

MANEJO DE LOS MOTORES DE UNA SILLA DE RUEDAS
ELÉCTRICA MEDIANTE COMANDOS DE VOZ.

OSKAR ALONSO VARGAS FAGUA
RUTH MAYORLY SÁNCHEZ FLORES

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA ELECTRÓNICA
BUCARAMANGA
2014

MANEJO DE LOS MOTORES DE UNA SILLA DE RUEDAS
ELÉCTRICA MEDIANTE COMANDOS DE VOZ.

OSKAR ALONSO VARGAS FAGUA
RUTH MAYORLY SÁNCHEZ FLORES

TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE:
INGENIERO ELECTRÓNICO

DIRECTOR
FABIO ALONSO GUZMÁN SERNA
Ingeniero Electrónico y Telecomunicaciones

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA ELECTRÓNICA
BUCARAMANGA
2014

DEDICATORIA

Dedicado a Dios por estar incondicionalmente con nosotros en el transcurso de este tiempo, a nuestros padres que hicieron todo en la vida por ver convertir en realidad nuestros sueños., a nuestros hermanos y hermanas por creer en nosotros.

Con todo el cariño y amor.

AGRADECIMIENTOS

Agradecemos primeramente a Dios por darnos vida, sabiduría y fuerzas para culminar este trabajo.

A nuestros padres por esforzarse y darnos su apoyo incondicional, sembrando en nosotros un futuro mejor.

A nuestros familiares y amigos por todo el cariño, compañía y aportes recibidos que nos impulsaron a este momento.

Al Ingeniero Fabio Guzmán Serna, director del proyecto, por sus recomendaciones, enseñanzas, horas de trabajo y toda su colaboración

A Jesús Eduardo Mateus, estudiante de ingeniería Civil de la Universidad Pontificia Bolivariana, quien como discapacitado colaboró en la realización de las pruebas finales de la silla de ruedas.

A María Camila García Díaz, estudiante de comunicación social, encargada de la emisora de la Universidad Pontificia Bolivariana, quien con mucha sencillez nos colaboró grabando los comandos de voz de advertencia del sistema.

A Daniel Rueda, Karen Castellanos, Daniel Ramos, Vanessa Arango, Alberto Archila, estudiantes de la Universidad Pontificia Bolivariana, quienes colaboraron realizando las pruebas de reconocimiento de voz con el Shield de Arduino EasyVR.

TABLA DE CONTENIDO

1. INTRODUCCIÓN	13
2. OBJETIVOS	14
2.1. Objetivo general.....	14
2.2. Objetivos específicos	14
3. MARCO TEÓRICO.....	15
3.1. DISCAPACIDAD MOTRIZ	15
3.2. ARDUINO	19
3.3. MÓDULO DE RECONOCIMIENTO DE VOZ Y REPRODUCCIÓN, EASYVR	20
3.4. SENSORES DE PROXIMIDAD	22
3.5. DRIVER ROBOCLAW 2X30A.....	22
3.5.1. Modo 1 – Entrada RC.....	24
3.5.2. Modo 2 - Análogo	24
3.5.3. Modo 3 - Serial	24
3.5.4. Modo 4 – Paquetes seriales	24
4. DESARROLLO.....	25
4.1. DIAGRAMA DE BLOQUES	26
4.2. SRF02, SENSOR ULTRASÓNICO.....	27
4.2.1. Ubicación de los sensores ultrasónicos en la silla de ruedas.....	27
4.2.2. Ancho del haz ultrasónico	28
4.2.3. Calibración del sensor SRF02.....	33
4.2.4. Comunicación I2C	40
4.3. Comunicación I2C entre sensores SRF02 y Arduino.....	42
4.3.1. Librería Wire.....	43
4.4. CONFIGURACIÓN DEL DRIVER ROBOCLAW 2X30A [10]	43
4.4.1. Modo 1: Entrada RC.....	43
4.4.2. Conexiones del driver RoboClaw 2x30A	45
4.4.3. Estado y error de los leds.....	48
4.5. CONFIGURACIÓN DEL MÓDULO DE RECONOCIMIENTO DE VOZ Y REPRODUCCIÓN EASYVR [11].....	49

4.5.1.	EasyVR commander: INTERFAZ GRÁFICO DE USUARIO GUI ..	51
4.5.2.	Voces definidas por el usuario	52
4.5.3.	Comandos de voz implementados para el funcionamiento de la silla	54
4.5.4.	Mensajes de realimentación.....	54
4.5.5.	Pruebas con los comandos predefinidos del sistema de reconocimiento de voz EasyVR.....	54
4.5.6.	Prueba con comandos personalizados del sistema de reconocimiento de voz EasyVR.....	57
4.6.	COMUNICACIÓN ENTRE EL ARDUINO MEGA Y EL ANTDUINO	57
5.	CONCLUSIONES.....	59
6.	RECOMENDACIONES	60
7.	BIBLIOGRAFÍA	61
8.	ANEXOS	62
8.1.	DIAGRAMA DE FLUJO, TARJETA ANTDUINO.....	62
8.2.	DIAGRAMA DE FLUJO, TARJETA ARDUINO MEGA.....	64

LISTA DE FIGURAS

Figura 1. Médula espinal, imagen tomada de ADAM.....	16
Figura 2. El cerebelo, imagen tomada de: ADAM.....	17
Figura 3. Hemisferios cerebrales, imagen tomada de: Discapacidad motriz, Guía didáctica para la inclusión en educación inicial y básica, pag. 17.	17
Figura 4. La corteza cerebral, imagen tomada de: Biología de los animales, Capítulo 49. Integración y control. IV: el cerebro de los vertebrados, corteza cerebral.....	18
Figura 5. Arduino MEGA, imagen tomada de: www.arduino.cc	20
Figura 6. Módulo EasyVR, imagen tomada de: guía rápida del módulo de reconocimiento de voz y reproducción, EasyVR.....	21
Figura 7. DIP Switch. Imagen tomada de: Datasheet driver RoboClaw 2X30A.	23
Figura 8. Tarjeta controladora RoboClaw 2X30A, imagen tomada de: http://www.pololu.com/picture/view/0J3794	23
Figura 9. Diagrama de bloques, imagen hecha en Microsoft Word	26
Figura 10. Sensor ultrasónico SRF02, imagen tomada de http://www.robot-electronics.co.uk/htm/srf02tech.htm	27
Figura 11. Ubicación de los sensores ultrasónicos SRF02.....	28
Figura 12. Vista frontal del transportador para la medición del ángulo máximo de detección del sensor SRF02	29
Figura 13. Procedimiento para la detección del ángulo máximo del sensor SRF02.....	29
Figura 14. Patrón del haz medido para el sensor SRF02	32
Figura 15. Barra de aluminio 5x5x15cm, imagen tomada de SolidWords.....	33
Figura 16. Mouse, tomada de www.geniusnet.com	33
Figura 17. Persona, imagen prediseñada de Word.....	33
Figura 18. Datos medidos con el objeto ubicado a 10 cm del sensor, imagen tomada de Arduino.....	34
Figura 19. Datos medidos con el objeto ubicado a 20 cm del sensor, imagen tomada de Arduino.....	34
Figura 20. Datos medidos con el objeto ubicado a 30 cm del sensor, imagen tomada de Arduino.....	35
Figura 21. Datos medidos con el objeto ubicado a 80 cm del sensor, imagen tomada de Arduino.....	35
Figura 22. Datos medidos con el objeto ubicado a 100 cm del sensor, imagen tomada de Arduino.....	36
Figura 23. Datos medidos con el objeto ubicado a 120 cm del sensor, imagen tomada de Arduino.....	36

Figura 24. Datos medidos con el objeto ubicado a 140cm del sensor, imagen tomada de Arduino.....	36
Figura 25. Datos medidos con la barra de aluminio ubicado a 150cm y un mouse ubicado a 140cm del sensor, imagen tomada de Arduino.....	37
Figura 26. Datos medidos con la barra de aluminio ubicada a 92cm y el mouse ubicado a 82cm del sensor, imagen tomada de Arduino.	38
Figura 27. Ubicación de dos objetos dentro del cono de detección del sensor SRF02.....	38
Figura 28. Medición del sensor SRF02 detectando el mouse a 72cm dentro del cono de detección, imagen tomada de Arduino.	39
Figura 29. Ubicación de un obstáculo a 72cm dentro del cono de detección. ..	39
Figura 30. Obstáculo frente al sensor ultrasónico SRF02 a 350cm de distancia, imagen tomada de Arduino.	39
Figura 31. Conexión del sensor SRF02 en modo I2C, imagen tomada de http://www.robot-electronics.co.uk/htm/srf02techI2C.htm	40
Figura 32. Configuración del DIP Switch en modo RC, imagen tomada de: Datasheet driver RoboClaw 2X30A.	44
Figura 33. Protección de bajo de voltaje, imagen tomada de: Datasheet Driver RoboClaw	45
Figura 34. Estado de los interruptores del DIPSwitch del driver RoboClaw, imagen tomada de datasheet Driver RoboClaw 2x30A, editada en Paint.....	45
Figura 35. Conexiones del driver RoboClaw 2x30A, imagen tomada de datasheet Driver RoboClaw 2x30A, editada en Paint.	46
Figura 36. Ejecución del comando "adelante", imagen hecha en Paint	47
Figura 37. Leds de estado y de error del driver RoboClaw 2x30A, imagen tomada de Datasheet Driver RoboClaw 2x30A.....	49
Figura 38. Conexión entre el EasyVR y un Host.....	50
Figura 39. Conexión del Shield EasyVR y Arduino Mega.	51
Figura 40. EasyVR commander, área de trabajo, imagen tomada de: EasyVR User Manual.....	52
Figura 41. Grabar comandos de voz en EasyVR commander, imagen tomada de EasyVR Commander 3.6	53
Figura 42. Conexión para establecer comunicación serial entre dos dispositivos, imagen hecha en Word.....	58

LISTA DE TABLAS

Tabla 1. Datos medidos por el sensor ultrasónico SRF02	31
Tabla 2. Distancia máxima medida por el sensor	31
Tabla 3. Registros del sensor SRF02.	40
Tabla 4. Comandos del sensor SRF02.	42
Tabla 5. Cambio de la dirección por defecto de la comunicación I2C entre sensores y Arduino.	43
Tabla 6. Señal de PWM para la ejecución de los comandos de voz.....	46
Tabla 7. Especificaciones técnicas del motor	47
Tabla 8. Prueba 1 con el sistema de comandos de voz EasyVR.....	55
Tabla 9. Prueba 2 con el sistema de comandos de voz EasyVR.....	55
Tabla 10. Prueba 3 con el sistema de comandos de voz EasyVR.....	56
Tabla 11. Prueba 4 con el sistema de comandos de voz EasyVR.....	56
Tabla 12. Prueba 5 con el sistema de comandos de voz EasyVR.....	56

RESUMEN GENERAL DE TRABAJO DE GRADO

TÍTULO: Manejo de los motores de una silla de ruedas eléctrica mediante comandos de voz.

AUTORES: Oskar Alonso Vargas Fagua
Ruth Mayorly Sánchez Flores

FACULTAD: Ingeniería Electrónica

DIRECTOR: Ing. Fabio Alonso Guzmán Serna

RESUMEN

En el presente trabajo se desarrolló una alternativa de comandos para la movilidad de personas con dificultades motrices, especialmente aquellas que poseen motricidad muscular baja y movimientos involuntarios en su cuerpo. La solución consiste de un sistema de reconocimiento de comandos por voz para los movimientos de: adelante, atrás, derecha, izquierda, y parar. El sistema está incorporado a una silla de ruedas. Cuenta con un sistema de protección a impactos por obstáculos presentes en la trayectoria de avance de la silla, el cual detiene la misma e informa al usuario el porqué de la acción, a través de un mensaje audible. Como característica adicional posee un mecanismo de arranque y frenado que permite movimientos suaves. El resultado de éste proyecto representa una solución integral a los sistemas de reconocimiento de voz para sillas de ruedas, porque integra diferentes funciones en un solo sistema.

Palabras Clave: Movilidad, sistema de reconocimiento de voz, protección, obstáculos, mensajes audibles.

