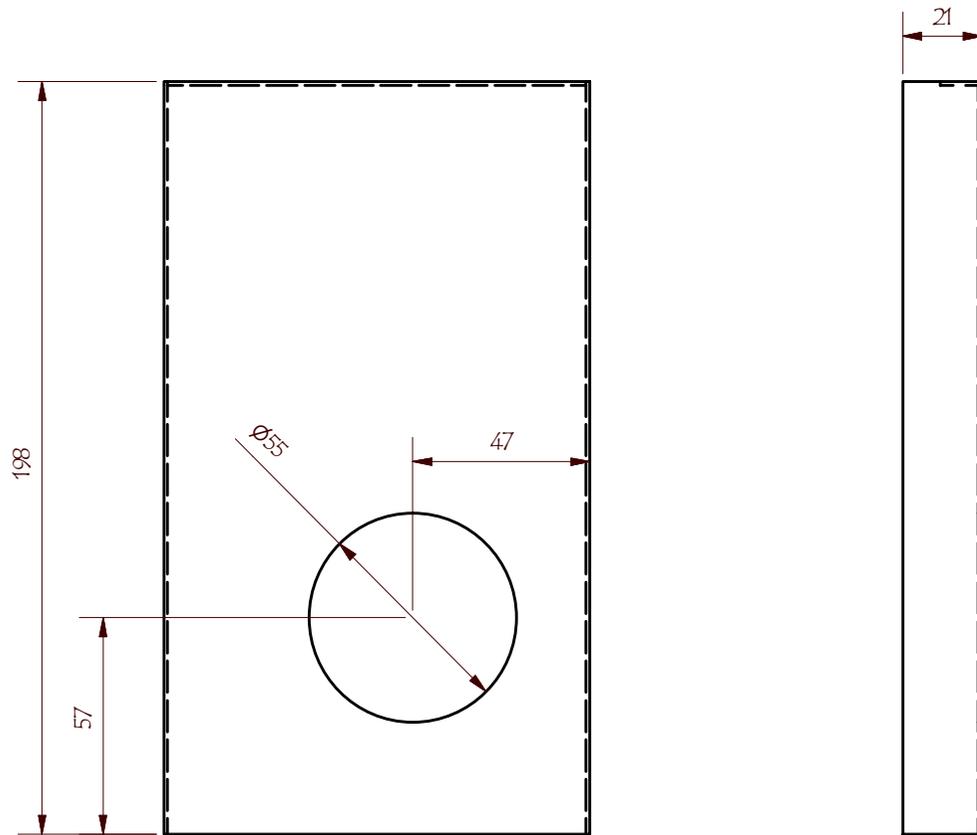
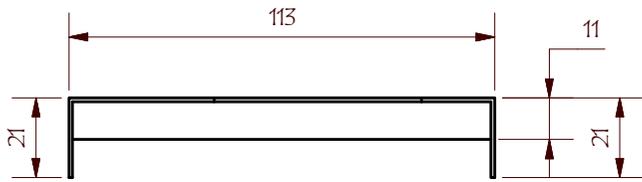


Nombre del Archivo		Tapa Inferior Derecha	
Nombre de los estudiantes		Álvaro Chávez Baquerizo Israel Silva Rosado	
Descripción		Parte	
Tapa Inferior ubicada a la derecha del Joystick Controlador con espero de plancha de 1 mm		6/10	
Proyección	Escala:	Material	Fecha
	1:2	Aluminio	08/09/2015



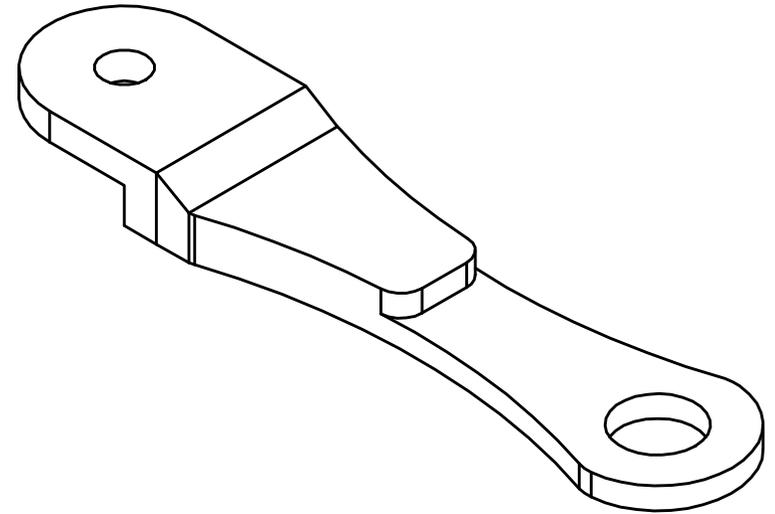
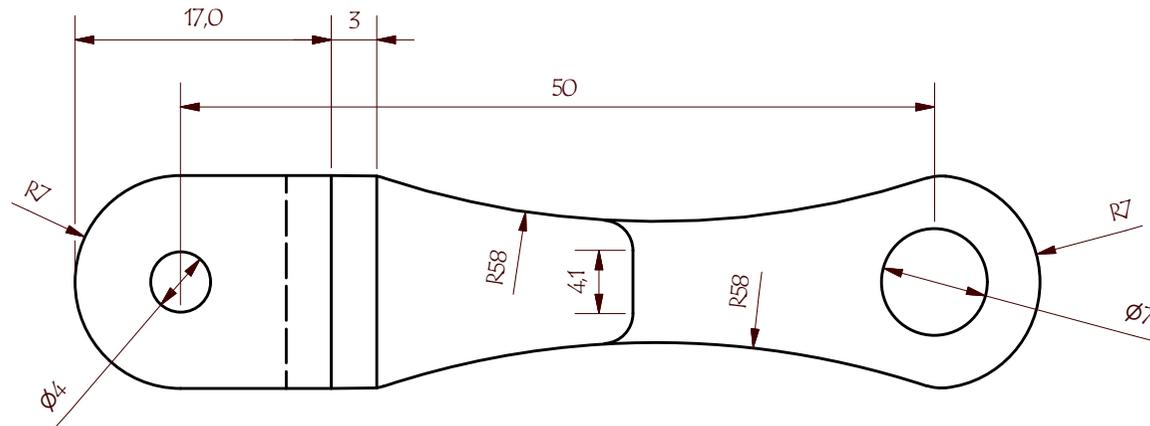
Isométrico
escala 1:1

Nombre del Archivo				Tapa Inferior Izquierda			
Nombre de los estudiantes				Álvaro Chévez Baquerizo Israel Silva Rosado			
Descripción			Pequeña caja metalica es espesor 1 mm		Parte		5/10
Proyección		Escala:		Material		Fecha	
		1:2		Aluminio		08/09/2015	

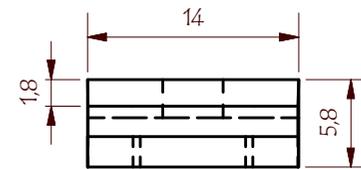
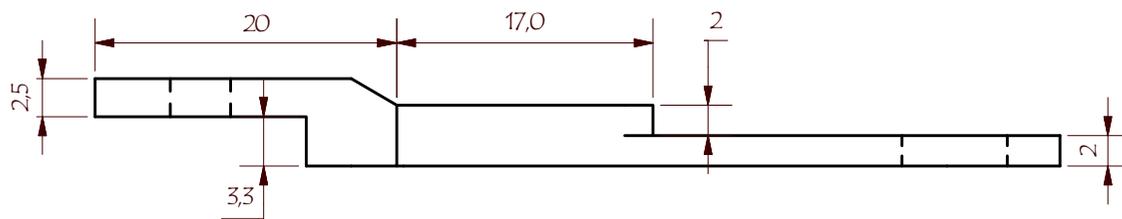


Isométrico

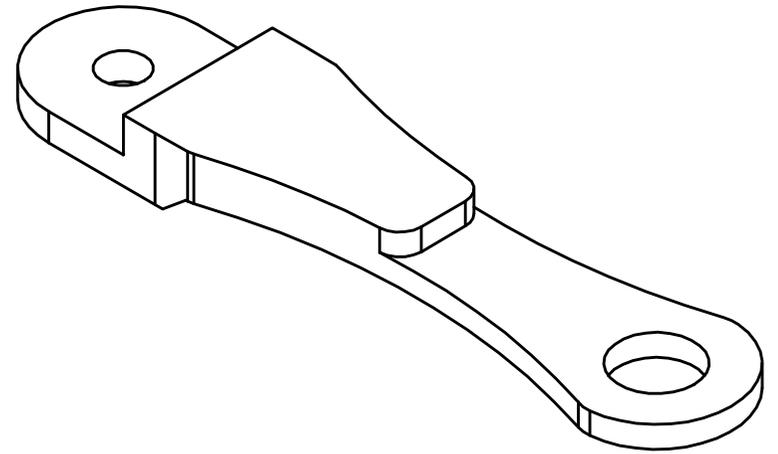
Nombre del Archivo		Base metálica lateral Izquierda	
Nombre de los estudiantes		Álvaro Chévez Baquerizo Israel Silva Rosado	
Descripción		Parte	
Plancha metálica de dimensiones 198x155 y espesor 1 mm		3/10	
Proyección	Escala:	Material	Fecha
	1:2	Aluminio	08/09/2015