GENERAL SUMMARY OF WORK OF GRADE

TÍTULO: Management of the engines of an electric wheelchair using voice commands.

AUTHOR'S: Oskar Alonso Vargas Fagua
Ruth Mayorly Sánchez Flores

FACULTY: Ingeniería Electrónica

DIRECTOR: Ing. Fabio Alonso Guzmán Serna

ABSTRACT

Within this research project an alternative to voice-commands for people with reduced motoric function was developed, especially for those who possess low muscular motoric function and involuntary body movements. The proposed solution is based on a voice commands' recognition system embedded in a wheelchair. The system handles the commands: forwards, backwards, left, right and stop. The system has an anti-obstacles impact protection feature, which stops the wheelchair and informs the user through audible messages. Additionally, an increasing a decreasing speed control mechanism enables smooth movements. The result of this project represents an integral solution to voice recognition systems for wheelchairs, because it integrates different functions into a unique system.

Keywords: reduced mobility, voice recognition systems, protection, obstacles, audible messages.

1. INTRODUCCIÓN

En este documento usted encontrará el procedimiento que se llevó a cabo para lograr el desarrollo del proyecto de grado “manejo de los motores de una silla de ruedas eléctrica mediante comandos de voz”, garantizando que el arranque y frenado del sistema sea suave y contando con la opción del aumento de la velocidad de sus motores en espacios abiertos, de esta forma dar continuidad a los proyectos “Automatización de una Silla de Ruedas Controlada por Biopotenciales” y “manejo de una silla de ruedas eléctrica para personas cuadripléjicas controlada mediante movimientos del cuello y con sistema de protección de impactos” realizados en la Universidad Pontificia Bolivariana.

Partiendo de los objetivos planteados, se consignó un marco teórico en donde se describen los aspectos y problemas más comunes que presenta una persona con discapacidad motriz y un abreboca de los elementos electrónicos y software usados para lograr alcanzar esta meta.

El manejo de los motores de la silla de ruedas dispone de un sistema de reconocimiento de voz llamado EasyVR, y está equipada con 7 sensores SRF02 para medir y procesar la información del entorno y así habilitar o no la posibilidad de aumentar la velocidad, todo este proceso está descrito por partes en el título cinco, donde se detallan las características de los elementos electrónicos implementados, como sensores, microcontroladores y todo el proceso que se siguió para unificar el sistema e implementarlo en la silla de ruedas disponible en la Universidad Pontificia Bolivariana, Seccional Bucaramanga.

2. OBJETIVOS

2.1. Objetivo general

Diseñar e implementar el control de la operación de los motores de una silla de ruedas eléctrica para tetraplégicos, mediante comandos de voz.

2.2. Objetivos específicos

- Seleccionar un sistema comercial de reconocimiento de voz.
- Programar y acoplar el sistema de reconocimiento de voz con el sistema Arduino, encargado del control de la operación de los motores de la silla de ruedas eléctrica.
- Implementar un arranque y frenado suave en la silla de ruedas eléctrica disponible en la Universidad Pontificia Bolivariana, seccional Bucaramanga, para mejorar la seguridad y el confort del usuario.
- Acoplar un sistema que varíe y limite la velocidad de la silla de ruedas de acuerdo a los datos medidos por los sensores de proximidad ubicados en ella, y así mejorar la operación de la misma.
- Realizar pruebas de manejo de la silla de ruedas con la ayuda de una persona discapacitada, que permita hacer retroalimentación para corregir posibles errores o defectos de la interfaz de comandos de voz.

3. MARCO TEÓRICO

3.1. DISCAPACIDAD MOTRIZ

Las personas con discapacidad motora presentan características clínicas y funcionales diversas. Su discapacidad puede ser de origen congénito como la parálisis cerebral o la espina bífida, por causas adquiridas como los traumatismos craneales, por enfermedades degenerativas como algunas distrofias musculares, y su prevalencia puede ser permanente o temporal. La mayoría de estas afectaciones dificultan o imposibilitan la movilidad funcional de una o varias partes del cuerpo. Esta disfunción orgánica da lugar a clasificaciones como hemiplejía, paraplejía, diplejía, tetraplejía, etc., y a diferentes grados de disfunción muscular que pueden producir espasticidad, atetosis, ataxia, etc. En algunos casos la discapacidad conlleva otros trastornos asociados como son dificultades visuales, auditivas, neuropsicológicas, cognitivas, entre otras. [1]

Los cambios en los movimientos se clasifican con base en las áreas del cuerpo afectadas:

- a) Monoplejía. Un solo miembro del cuerpo.
- b) Diplejía. Afecta las extremidades inferiores (piernas).
- c) Triplejía. Afecta un miembro superior (un brazo) y las extremidades inferiores (piernas).
- d) Hemiplejía. Afecta el lado derecho o izquierdo del cuerpo.
- e) Cuadriplejía. Afecta las cuatro extremidades del cuerpo (brazos y piernas).

En la discapacidad motriz intervienen las funciones cerebrales que envían información al cuerpo a través de los nervios, sobre cómo moverse y por qué es necesaria la interpretación adecuada de las sensaciones que llegan a él. El cerebro cumple con tres funciones principales:

- a) *Sensitiva*. El cerebro es capaz de sentir determinados cambios o estímulos del interior del organismo, es decir, el medio interno (como un dolor de estómago); también fuera de él, es decir, el medio externo (por ejemplo, una gota de lluvia que cae en la mano o el perfume de una rosa).
- b) *Integradora*. La información sensitiva se analiza, se almacenan algunos aspectos y se toman decisiones de la conducta a seguir.

c) *Motora*. Es la respuesta que el cerebro organiza a partir de los estímulos; por ejemplo, contracciones musculares, movimientos o secreciones glandulares como la salivación al ver un alimento.

El sistema nervioso está compuesto por:

- a) *Neuronas*. Células que forman el sistema nervioso central. Se encuentran en el cerebro y en la médula espinal. Su principal tarea es recibir y enviar información sobre el cuerpo y ambiente, y cómo producir y dirigir las acciones y pensamientos.
- b) *Médula espinal*. Además de neuronas, contiene terminaciones nerviosas que llevan la información de los sentidos al cerebro; otras llevan de regreso la información motora a músculos y órganos. Observar la siguiente figura.

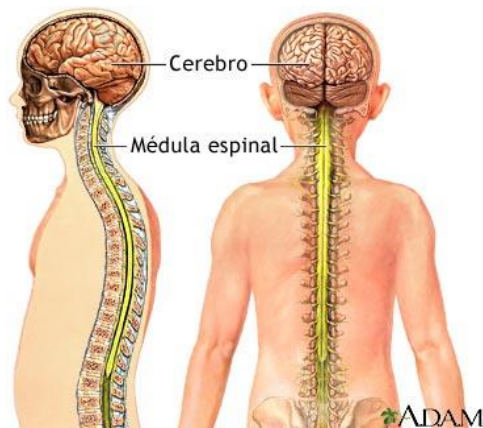


Figura 1. Médula espinal, imagen tomada de ADAM.

- c) *Tallo cerebral*. Aquí llegan todas las terminaciones nerviosas de la médula espinal. Interpreta las sensaciones de todas las funciones que no se controlan de forma consciente y voluntaria (como la respiración durante el sueño y el latido del corazón). También organiza y relaciona las actividades de cada uno de los lados del cerebro para funcionar armónicamente.
- d) *Cerebelo*. Procesa todo tipo de sensaciones, pero en especial organiza las de gravedad, movimiento, sistema muscular y articulaciones, para que el cuerpo se mueva suavemente y con precisión. Observar la Figura 2.



Figura 2. El cerebelo, imagen tomada de: ADAM.

- e) *Hemisferios cerebrales*. Procesan la información sensorial, incluyendo su significado. También planean y desarrollan la respuesta que dará el cuerpo. Dentro de los hemisferios cerebrales se encuentra el sistema límbico, responsable de regular el crecimiento y la respuesta emocional.



Figura 3. Hemisferios cerebrales, imagen tomada de: Discapacidad motriz, Guía didáctica para la inclusión en educación inicial y básica, pag. 17.

- f) *Corteza cerebral*. Existen áreas para la interpretación de los sonidos, de las imágenes y para el control voluntario del cuerpo. Significa que una parte específica del cerebro mueve cada parte del cuerpo (por ejemplo, las áreas para los dedos, las manos y los músculos del habla son más grandes que las secciones para el resto del cuerpo). Para mover las manos y la boca se establecen movimientos dirigidos, precisos y complejos como escribir y

hablar, respectivamente. La corteza cerebral también recibe información de otras áreas; por ejemplo, el área de la visión toma la información sonora, del movimiento o del tacto, para darle un significado a la imagen que percibe a través de las “áreas de asociación”.

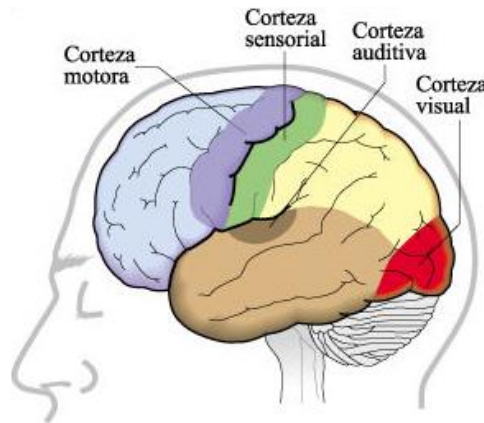


Figura 4. La corteza cerebral, imagen tomada de: Biología de los animales, Capítulo 49. Integración y control. IV: el cerebro de los vertebrados, corteza cerebral.

Toda la piel y las articulaciones actúan como receptores de sentidos que envían, a través de los nervios, la información de todas las sensaciones al cerebro. A su vez, éste produce conciencia, percepción y conocimiento; también origina posturas, movimientos, planeación y coordinación de movimientos, emociones, pensamientos, memoria y aprendizaje. Las sensaciones y los impulsos motores mantienen la postura, el balance y el movimiento. Sin embargo, en su desarrollo, los bebés tienen que aprender a dar significado a las sensaciones y a moverse. Todos los aprendizajes se fortalecen aproximadamente hasta los siete años; además, a esta edad, los niños son más activos y aprenden a hacer muchas cosas con su cuerpo. De igual modo, muestran respuestas adecuadas que les permiten aumentar su capacidad de integrar sentidos para perfeccionar el uso de herramientas, usar cubiertos, vestirse y escribir.

Se produce una discapacidad motriz, cuando se presenta un cambio en la información que el cerebro envía a las extremidades o cuando hay un problema en los huesos, en los músculos y en las articulaciones, constituye una alteración de la capacidad del movimiento que afecta, en distinto nivel, las funciones de desplazamiento, manipulación o respiración, y que limita a la persona en su desarrollo personal y social. La discapacidad motriz se clasifica en los siguientes trastornos:

1. *Trastornos físicos periféricos.* Afectan huesos, articulaciones, extremidades y músculos. Se presentan desde el nacimiento (por ejemplo, algunas malformaciones de los huesos), o bien, son consecuencias de enfermedades en la infancia (como la tuberculosis ósea articular). Algunos accidentes o lesiones en la espalda dañan la médula espinal e interrumpen la comunicación de las extremidades (brazos y piernas) hacia el cerebro y viceversa.
2. *Trastornos neurológicos.* Significan el daño originado en el área del cerebro (corteza motora cerebral) encargada de procesar y enviar la información de movimiento al resto del cuerpo. Origina dificultades en el movimiento, y en el uso, sensaciones y control de ciertas partes del cuerpo. Los más comunes son la parálisis cerebral, los traumatismos craneoencefálicos y los tumores localizados en el cerebro. [2].

3.2. ARDUINO

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. El microcontrolador en la tarjeta Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring¹) y el entorno de desarrollo Arduino (basado en Processing²), un entorno de programación multimedia

Las tarjetas pueden ser hechas a mano o compradas ensambladas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia están disponibles bajo una licencia abierta.

Arduino, además de simplificar el proceso de trabajar con microcontroladores, ofrece algunas ventajas respecto a otros sistemas:

Asequible. Las tarjetas Arduino son más asequibles comparadas con otras plataformas de microcontroladores.

Multi-Plataforma. El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux.

Entorno de programación simple y directo. El entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados.

Software ampliable y de código abierto. El software Arduino está publicado bajo una licencia libre y preparada para ser ampliada por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y si

¹ Marco de programación de código abierto para microcontroladores.

² Lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java.

se está interesado en profundizar en los detalles técnicos, se puede programar en el lenguaje AVR C, en el que está basado.

Hardware ampliable y de Código abierto. Arduino está basado en los microcontroladores ATMEGA168, ATMEGA328 y ATMEGA1280. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. [3]. En la siguiente figura se muestra la tarjeta de desarrollo Arduino que se usará para el desarrollo del proyecto.



Figura 5. Arduino MEGA, imagen tomada de: www.arduino.cc

3.3. MÓDULO DE RECONOCIMIENTO DE VOZ Y REPRODUCCIÓN, EASYVR

EasyVR es un módulo pequeño, económico y de manejo amigable. Dispone de 26 comandos o voces pregrabadas que se pueden dictar en 6 idiomas diferentes para su reconocimiento: inglés, italiano, japonés, alemán, español y francés. Tiene capacidad de memorizar, para su posterior reconocimiento, hasta 32 voces definidas por el usuario en cualquier idioma. Cada voz se asocia también a un comando o etiqueta que es devuelto cada vez que es reconocida.

EasyVR también es capaz de reproducir los sonidos de una lista previamente grabada en su memoria flash interna. Dichos sonidos pueden ser de diferente tipo (música, señales acústicas, voces habladas, etc...) que se pueden obtener a partir de archivos tipo WAV. Cuenta con salida de audio PWM que admite la conexión con un altavoz de 8 ohmios, con capacidad de reproducción de hasta 9 minutos.

El módulo dispone de un software de desarrollo, el EasyVR Commander para Windows, que permite una rápida familiarización y empleo de todas sus funciones. También se puede conectar, vía UART³, con cualquier tipo de microcontrolador (PIC, BASIC STAMP, ATMEL, ARDUINO, etc...) desde donde se pueden controlar todas las funciones de edición, enseñanza, reconocimiento de voces y reproducción de sonido, lo que permite su empleo en cualquier aplicación o proyecto del usuario. [4]

A continuación se muestra el módulo EasyVR:



Figura 6. Módulo EasyVR, imagen tomada de: guía rápida del módulo de reconocimiento de voz y reproducción, EasyVR.

En el mercado existen otros módulos de reconocimiento de voz como el VRbot, el módulo SmartVR y el "MPLAB Starter Kit for dsPIC DSC", pero teniendo en cuenta que el proyecto es desarrollado con el fin de brindar una alternativa eficiente y al alcance de todos, se seleccionó el Shield EasyVR para arduino, el cual tiene un precio bastante asequible además de que en la misma tarjeta se tienen todos los componentes para crear un sistema de reconocimiento y reproducción de voz: micrófono, salida de audio y chip de reconocimiento de voz.

³ UART: son las siglas de "Universal Asynchronous Receiver-Transmitter". Éste controla los puertos y dispositivos serie. Se encuentra integrado en la tarjeta base o en la tarjeta adaptadora del dispositivo. Tomado de: <http://es.wikipedia.org/wiki/UART>.

3.4. SENSORES DE PROXIMIDAD

El sensor de proximidad es un transductor que detecta objetos o señales que se encuentran cerca o próximos al mismo. Existen varios tipos de sensores de proximidad según el principio físico que utiliza. Entre los cuales se tienen:

Sensor capacitivo: su función está en señalar un cambio de estado, basado en la variación del estímulo de un campo eléctrico, estos miden cambios en la capacitancia, la cual depende de la constante dieléctrica del material a detectar, su masa, tamaño y distancia hasta la superficie sensible del sensor.

Sensor inductivo: los sensores inductivos de proximidad han sido diseñados para trabajar generando un campo magnético y detectando las pérdidas de corriente de dicho campo, generadas al introducirse en él los objetos de detección férricos y no férricos. Consiste en una bobina con núcleo de ferrita, un oscilador, un sensor de nivel de disparo de la señal y un circuito de salida. El funcionamiento es similar al sensor capacitivo; la bobina detecta el objeto cuando se produce un cambio en el campo electromagnético y envía la señal al oscilador, luego se activa el disparador y finalmente al circuito de salida hace la transición entre abierto o cerrado.

Sensor ultrasónico: tienen como función principal la detección de objetos a través de la emisión y reflexión de ondas acústicas. Funcionan emitiendo un pulso ultrasónico contra el objeto a sensor, y al detectar el pulso reflejado, se para un contador de tiempo que inició su conteo al emitir el pulso. Este tiempo es referido a distancia y de acuerdo con los parámetros elegidos de respuesta, con ello envía una señal eléctrica digital o analógica.

Además de los anteriores, existen otros sensores de proximidad en el mercado, que se usan de acuerdo a las necesidades y requisitos que se tengan. [5].

3.5. DRIVER ROBOCLAW 2X30A

Es una tarjeta muy eficiente, versátil y diseñada para el control de 2 canales síncronos y regenerativos, cada canal maneja una corriente nominal de 30 amperios y una corriente pico hasta de 60 amperios. Cuenta con 4 posibles modos de configuración, cada uno cuenta con una forma específica de controlar el driver, estos se configuran a través del DIP Switch mostrado en la Figura 7.

Con éste se ajusta el modo de funcionamiento y las muchas opciones del driver, el DIP switch está referido con una etiqueta ON en la parte superior. Los switches están etiquetados de izquierda a derecha empezando por el SW1 y

terminando con el SW10, cuando un switch se mueve hacia la etiqueta ON se considera activo, de lo contrario se considera apagado. Observar que en la siguiente figura sólo el SW1 se encuentra activo:

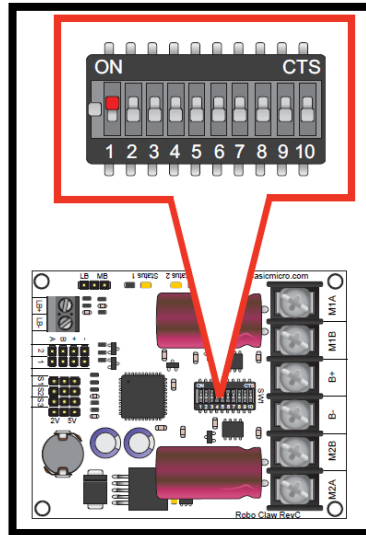


Figura 7. DIP Switch. Imagen tomada de: Datasheet driver RoboClaw 2X30A.

El driver permite la entrada de 2 encoders (codificador rotatorio) en cuadratura y está equipado con un modo síncrono de frenado regenerativo, por lo cual se puede recargar la batería en desaceleración, frenada o inversión de giro, prolongando de esta manera las horas de uso del dispositivo controlado. En la Figura 8 se muestra la tarjeta de control RoboClaw 2X30A. [6]



Figura 8. Tarjeta controladora RoboClaw 2X30A, imagen tomada de: <http://www.pololu.com/picture/view/0J3794>

A continuación se expone cada modo de configuración del RoboClaw con sus aplicaciones ideales:

3.5.1. Modo 1 – Entrada RC

En este caso el RoboClaw puede ser controlado desde algún sistema de control de radio o mediante un microcontrolador. El driver espera pulsos de entrada para controlar la velocidad y dirección del motor, muy similar a como se controla un servomotor regular. Este modo no permite el manejo de encoders.

3.5.2. Modo 2 - Análogo

Usa una señal análoga de 0V a 5V para controlar la velocidad y dirección de cada motor. RoboClaw puede ser controlado usando un potenciómetro o una señal PWM proveniente de un microcontrolador. El modo análogo es ideal para la interconexión de sistemas de posicionamiento con joystick u otro que no sea la interconexión de un microcontrolador. En este modo no se pueden implementar encoders.

3.5.3. Modo 3 - Serial

En este modo el RoboClaw espera una señal de nivel TTL a través de la interfaz RS-232 para controlar serialmente la dirección y velocidad de cada motor. Es usado típicamente para ser controlado a través de un microcontrolador o PC. Si se usa un PC se debe usar un tipo de circuito MAX232 desde el RoboClaw que trabaja solamente con entradas de nivel TTL. Este modo permite incluir un modo esclavo que permite conectar múltiples RoboClaws para ser controlados con una señal a través del puerto RS-232 con PC o microcontrolador.

3.5.4. Modo 4 – Paquetes seriales

En este modo el RoboClaw espera una señal de nivel TTL a través de la interfaz RS-232 para controlar serialmente la dirección y velocidad de cada motor. Es usado típicamente para ser controlado a través de un microcontrolador o PC. Si se usa un PC se debe usar un tipo de circuito MAX232 desde el RoboClaw que trabaja solamente con entradas de nivel TTL. En el modo de paquetes seriales a cada RoboClaw se le asigna una dirección usando el DIP switch. Hay 8 direcciones disponibles, lo cual significa que pueden estar activos 8 RoboClaws al mismo puerto serial. Cuando se use este modo se requiere usar decodificación por cuadratura, debido a que es un formato de comunicación bidireccional. Esto permite transmitir información de la posición y velocidad de los encoders.

4. DESARROLLO

Para el desarrollo de este proyecto se usó un Shield⁴ de reconocimiento de voz para Arduino llamado EasyVR del fabricante VEEAR⁵. Este módulo se acopló directamente a una tarjeta de Arduino Mega, en donde se programó únicamente la parte de los comandos de voz, para lo cual fue necesario establecer comunicación serial con el software EasyVR commander, encargado de la edición de las voces.

Seguidamente, para evitar conflictos en el sistema, se vinculó el Arduino Mega con una tarjeta Antduino (versión copia del Arduino uno) a través de comunicación serial, la cual se encargó de dar funcionamiento al driver RoboClaw 2x30A para que accione los motores de la silla de ruedas a través de señal PWM de acuerdo a los comandos de voz dictados; además de la supervisión de los sensores SRF02 detectores de obstáculos, mediante comunicación I2C, los cuales sensan el espacio en el que se encuentra la silla y partiendo de ello se toman las medidas necesarias para mantener la seguridad del usuario.

Observar el diagrama de bloques que se muestra en la Figura 9.

⁴ Las "Shields" son tarjetas de desarrollo que pueden ser conectadas encima de la tarjeta Arduino extendiendo sus capacidades. Definición tomada de: <http://arduino.cc>

⁵ <http://www.veear.eu/>

4.1. DIAGRAMA DE BLOQUES

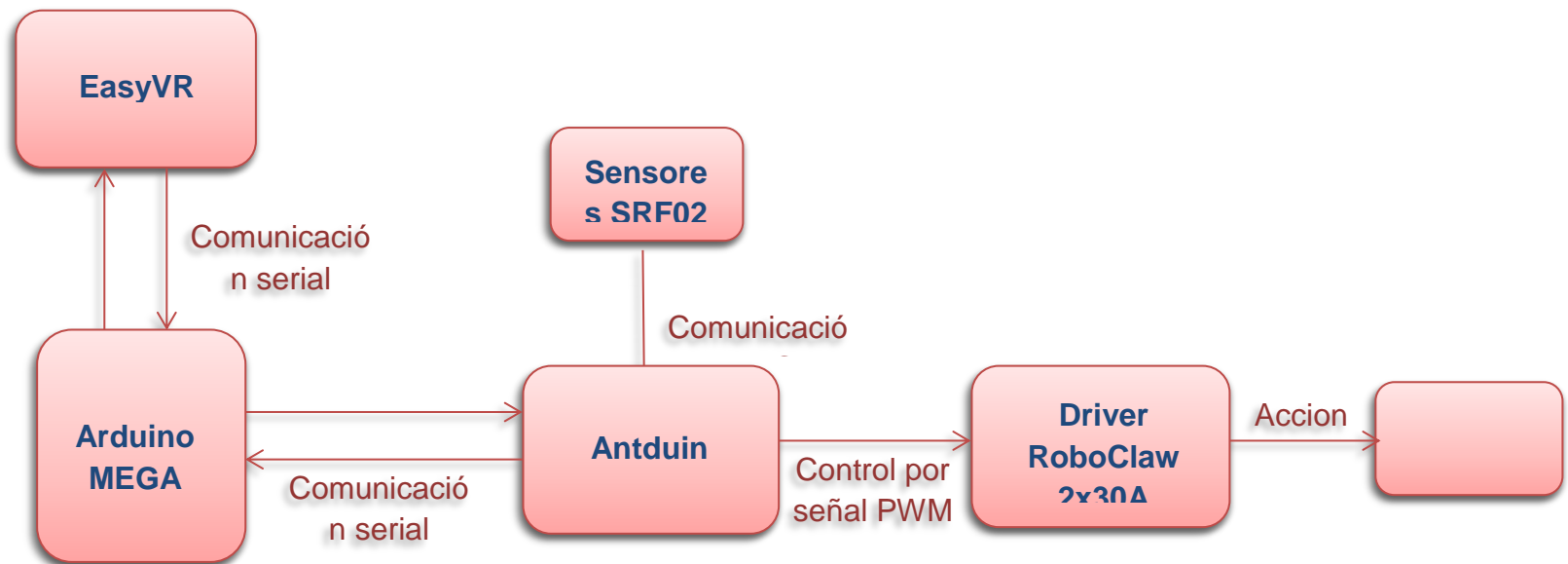


Figura 9. Diagrama de bloques, imagen hecha en Microsoft Word

4.2. SRF02, SENSOR ULTRASÓNICO

El sensor de distancias por ultrasonidos SRF02, mostrado en la Figura 10, es un sensor de pequeño tamaño y mínimo consumo, que permite establecer comunicación serial o comunicación I2C. La comunicación serial tiene un formato estándar de 9600 baudios, un bit de comienzo, 8 bits de datos y un bit de parada, con un nivel de tensión TTL.