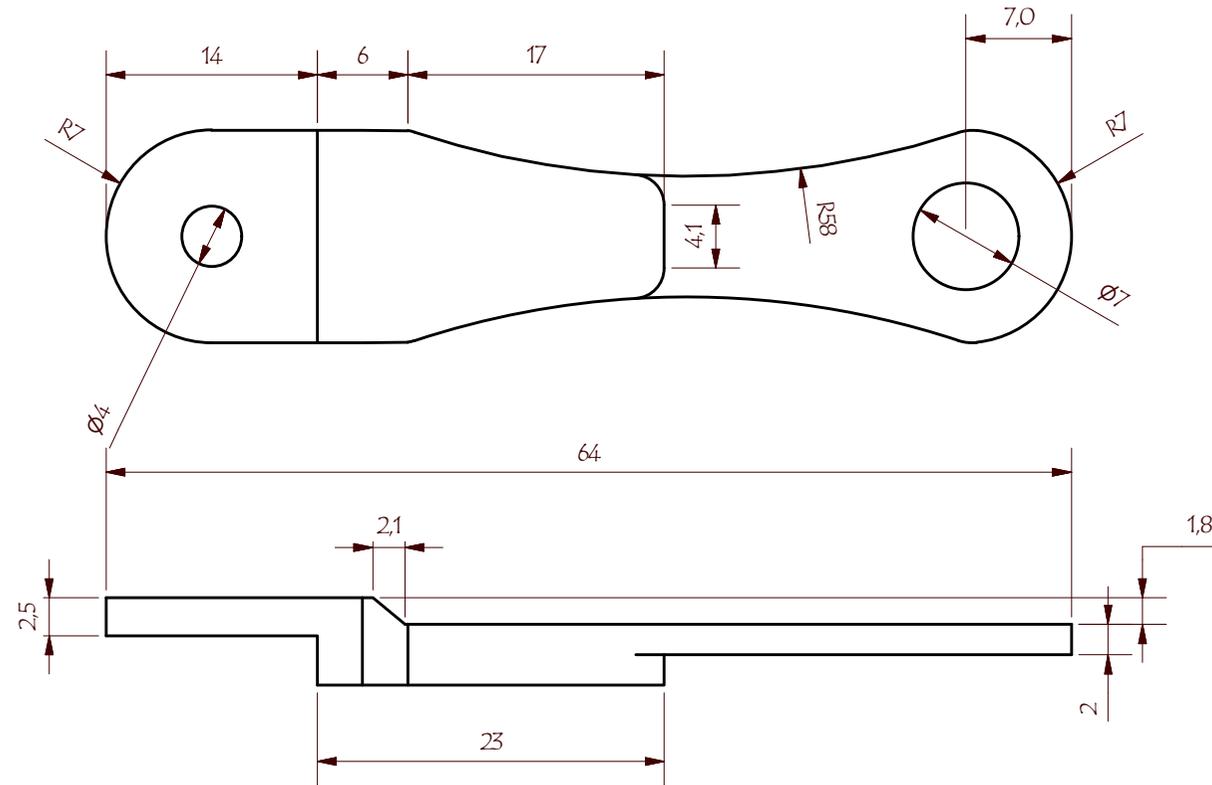
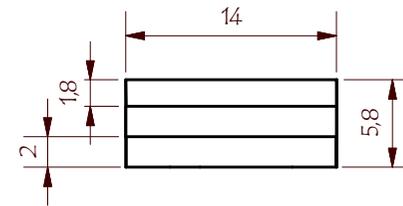
Isométrico