En un mismo bus de datos se pueden conectar hasta 16 sensores SRF02, ya sea por comunicación serial o I2C.

Este sensor utiliza sólo un transductor para transmitir la ráfaga ultrasónica y recibir el eco de la misma, permitiendo esto medir la distancia a la que se encuentra el objeto detectado. La distancia mínima que es capaz de medir es alrededor de 15cm (6"). El SRF02 puede entregar medidas en microsegundos, centímetros o pulgadas.

Para elegir el modo de operación del SRF02 se usa el pin "*mode*", el cual al conectarlo a tierra hace que el sensor trabaje con comunicación serial y si se quiere modo I2C hay que dejarlo desconectado (o puede conectarse a +5V). Para el desarrollo de este proyecto se implementó el modo I2C. [7]

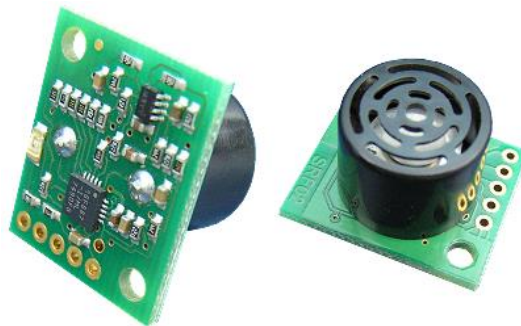


Figura 10. Sensor ultrasónico SRF02, imagen tomada de <http://www.robot-electronics.co.uk/htm/srf02tech.htm>

4.2.1. Ubicación de los sensores ultrasónicos en la silla de ruedas

En la siguiente figura se muestra la manera en que se distribuyeron 7 sensores ultrasónicos SRF02, dispuestos para el sensado del entorno en que se encuentra la silla de ruedas y de acuerdo a esto manipular el driver RoboClaw 2x30A.



Figura 11. Ubicación de los sensores ultrasónicos SRF02

4.2.2. Ancho del haz ultrasónico

Se realizaron unas mediciones con ayuda del transportador mostrado en la Figura 12, donde se ubicó un sensor ultrasónico SRF02 el cual se iba girando cada cinco o diez grados, hasta que no detectara el obstáculo que se ubicó frente a él a 350cm de distancia, esto con el fin de conocer el ancho del haz del sensor.

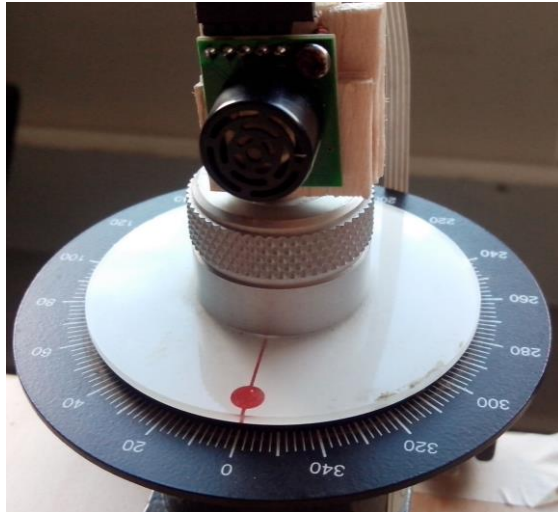


Figura 12. Vista frontal del transportador para la medición del ángulo máximo de detección del sensor SRF02

El objeto con el que se le realizó la prueba de detección al sensor SRF02 fue una barra metálica (remitirse a la Figura 15, pág. 33), la cual se ubicó frente al sensor inicialmente sin ningún ángulo de giro, observar la Figura 13.



Figura 13. Procedimiento para la detección del ángulo máximo del sensor SRF02

Cabe resaltar que la distancia máxima medida por el sensor se limitó por programación a 350 cm, por lo tanto la medida del sensor a 0° fue 350cm. En la Tabla 1 se muestran datos medidos por el sensor a un objeto ubicado a 5, 10, 15, 20 y 30 grados hacia la derecha, inicialmente a una distancia de 30 cm hasta 350cm:

30°	20 °	15°	10°	5°
36	33	32	26	32
33	33	32	27	32
38	33	33	26	31
37	33	33	25	31
fuera4	33	52	29	50
53	fuera4	53	29	50
53	52	52	29	50
53	52	52	32	50
53	52	53	42	50
fuera4	52	72	41	72
74	52	72	41	72
73	98	72	41	72
73	102	72	41	72
84	102	72	40	89
83	105	72	80	90
83	102	72	80	90
84	101	91	80	90
84	101	91	fuera4	109
84	fuera4	92	96	109
89	fuera4	93	102	109
104	151	93	120	111
103	153	fuera4	120	fuera4
102	150	93	120	130
102	152	110	123	130
102	152	112	138	130
102	153	113	139	129
fuera4	153	fuera4	139	149
fuera4	153	133	139	149
fuera4	153	133	139	149
fuera4	152	133	160	181
fuera4	181	133	159	180
	199	136	159	207
	200	139	fuera4	208
	200	150	179	208
	200	148	178	206

	200	149	178	212
	200	149	178	220
	200	149	179	222
	248	188	178	227
	248	188	200	268
	249	189	199	269
	248	188	197	272
	248	189	198	290
	248	232	219	288
	249	248	fuera4	287
	248	248	219	fuera4
	298	248	219	fuera4
	298	249	219	307
	298	249	219	310
	298	248	220	310
	298	248	300	308
	298	248	301	308
	349	249	301	310
	350	348	346	329
	350	348	347	329
	349	348	346	328
	350	348	347	348
	349	348	346	346
	fuera4	fuera4	fuera4	fuera4
	fuera4	fuera4	fuera4	fuera4
	fuera4	fuera4	fuera4	fuera4
	fuera4	fuera4	fuera4	fuera4

Tabla 1. Datos medidos por el sensor ultrasónico SRF02

De acuerdo a los datos, el punto máximo de distancia medida para cada uno de los anteriores casos está mostrado en la Tabla 2:

Grados	Punto máximo
5°	350
10°	346
15°	348
20°	349
30°	102

Tabla 2. Distancia máxima medida por el sensor

En las Tabla 1 y Tabla 2 no se consignó la medida con el sensor a 0° debido a que en ese caso la medida de distancia será más clara pues el haz ultrasónico apuntará directamente al objeto, con lo cual se tendrá como distancia máxima 350cm. Tampoco se hicieron mediciones con el sensor girado a la izquierda porque es el espejo de las medidas hechas hacia la derecha. A continuación se muestra el patrón del haz medido para el sensor SRF02 utilizando un objeto metálico.

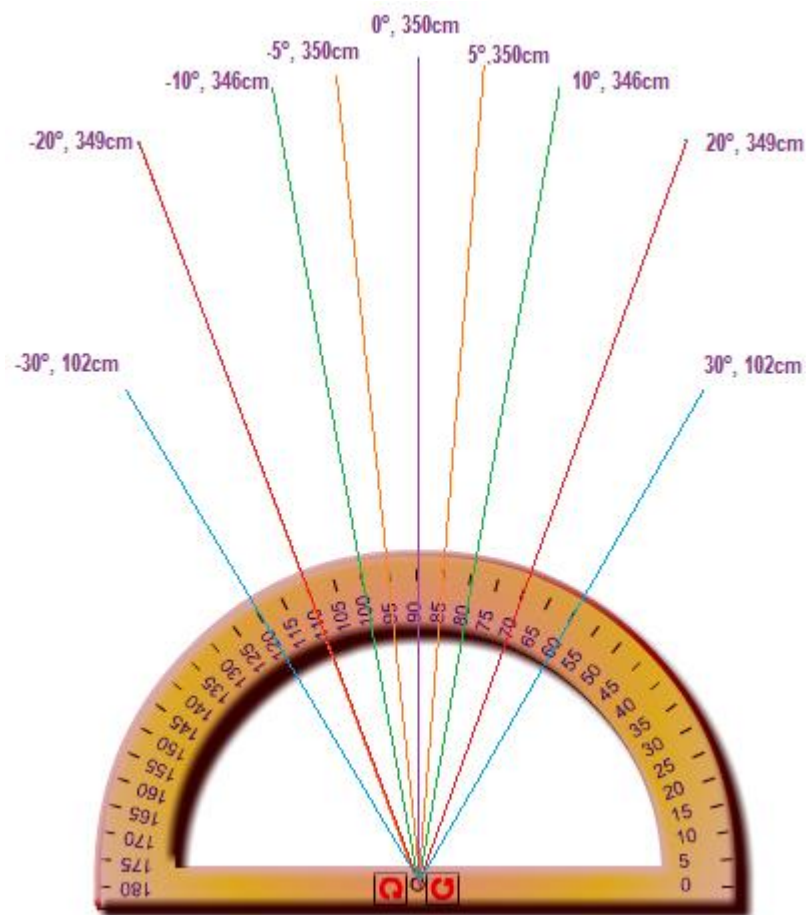


Figura 14. Patrón del haz medido para el sensor SRF02

Partiendo de la Figura 14, se dibujó el patrón del ancho del haz hallado y se le realizaron pruebas al sensor SRF02, lo cual se expone en el siguiente punto.

4.2.3. Calibración del sensor SRF02

Se utilizaron los obstáculos mostrados en las Figura 15, Figura 16 y Figura 17 para detectar la precisión del sensor SRF02:

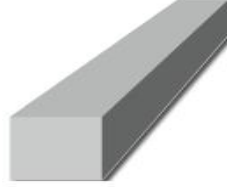


Figura 15. Barra de aluminio 5x5x15cm, imagen tomada de SolidWords.



Figura 16. Mouse, tomada de www.geniusnet.com.



Figura 17. Persona, imagen prediseñada de Word.

Se ubicó el obstáculo mostrado en la Figura 15 frente al sensor SRF02 y se tomaron las siguientes mediciones:

- Con el objeto a 10cm del sensor se observó que registra 17cm, con un porcentaje de error del 70%

$$\%e = \frac{VT - VE}{VT}$$

$$\%e = \frac{10 - 17}{10} \times 100 = 70\%$$

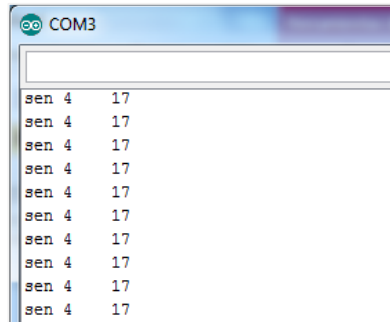


Figura 18. Datos medidos con el objeto ubicado a 10 cm del sensor, imagen tomada de Arduino.

- Con el mismo objeto a 20cm el porcentaje de error disminuyó considerablemente al 10%.

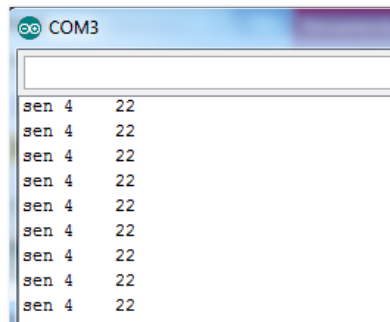
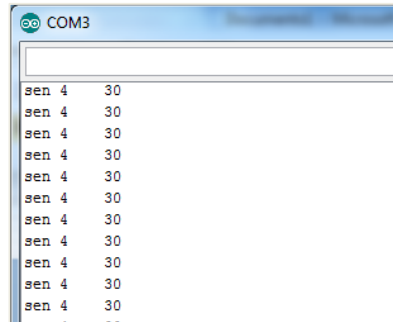


Figura 19. Datos medidos con el objeto ubicado a 20 cm del sensor, imagen tomada de Arduino.

$$\%e = \frac{20 - 22}{20} \times 100 = 10\%$$

- Con el mismo objeto a 30cm el porcentaje de error fue del 0%

$$\%e = \frac{30 - 30}{30} \times 100 = 0\%$$

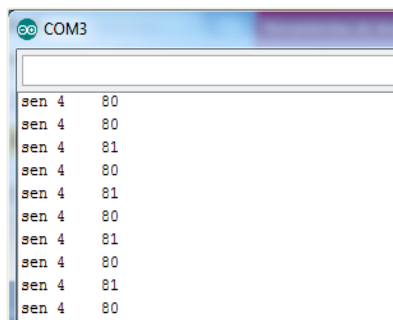


sen 4	30
sen 4	30
sen 4	30
sen 4	30
sen 4	30
sen 4	30
sen 4	30
sen 4	30
sen 4	30
sen 4	30
sen 4	30
sen 4	30

Figura 20. Datos medidos con el objeto ubicado a 30 cm del sensor, imagen tomada de Arduino.

- ▶ Con el objeto ubicado a 80cm se obtuvo las siguientes medidas, dando un promedio de 80,4cm. El porcentaje de error fue del 0,5%

$$\%e = \frac{80 - 80,4}{80} \times 100 = 0.5\%$$

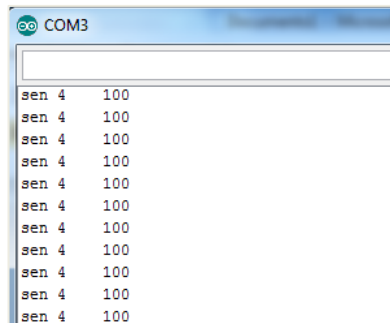


sen 4	80
sen 4	80
sen 4	81
sen 4	80
sen 4	81
sen 4	80
sen 4	81
sen 4	80
sen 4	81
sen 4	80
sen 4	81
sen 4	80

Figura 21. Datos medidos con el objeto ubicado a 80 cm del sensor, imagen tomada de Arduino.

- ▶ Con el objeto ubicado a 100cm del sensor ultrasónico. A continuación se muestran las medidas tomadas y el porcentaje de error calculado.

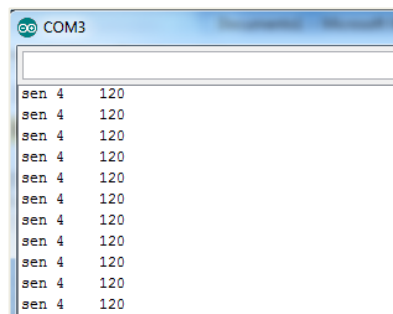
$$\%e = \frac{100 - 100}{100} \times 100 = 0\%$$



sen 4	100
sen 4	100
sen 4	100
sen 4	100
sen 4	100
sen 4	100
sen 4	100
sen 4	100
sen 4	100
sen 4	100
sen 4	100
sen 4	100

Figura 22. Datos medidos con el objeto ubicado a 100 cm del sensor, imagen tomada de Arduino.

- Con el objeto ubicado a 120cm del sensor ultrasónico. A continuación se muestran las medidas tomadas y el porcentaje de error calculado.

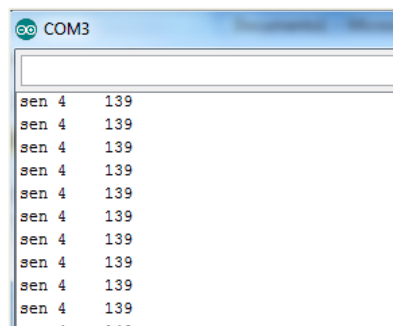


sen 4	120
sen 4	120
sen 4	120
sen 4	120
sen 4	120
sen 4	120
sen 4	120
sen 4	120
sen 4	120
sen 4	120
sen 4	120
sen 4	120

Figura 23. Datos medidos con el objeto ubicado a 120 cm del sensor, imagen tomada de Arduino.

$$\%e = \frac{120 - 120}{120} \times 100 = 0\%$$

- Con el objeto ubicado a 140cm del sensor ultrasónico. A continuación se muestran las medidas tomadas y el porcentaje de error calculado.



sen 4	139
sen 4	139
sen 4	139
sen 4	139
sen 4	139
sen 4	139
sen 4	139
sen 4	139
sen 4	139
sen 4	139
sen 4	139
sen 4	139

Figura 24. Datos medidos con el objeto ubicado a 140cm del sensor, imagen tomada de Arduino.

$$\%e = \frac{140 - 139}{140} \times 100 = 0,7\%$$

Seguidamente se procedió a realizar mediciones con dos objetos ubicados dentro del patrón del haz ultrasónico del sensor SRF02.

- Con el objeto mostrado en la Figura 16 a 140cm y el objeto mostrado en la Figura 15 a 150cm, el sistema del sensor ultrasónico detectó el objeto más cercano, a continuación se muestran las mediciones y el porcentaje de error:

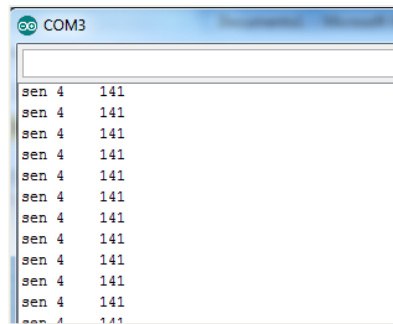
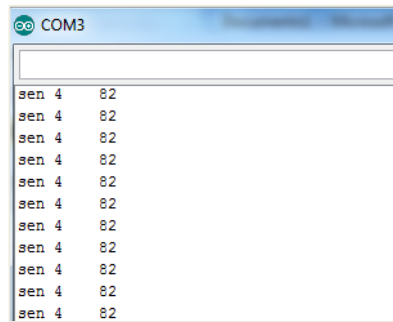


Figura 25. Datos medidos con la barra de aluminio ubicado a 150cm y un mouse ubicado a 140cm del sensor, imagen tomada de Arduino.

$$\%e = \frac{140 - 141}{140} \times 100 = 0,7\%$$

- Con el objeto mostrado en la Figura 16 a 82cm y el objeto mostrado en la Figura 15 a 92cm como se muestra en la Figura 27, el sistema del sensor ultrasónico midió lo siguiente:



COM3	
sen 4	82
sen 4	82
sen 4	82
sen 4	82
sen 4	82
sen 4	82
sen 4	82
sen 4	82
sen 4	82
sen 4	82
sen 4	82
sen 4	82

Figura 26. Datos medidos con la barra de aluminio ubicada a 92cm y el mouse ubicado a 82cm del sensor, imagen tomada de Arduino.

De acuerdo a la medición mostrada en la Figura 26, el porcentaje de error es cero.

$$\%e = \frac{82 - 82}{82} \times 100 = 0\%$$

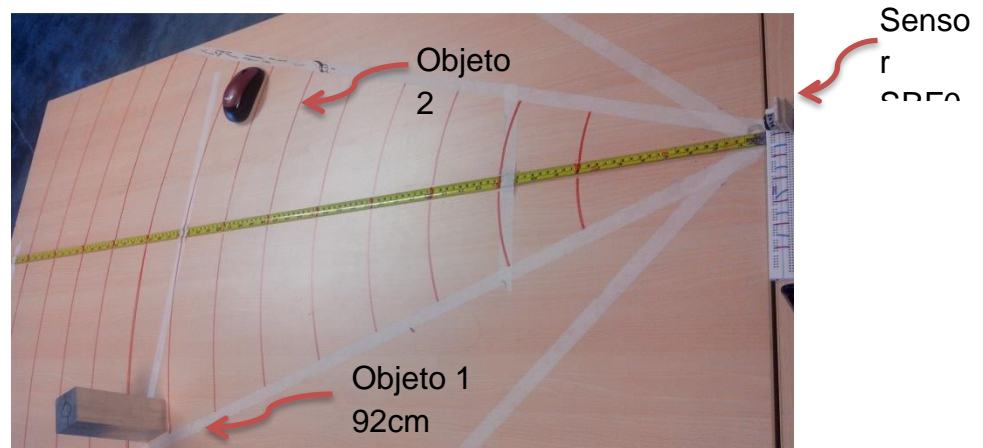


Figura 27. Ubicación de dos objetos dentro del cono de detección del sensor SRF02.

- Con el objeto mostrado en la Figura 16 a 72cm, ubicado como se muestra en la Figura 29, el sistema del sensor realizó las siguientes mediciones:

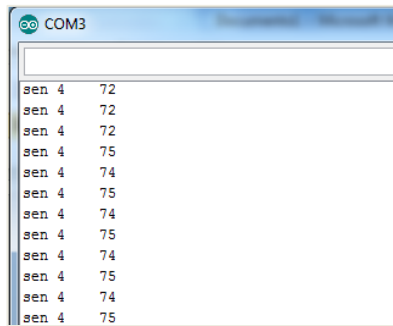


Figura 28. Medición del sensor SRF02 detectando el mouse a 72cm dentro del cono de detección, imagen tomada de Arduino.

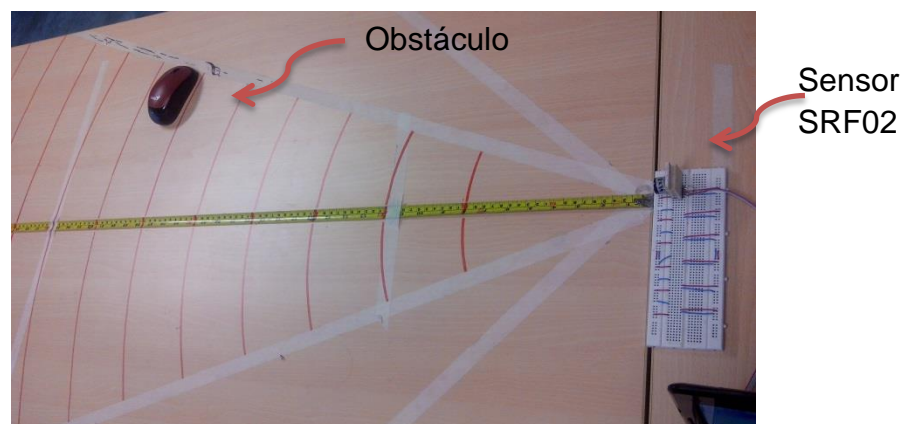


Figura 29. Ubicación de un obstáculo a 72cm dentro del cono de detección.

- A continuación se muestran unas mediciones con el obstáculo mostrado en la Figura 17, el cual es una persona, parada frente al sensor a 3.5m.

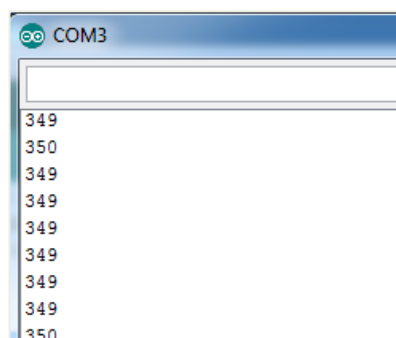


Figura 30. Obstáculo frente al sensor ultrasónico SRF02 a 350cm de distancia, imagen tomada de Arduino.