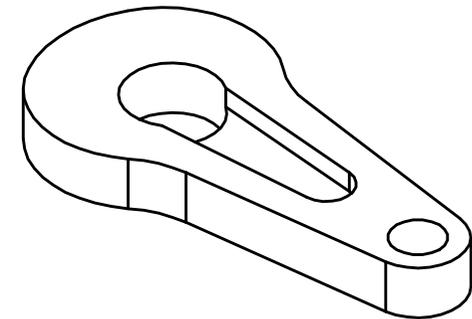
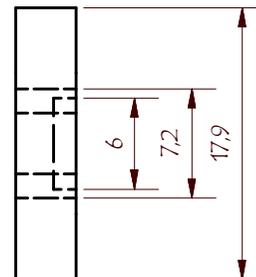
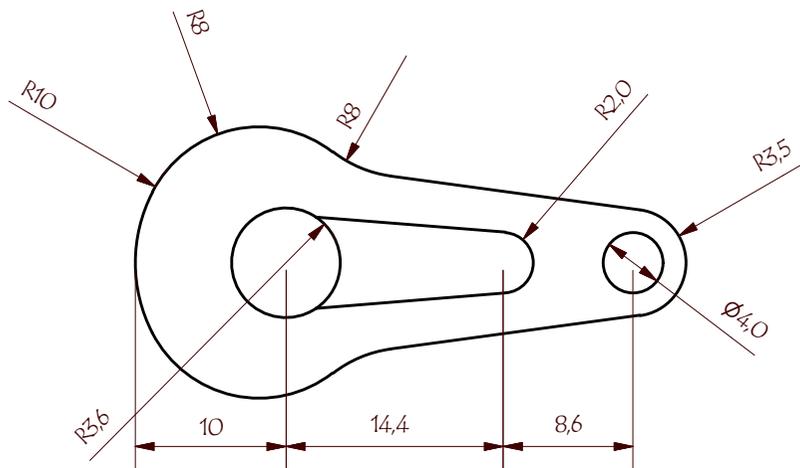
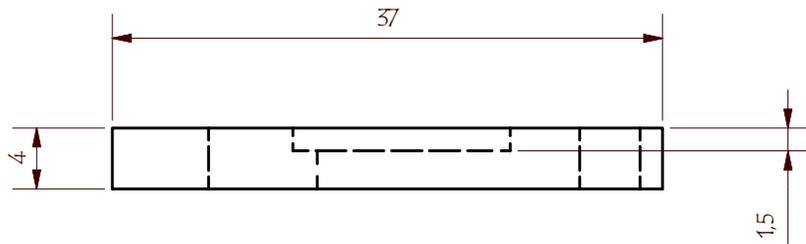
Nombre del Archivo		Eslabón Derecho	
Nombre de los estudiantes		Álvaro Chávez Baquerizo Israel Silva Rosado	
Descripción		Parte	
Eslabón acoplado al servomotor derecho impreso en plástico estructural		10/10	
Proyección	Escala:	Material	Fecha
	2:1	Plástico	08/09/2015



Isométrico

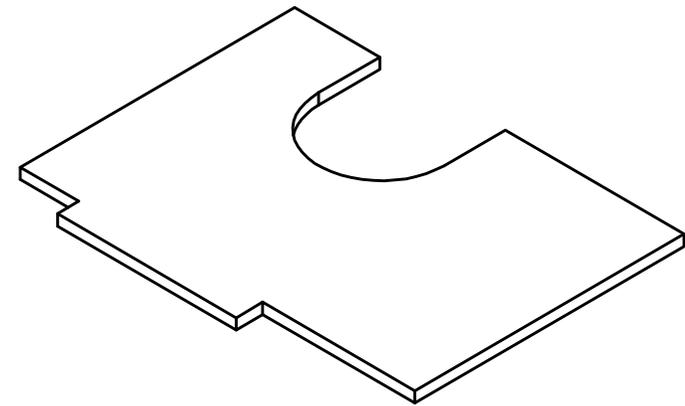
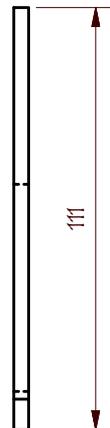
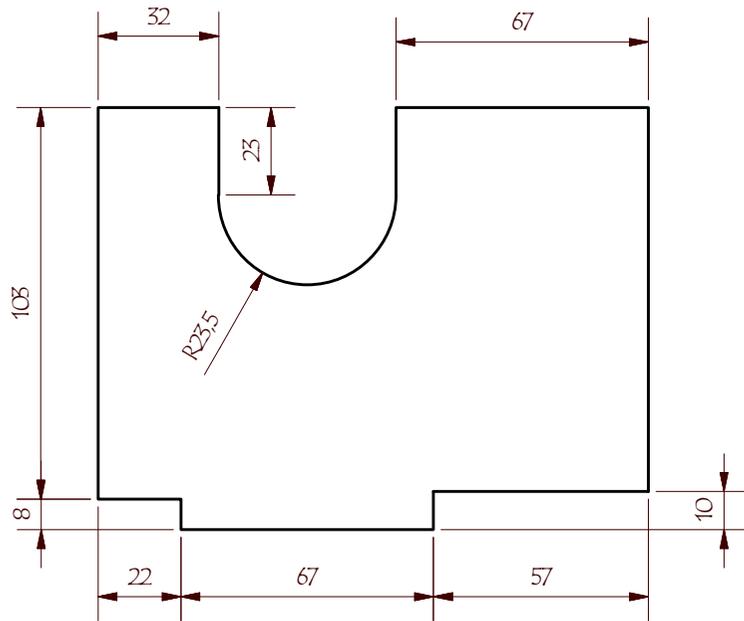
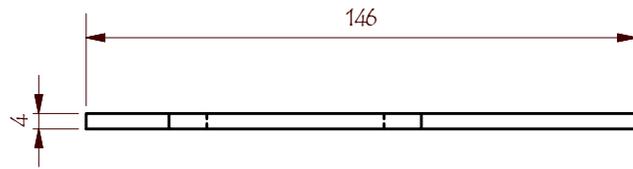


Nombre del Archivo		Eslabon Izquierdo	
Nombre de los estudiantes		Álvaro Chávez Baquerizo Israel Silva Rosado	
Descripción		Eslabon impreso en plástico estructural se acopla a la manivel impresa	Parte 9/10
Proyección	Escala:	Material	Fecha
	2:1	Plástico	08/09/2015



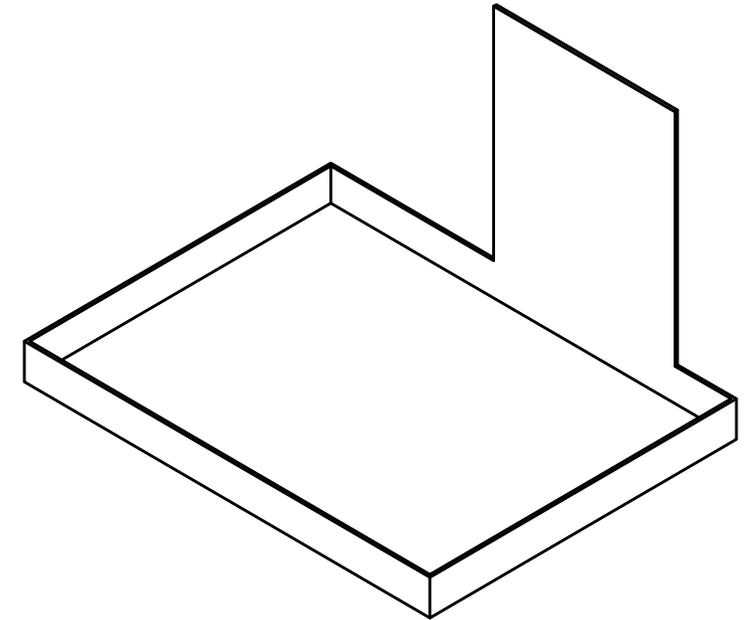
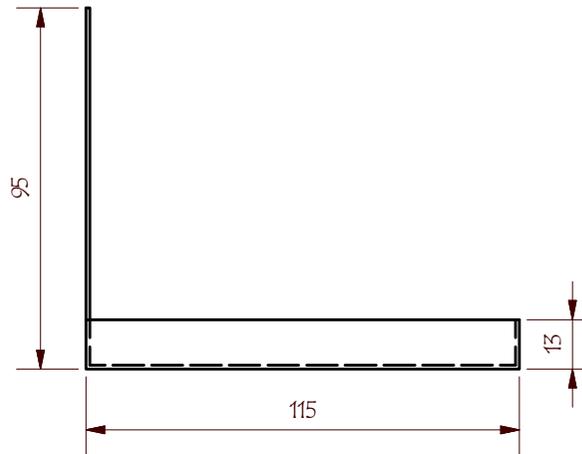
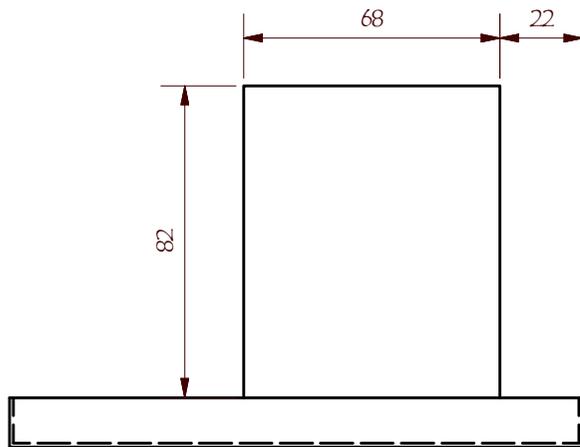
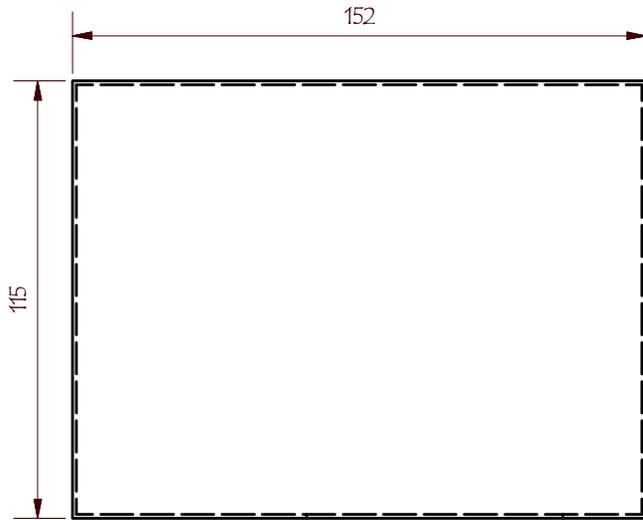
Isométrico