El porcentaje de error en las mediciones tomadas por el sistema del sensor SRF02 es bastante bajo, a continuación se calcula:

$$\%e = \frac{350 - 349,22}{350} \times 100 = 0,22\%$$

4.2.4. Comunicación I2C

Para seleccionar el modo de comunicación I2C se debe dejar sin conectar el Pin “Mode” del sensor SRF02. El pin “SDA” corresponde a la señal de datos y el pin “SCL” a la señal de reloj. Ambas señales se deben polarizar a +5Vcc a través de dos resistencias de polarización positiva que normalmente se encuentran en el circuito maestro del bus I2C. Observar la Figura 31.

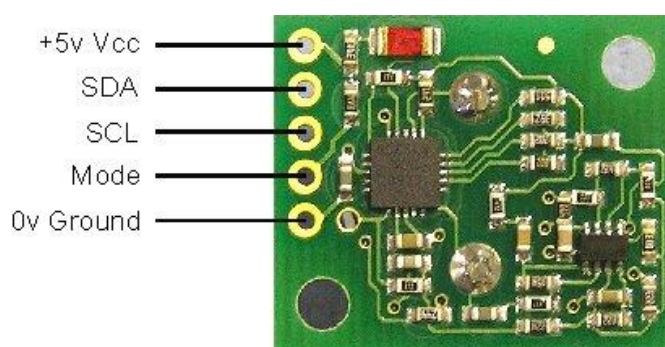


Figura 31. Conexión del sensor SRF02 en modo I2C, imagen tomada de <http://www.robot-electronics.co.uk/htm/srf02techI2C.htm>

La dirección I2C del sensor SRF02 por defecto es 0xE0 pero se puede elegir cualquiera de las siguientes para conectar otros sensores: 0xE0, 0xE2, 0xE4, 0xE6, 0xE8, 0xEA, 0xEC, 0xEE, 0xF0, 0xF4, 0xF6, 0xF8, 0xFA, 0xFC, 0xFE.

4.2.4.1. Registros

El SRF02 está compuesto por un conjunto de 6 registros, en la Tabla 3 se destacan:

Posición	Lectura	Escritura
0	Revisión de software	Registro de comando
1	No se usa (se lee 0X80)	N/A
2	Byte alto de la medición	N/A
3	Byte bajo de la medición	N/A
4	Byte algo-valor mínimo de distancia	N/A
5	Byte bajo-valor mínimo de distancia	N/A

Tabla 3. Registros del sensor SRF02.

El “registro 0” es el único en el que puede escribirse, es usado para iniciar una nueva medición. No puede ser leído. Cada medición tarda 65mS en efectuarse, mientras tanto el SRF02 no responderá a ningún comando en el bus I2C.

Los “registros 2 y 3” son de 16 bits cada uno, almacenan la última medición sin signo en pulgadas, centímetros o microsegundos según el comando que se utilice. Un cero indica que el sensor no ha detectado ningún objeto y debe tenerse en cuenta no realizar mediciones sin tener en cuenta los 65ms que tarda un haz ultrasónico en regresar, con esto se da tiempo a que la anterior ráfaga se desvanezca y no interfiera en la nueva medición.

En los “registros 4 y 5” de 16 bits cada uno, se almacena la distancia mínima que puede medir el sensor SRF02, la cual normalmente es de 11 a 16 cm (4” a 6”).

4.2.4.2. Comandos

Hay 3 comandos para iniciar una medición (80, 81 y 82) con los cuales establecen si el resultado de esta será en centímetros, pulgadas o microsegundos.

El otro conjunto de comandos (86, 87 y 88) hacen lo mismo pero sin transmitir ráfaga ultrasónica, se usan para detectar la ráfaga ultrasónica de otro sensor, esto ayuda para sincronizar los comandos enviados a los dos sonares.

El comando 92 transmite una ráfaga ultrasónica pero no hace recepción de la misma.

El comando 96 reinicia el SRF02 realizando un ciclo de auto ajuste. Es como si se conectara la alimentación.

El conjunto de comandos (160, 165 y 170) se utiliza para cambiar la dirección del bus I2C. En la

Tabla 4 se muestra lo anterior dicho:

Comandos		Descripción
Decim al	Hexadecim al	
80	0x50	Modo de medición real. Resultado en pulgadas.
81	0x51	Modo de medición real. Resultado en centímetros.
82	0x52	Modo de medición real. Resultado en microsegundos.
86	0x56	Modo de medición falsa. Resultado en pulgadas.

87	0x57	Modo de medición falsa. Resultado en centímetros.
88	0x58	Modo de medición falsa. Resultado en microsegundos.
92	0x5C	Transmite una ráfaga de 8 ciclos de 40KHz-No hace cálculos de medición.
96	0x60	Fuerza un reinicio del sensor realizando un ciclo de autoajuste.
160	0xA0	1 comando de la secuencia para cambiar la dirección I2C.
165	0xA5	2 comando de la secuencia para cambiar la dirección I2C.
170	0xAA	3 comando de la secuencia para cambiar la dirección I2C.

Tabla 4. Comandos del sensor SRF02.

Para iniciar una medición se escribe un comando del primer conjunto y se espera un tiempo apropiado para la finalización (aproximadamente 65 ms) y después se procede a leer los registros 2 y 3 donde se almacena la distancia medida por el sensor en centímetros, pulgadas o microsegundos, dependiendo del comando inicial.

Para chequear que la medición haya terminado no es necesario tener un temporizador en el programa, se puede aprovechar que el sensor SRF02 no responde a ninguna operación en el bus I2C mientras está midiendo, por lo tanto si se intenta leer el registro 0 (revisión de software) el sensor arrojará 255 (0xFF) mientras esté ocupado, tan pronto termine, volverá a responder al bus I2C, así que se leerá el registro 0 hasta que ya no sea 255 (0xFF) pues en ese momento ya se tendrá lista una medida del sensor. [8]

4.3. Comunicación I2C entre sensores SRF02 y Arduino

Al conectar un sensor SRF02 al Arduino mediante comunicación I2C, este le asigna la dirección 0xE0 por defecto, pero como la tarjeta Arduino es de 7 bits y el SRF02 de 8 bits, se tiene que despreciar el bit menos significativo, con lo cual la dirección será 0x70. Si se quieren conectar otros sensores se debe tener en cuenta lo anterior. En la

Tabla 5 se describe el proceso para conectar 8 sensores SRF02:

SENSOR SRF 02		TARJETA ARDUINO	
HEXADECIMAL	BINARIO	BINARIO	HEXADECIMAL
0xE0	11100000	1110000	0x70
0xE2	11100010	1110001	0x71

0xE4	11100100	1110010	0x72
0xE6	11100110	1110011	0x73
0xE8	11101000	1110100	0x74
0xEA	11101010	1110101	0x75
0xEC	11101100	1110110	0x76
0xEE	11101110	1110111	0x77
...
...
0xFE	11111110	1111111	0x7F

Tabla 5. Cambio de la dirección por defecto de la comunicación I2C entre sensores y Arduino.

Para establecer la comunicación I2C entre los sensores SRF02 y el Arduino, se implementó la librería Wire de Arduino.

4.3.1. Librería Wire

Esta librería permite comunicar la mayoría de las tarjetas Arduino con dispositivos que funcionan a través de comunicación I2C / TWI. La línea de datos (SDA) está en el pin analógico 4, y la línea de reloj (SCL) está en el pin analógico 5. En Arduino Mega, SDA está en el pin digital 20 y SCL en el 21. La librería Wire usa 7 bits de direccionamiento, así que como los sensores que se van a implementar son de 8 bits, se tendrá que renunciar al bit más bajo y además desplazar el valor un bit a la derecha, cediendo una dirección entre 0 y 127. En la

Tabla 5 Se expone el procedimiento que se hizo para darle una dirección específica a cada sensor, teniendo en cuenta que el Arduino trabaja con 7 bits. [9]

4.4. CONFIGURACIÓN DEL DRIVER ROBOCLAW 2X30A [10]

A continuación se expone el modo de funcionamiento con el que se escogió trabajar el proyecto: Manejo de los motores de una silla de ruedas mediante comandos de voz.

4.4.1. Modo 1: Entrada RC

Se usa para controlar el driver ROBOCLAW a través del control señales de radio RC, lo cual es usado frecuentemente para controlar modelos de vehículos desde un radio transmisor portátil. El modo RC también puede ser usado para simplificar el manejo del driver mediante un microcontrolador usando PWM.

Para configurar este modo se pone SW1 ON como se muestra en la siguiente figura.

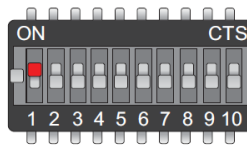


Figura 32. Configuración del DIP Switch en modo RC, imagen tomada de: Datasheet driver RoboClaw 2X30A.

Hay 4 opciones para controlar el modo RC, lo cual se hace a través de los switches: SW4, SW5, SW6, y SW7. A continuación se describen:

4.4.1.1. Switch 4 – Modo mixing

SW4 = ON: Modo mixing activado. S1 controla adelante y reversa. S2 controla la dirección. Este control puede ser usado en un carro.

SW4 = OFF: Modo mixing apagado. S1 controla la velocidad y dirección del motor 1. S2 controla la velocidad y dirección del motor 2.

4.4.1.2. Switch 5 – Modo Exponencial

SW5 = ON: Modo exponencial activo. Suaviza el control de posición. Este modo es ideal para el control de robots estilo tanque.

SW5 = OFF: Modo exponencial apagado. La respuesta del motor será lineal y directamente proporcional a la entrada de control. Ideal para robots de 4 ruedas.

4.4.1.3. Switch 6 – MCU o Control RC

SW6=ON: Modo MCU activo. RoboClaw continuará ejecutando el último pulso recibido hasta recibir uno nuevo. La prueba de fallos y calibración automática están desactivadas en este modo.

SW6=OFF: Control RC activo. En este modo el driver RoboClaw calibrará el punto central y final automáticamente. Este modo incluye una prueba de fallos. El driver se apagará si la entrada de control se pierde.

4.4.1.4. Switch 7 – Entrada Flip Switch

SW7=ON: Se requiere un pulso de servo para activar la entrada del Flip Switch, un pulso mayor a 1.5ms revertirá el control de la dirección. Este interruptor se suele utilizar en robots de combate para revertir automáticamente los controles si un enemigo se dio la vuelta.

SW7=OFF: en este caso la entrada Flip switch espera una señal de control TTL. 0V para activar y 5V para funcionar normal.

El driver RoboClaw incluye una forma de protección para bajo voltaje a través del DIPSwitch, con los switches: SW8, SW9 y SW10. Esto es hecho con dos propósitos principales, para cuando el voltaje de la batería sea muy bajo el driver deje de funcionar correctamente y para cuidar que la batería de litio no se dañe. A continuación se muestra las posibilidades de configuración:

Voltage	SW8	SW9	SW10
Not Monitored	OFF	OFF	OFF
Lead Acid - Auto	ON	OFF	OFF
2- Cell (6V Cutoff)	OFF	ON	OFF
3- Cell (9V Cutoff)	ON	ON	OFF
4- Cell (12V Cutoff)	OFF	OFF	ON
5- Cell (15V Cutoff)	ON	OFF	ON
6- Cell (18V Cutoff)	OFF	ON	ON
7- Cell (21V Cutoff)	ON	ON	ON

Figura 33. Protección de bajo de voltaje, imagen tomada de: Datasheet Driver RoboClaw

4.4.2. Conexiones del driver RoboClaw 2x30A

Se seleccionó el modo RC y la protección a bajo voltaje de 9V, a continuación se muestra la configuración del DIPSwitch

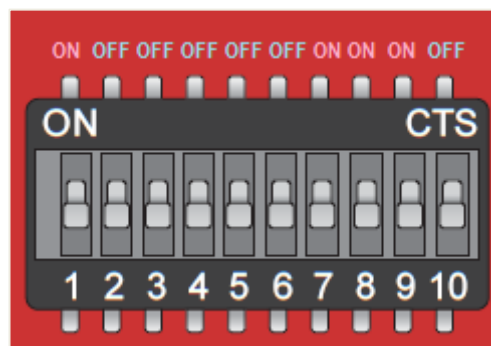


Figura 34. Estado de los interruptores del DIPSwitch del driver RoboClaw, imagen tomada de datasheet Driver RoboClaw 2x30A, editada en Paint.