Nombre del Archivo			
Manivela			
Nombre de los estudiantes			
Álvaro Chévez Baquerizo Israel Silva Rosado			
Descripción			Parte
Pequeña eslabón que hace las veces de manivela que va en cada servomotor impresa en plástico estructural			8/10
Proyección	Escala:	Material	Fecha
	2:1	Plástico	08/09/2015



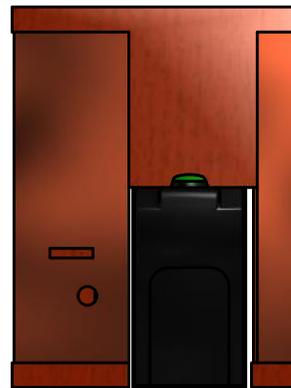
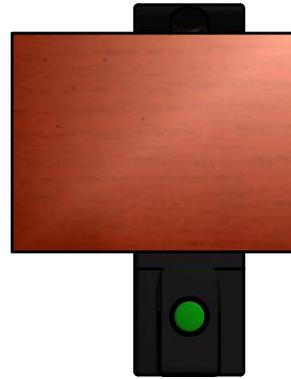
Isométrico

Nombre del Archivo		Plataforma para los servomotores	
Nombre de los estudiantes		Álvaro Chévez Baquerizo Israel Silva Rosado	
Descripción			Parte
Plataforma de madera donde se ubicarán los servomotores debidamente espaciados el espesor de la madera es de 4 mm			7/10
Proyección	Escala:	Material	Fecha
	1:2	Madera	08/09/2015