Para el desarrollo de este proyecto se utilizó el Jumper conectado a LB para separar la fuente principal de la regulada de 5V que alimenta los dispositivos Arduino e EasyVR y así no generar conflictos entre ellas, observar la siguiente figura:

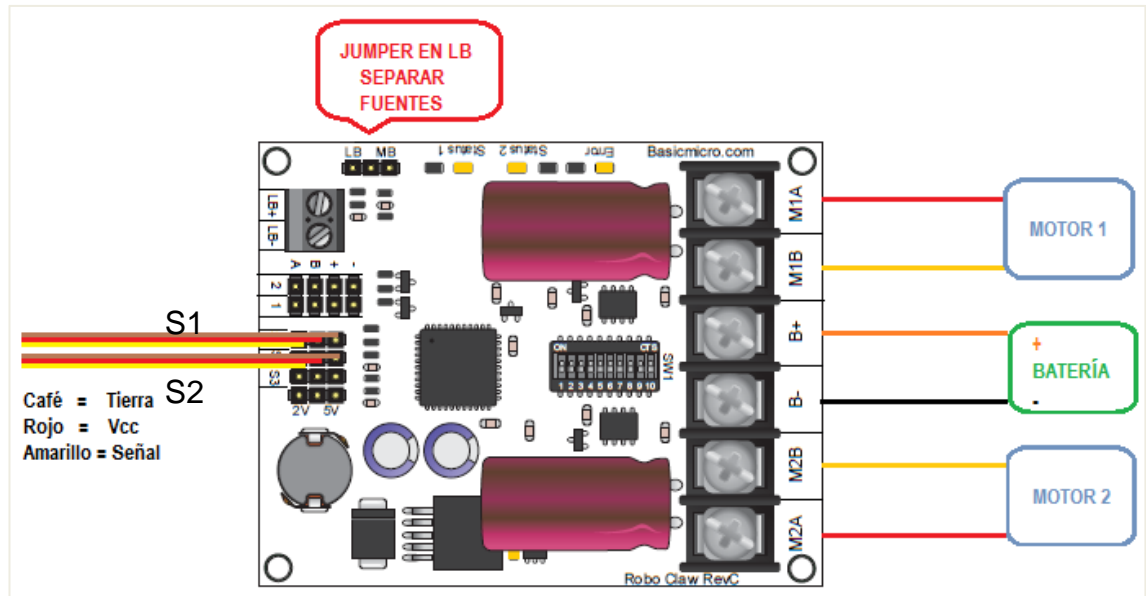


Figura 35. Conexiones del driver RoboClaw 2x30A, imagen tomada de datasheet Driver RoboClaw 2x30A, editada en Paint.

Para el movimiento de los motores el driver RoboClaw espera unos pulsos de PWM en S1 y S2, en la anterior figura se muestra la ubicación de estos. 1000us es para reversa a plena velocidad y 2000us para marcha adelante a plena velocidad; el punto medio entre 1000 y 2000us, es decir 1500us mantiene detenidos los motores.

A continuación se muestra una tabla que contiene los valores de PWM definidos para el manejo de la silla de ruedas eléctrica a través de los comandos de voz.

COMANDOS		VALOR MIN	VALOR MÁX
Adelante	M1 y M2	1520	1650
Atrás	M1 y M2	1480	1350
Izquierda	M1	1520	1570
	M2	1480	1420
Derecha	M1	1480	1430
	M2	1520	1570
Para	M1 y M2	1500	1500

Tabla 6. Señal de PWM para la ejecución de los comandos de voz

Se implementó un valor mínimo y máximo para que el arranque no sea brusco sino que vaya aumentando en forma de rampa, con esto se ayuda a que el

usuario de la silla esté más cómodo con el avance suave de la silla. A continuación se visualiza la ejecución del comando “adelante”, donde se fue variando el PWM cada 40us a partir de 1520 hasta 1650.

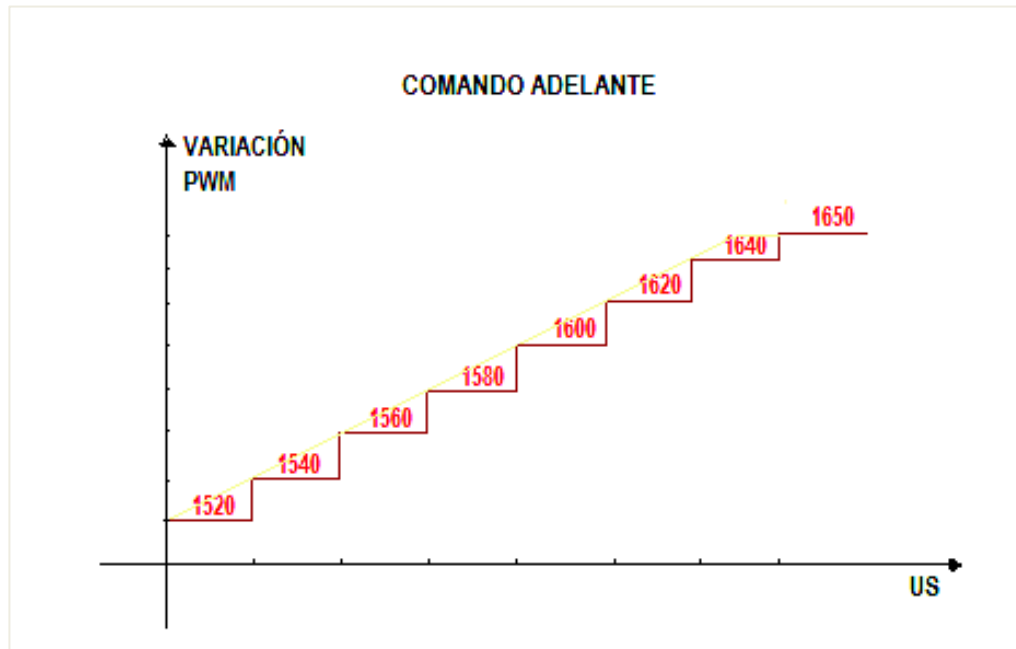


Figura 36. Ejecución del comando "adelante", imagen hecha en Paint

Los motores usados para el movimiento de la silla de ruedas son motores DC, los cuales tienen las siguientes especificaciones técnicas:

Voltaje: 24V	180 Watts
Corriente: 10A	Ratio: 32:1
3800 RPM	Brakes: 24Vdc

Tabla 7. Especificaciones técnicas del motor

A una velocidad normal, la velocidad de los motores es 260 RPM. Cuando se ejecuta el comando "aumentar velocidad", los motores alcanzan los 320 RPM.

4.4.3. Estado y error de los leds

El driver RoboClaw 2x30A cuenta con 3 leds principales (ver Figura 37), 2 indicadores de estado y 1 Led de error, los cuales indicarán ciertas condiciones dependiendo del modo de trabajo seleccionado y partiendo de que para este proyecto se escogió el modo RC, se explicarán a continuación los posibles estados del driver:

4.4.3.1. Leds de estado:

- ▶ Estado del LED 1 = Continuamente encendido, parpadea cuando recibe un pulso.
- ▶ Estado del LED 2 = Encendido cuando los motores están activos.

4.4.3.2. Led de error

- ▶ Sobre-Corriente = Led encendido. Los leds de estado 1 y 2 indican si el motor afectado es el 1 y o 2.
- ▶ Sobre-temperatura= El led de error parpadea una vez con una larga pausa.
- ▶ Batería principal baja = El led de error parpadea dos veces con una larga pausa.
- ▶ Batería principal alta = El led de error parpadea rápido hasta que la condición cambie.
- ▶ Batería lógica baja = El led de error parpadea tres veces con una larga pausa.
- ▶ Batería lógica alta = El led de error parpadea cuatro veces con una larga pausa.

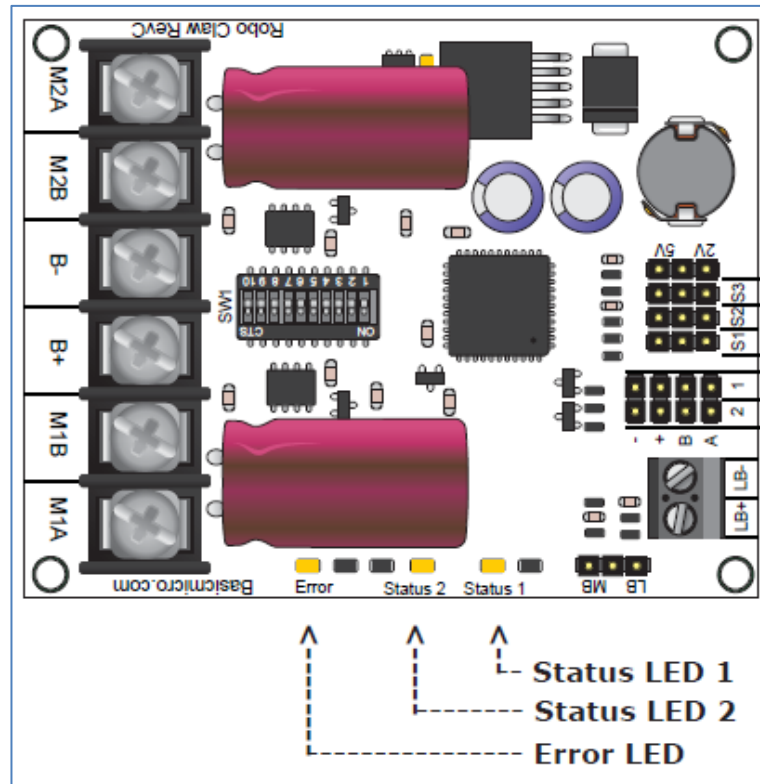


Figura 37. Leds de estado y de error del driver RoboClaw 2x30A, imagen tomada de Datasheet Driver RoboClaw 2x30A.

4.5. CONFIGURACIÓN DEL MÓDULO DE RECONOCIMIENTO DE VOZ Y REPRODUCCIÓN EASYVR [11]

El módulo Easy VR se comporta como un dispositivo esclavo que se controla desde un dispositivo Master o controlador mediante una comunicación serie asíncrona tipo UART con una velocidad de comunicación de 9600 baudios por defecto, aunque se puede cambiar a 19200, 38700, 57600, 115200 y una trama de 8 bits de datos, 1 bit de parada y sin paridad. Para este proyecto se utilizó la velocidad de comunicación por defecto. Se recomienda tener cuidado con las conexiones de las señales ETX, ERX y alimentación, ya que las señales de comunicación de datos son compatibles con niveles de tensión TTL, por lo cual no se pueden conectar directamente al puerto COM del PC. Observar la conexión en la Figura 38:

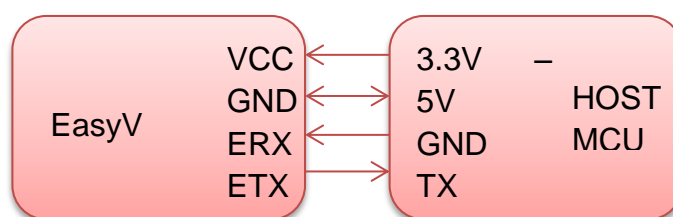


Figura 38. Conexión entre el EasyVR y un Host

Para el desarrollo de este proyecto no hubo problema con las conexiones porque se usó un shield EasyVR para conectar directamente al Arduino, en el ya vienen hechas todas las conexiones necesarias para su implementación, observar la Figura 39.

El módulo EasyVR incluye un <<micrófono unidireccional del tipo electret (Horn EM9745P-382): Sensibilidad -38dB; Impedancia 2.2K; Tensión de trabajo 3V; Respuesta a un amplio rango de frecuencias de 100Hz hasta 20Khz>>⁶

La postura correcta del micrófono es de suma importancia porque si hay una colocación acústica inadecuada en el micrófono, se disminuirá la posibilidad de que las voces y comandos sean reconocidas. Para esta aplicación se implementó la carcasa de unos auriculares de diadema, en la cual se instaló el micrófono, con lo cual se disminuyó la posibilidad que existan fallas en el reconocimiento de voz pues este permanece en una posición fija.