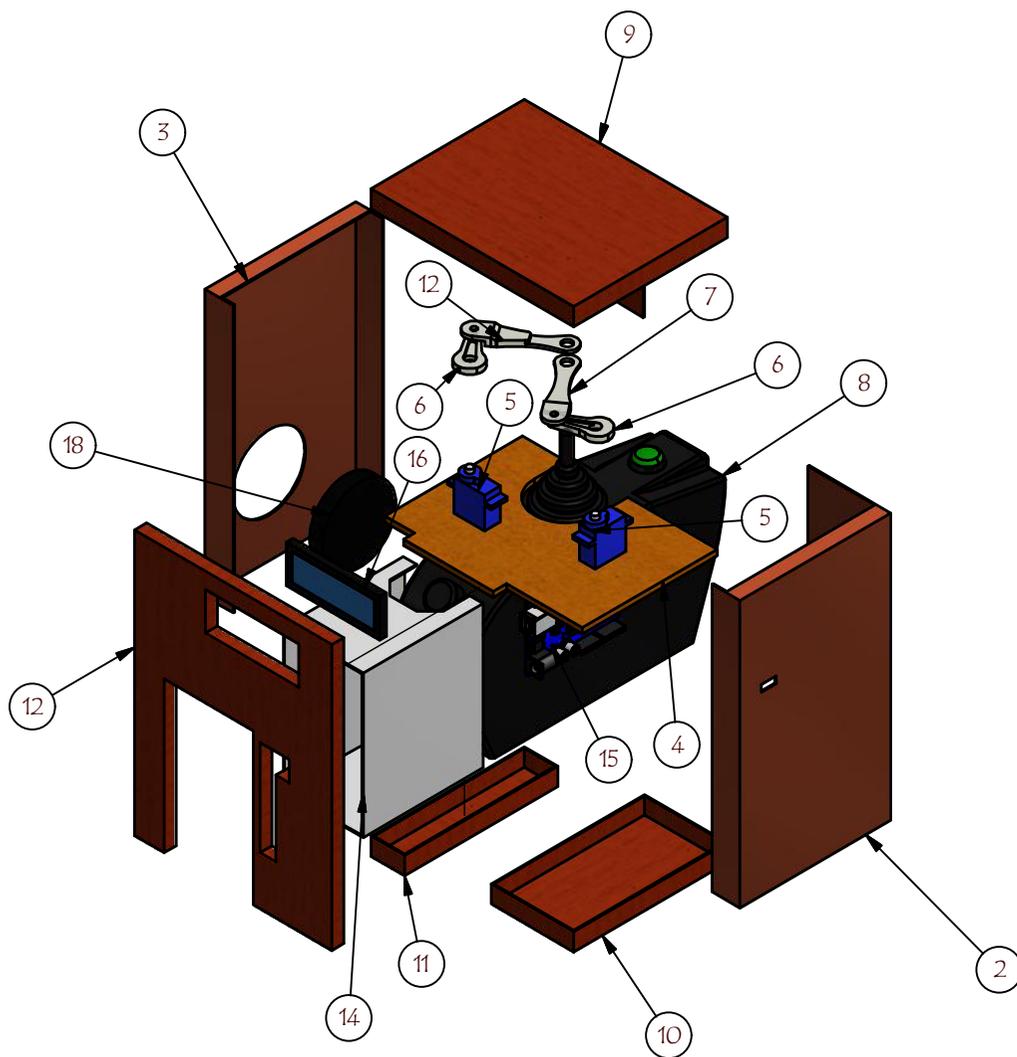


Isométrico

Nombre del Archivo				Tapa Superior			
Nombre de los estudiantes				Álvaro Chévez Baquerizo Israel Silva Rosado			
Descripción						Parte	
Tapa Superior metálica con un espesor de plancha de 1 mm						4/10	
Proyección		Escala:		Material		Fecha	
		1:2		Aluminio		08/09/2015	



Nombre del Archivo		Sistema Completo	
Nombre de los estudiantes		Álvaro Chávez Baquerizo Israel Silva Rosado	
Descripción			Parte
Sistema Completo que se va a adaptar a la silla de ruedas eléctrica			
Proyección	Escala:	Material	Fecha
	Varias	Madera y Aluminio	08/09/2015



LISTA DE PIEZAS			
ELEMENTO	CTDAD	Nº DE PIEZA	DESCRIPCIÓN
12	1	Base de Madera	
2	1	Base Derecha	
3	1	Base Izquierda	
4	1	Plataforma de Madera	
5	2	Servomotor	
6	2	Manivela	
7	1	Eslabon Derecho	
8	1	Joystick Controlador	
9	1	Tapa Superior	
10	1	Base inferior derecha	
11	1	Base inferior izquierda	
12	1	Eslabon Izquierdo	
14	1	Base para arduino	
15	1	Arduino uno	
16	1	Pantalla LCD	
18	1	Parlante	

Nombre del Archivo		Vista del Ensamble	
Nombre de los estudiantes		Álvaro Chávez Baquerizo Israel Silva Rosado	
Descripción		Parte	
Vista del Ensamble del Sistema la ubicacion de cada parte			
Proyección	Escala:	Material	Fecha
	1:4	Varios	08/09/2015