La distancia entre la boca y el micrófono es otro factor importante al momento de realizar reconocimiento por voz, se recomienda que el micrófono no esté a más de 10 cm de la boca. En este aspecto también influye que el usuario de la silla dicte el comando de forma muy fuerte o muy suave, pues se puede crear resonancia y por lo tanto afectar el reconocimiento de voz, por lo tanto se debe pronunciar y vocalizar correctamente cada palabra.

El módulo de reconocimiento de voz y reproducción EasyVR soporta una salida de audio de 8 ohmios. Conectar directamente altavoces de menor valor puede dañar la salida de audio o el módulo en su totalidad. El Shield EasyVR para Arduino cuenta con una salida de audio o zócalo Jack de 3.5mm para conectar directamente unos auriculares. En la Figura 39 se muestra la conexión entre el Shield EasyVR y Arduino Mega.

⁶ EasyVR guía rápida de usuario

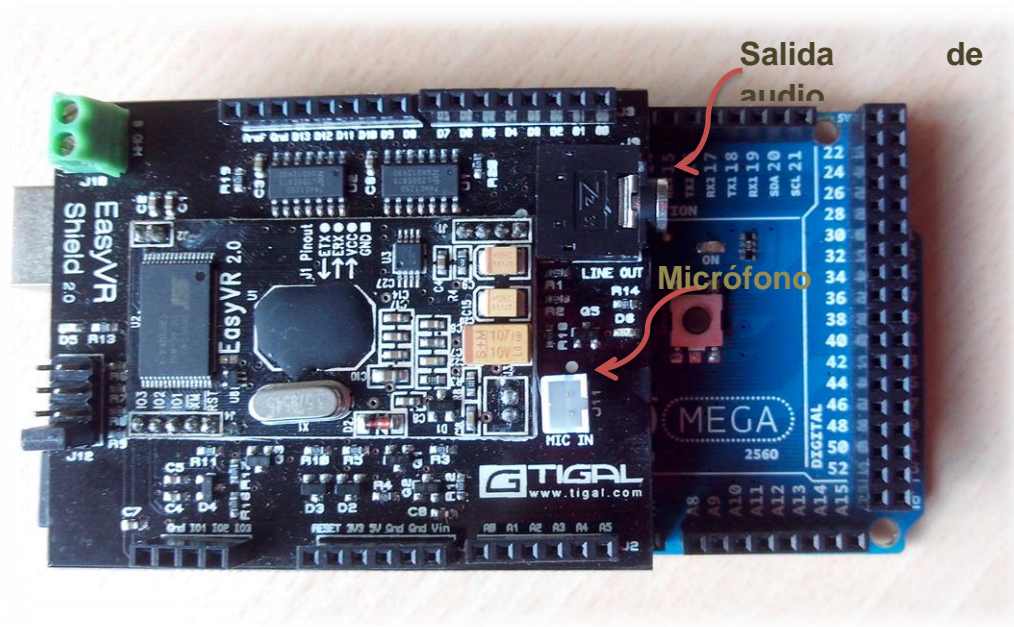


Figura 39. Conexión del Shield EasyVR y Arduino Mega.

Este módulo de reconocimiento de voz trae comandos predefinidos en diferentes idiomas y permite grabar hasta 32 más como se explicó en el numeral 4.3. Para usar y configurar el módulo EasyVR se utilizó el software EasyVR commander, el cual se comunica con el puerto COM del PC, conectado directamente al sistema Arduino.

4.5.1. EasyVR commander: INTERFAZ GRÁFICO DE USUARIO GUI

Al ejecutar el programa aparece la ventana de trabajo que se observa en la Figura 40, donde se tienen grupos de 0-16 en los que se ubican los comandos de voz definidos por el usuario organizados como convenga, los wordsets 1-3 donde están las voces predefinidas por el sistema y SoundT donde también se pueden almacenar diferentes sonidos digitalizados para su posterior reproducción.

Al lado de cada grupo aparece la cantidad de comandos guardada en cada uno.



Figura 40. EasyVR commander, área de trabajo, imagen tomada de: EasyVR User Manual.

En el área de comandos se edita/visualiza las etiquetas asociadas a las voces y la posición que ocupan en el grupo seleccionado.

4.5.2. Voces definidas por el usuario

Para grabar voces y asociarlas a un comando se selecciona el grupo en el que se quiere guardar, luego en el área de tareas mostrado en la Figura 40 hay un ícono llamado “add command”, con el cual se podrán agregar todos los comandos deseados. También se pueden utilizar los íconos “remove command” y “rename command” para borrar o editar etiquetas respectivamente, de los comandos agregados. A continuación en la Figura 41 se muestra 5 comandos definidos por el usuario, ubicados en el grupo 2, a los que se les asignó automáticamente un index.

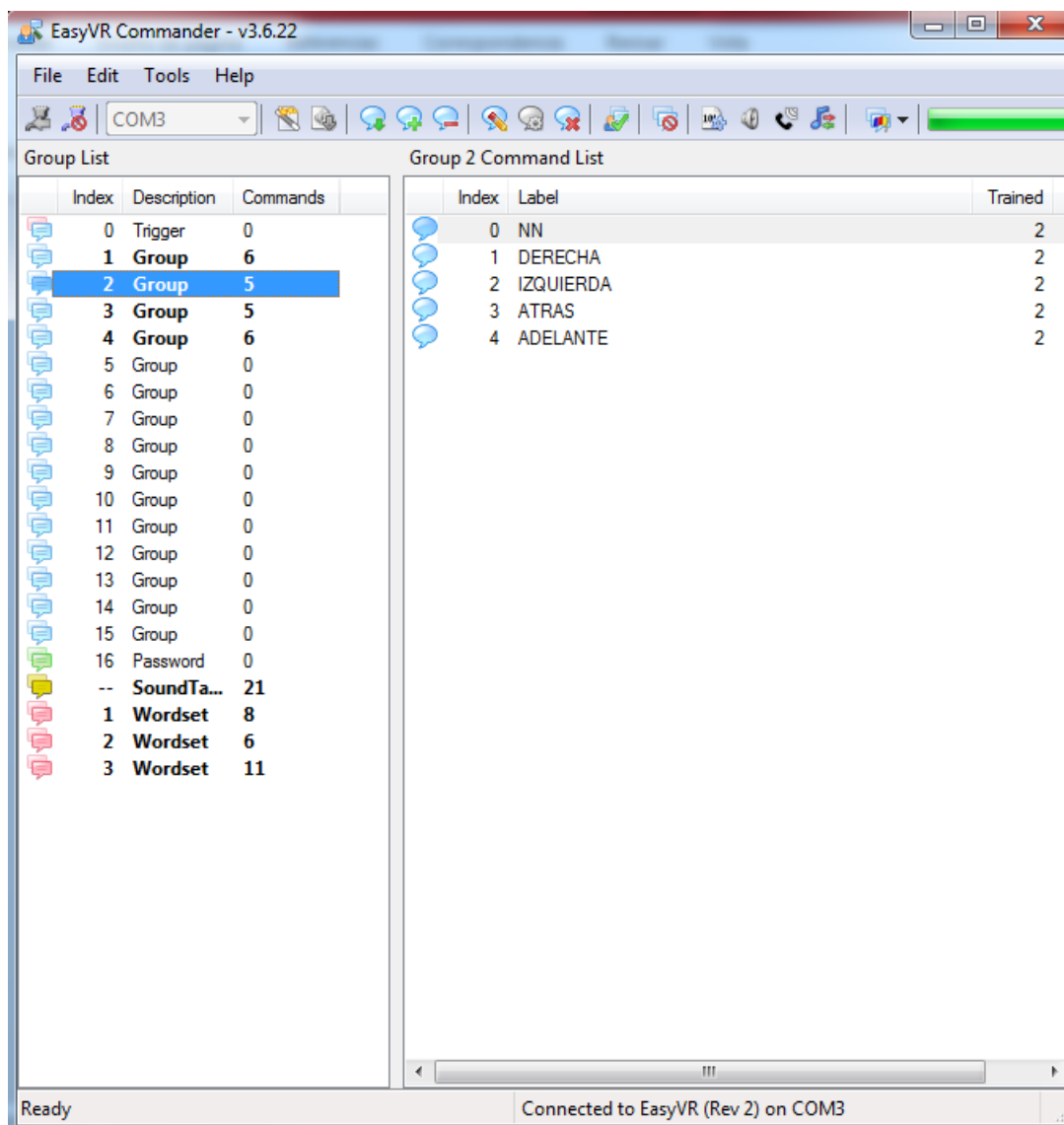


Figura 41. Grabar comandos de voz en EasyVR commander, imagen tomada de EasyVR Commander 3.6

Para asignarle una voz a cada etiqueta se utiliza el ícono “train command”, el solicitará una primera dicción de la voz a asociar, después de ser reconocida, indicará “1” en la columna Trained y solicitará una segunda dicción, es decir, hay dos fases de reconocimiento de voz para asociarla a una etiqueta. Si ambas coinciden, aparecerá “2” en la columna Trained y la voz quedará asociada a la etiqueta. En caso contrario indicará el error de reconocimiento y habrá que repetir el procedimiento.

4.5.3. Comandos de voz implementados para el funcionamiento de la silla

Para poner en marcha los motores de la silla de ruedas, se estipuló un comando inicial que después de ser reconocido por el sistema EasyVR, se tienen 2 segundos para dictar la orden que el usuario desee ejecutar, ya sea: adelante, atrás, derecha, izquierda, aumentar o disminuir, y también un comando para detener el sistema. Estas acciones pueden ser asociadas a la voz que el usuario desee asignarle a cada una a través de la interfaz GUI EasyVR commander, por ejemplo, asignar la palabra “avanzar” a la acción “adelante”.

4.5.4. Mensajes de realimentación

Se cargaron al EasyVR commander los siguientes mensajes en formato WAV, con 16 bits sin compresión y monofónicos de 22050Hz:

Movimiento no permitido

Objeto adelante

Objeto atrás

Objeto a la derecha

Objeto a la izquierda

Espacio abierto

Espacio cerrado

Los cinco primeros se utilizan cuando la silla se enfrenta a obstáculos y tenga que detenerse para evitar chocar, de tal manera que si el usuario dicta el comando “adelante” y uno de los sensores delanteros capta que a poca distancia hay un obstáculo el sistema no ejecute el comando, sino que emita el mensaje “movimiento no permitido, objeto adelante”, con lo cual se le explica al usuario la razón por la que la silla de ruedas está estática.

Los dos últimos sirven para alertar al usuario del entorno en el que se encuentra cuando pronuncia el comando de aumentar velocidad, es decir, si el espacio es abierto se reproducirá “espacio abierto” y habrá cambio en la velocidad, pero si el espacio es cerrado se reproducirá “espacio cerrado” y se mantendrá una velocidad baja.

4.5.5. Pruebas con los comandos predefinidos del sistema de reconocimiento de voz EasyVR

Se realizaron pruebas con 5 personas, entre ellas 2 mujeres y 3 hombres, a las cuales se les hizo repetir 10 veces todos los comandos que se encargan del movimiento de los motores de la silla de ruedas.

A continuación se muestran los datos medidos organizados en tablas, donde 1 significa acierto y 0 desacierto, es decir, 1 si el comando de voz es reconocido por la tarjeta EasyVR y 0 que el usuario no pronunció de forma correcta el comando de voz y la tarjeta EasyVR no lo reconoce.

TAR=Total de aciertos por ronda
TAC=Total de aciertos por comando

Usuario 1											
	1	2	3	4	5	6	7	8	9	10	TAC
Mover adelante	1	1	1	1	1	1	1	1	1	1	10
Mover atrás	1	1	1	1	1	1	1	1	1	1	10
Mover a la derecha	1	1	1	1	1	1	1	1	1	1	10
Mover a la izquierda	1	1	1	1	1	1	1	1	1	1	10
Para	1	0	0	1	0	0	1	0	1	1	5
TAR	5	4	4	5	4	4	5	4	5	5	

Tabla 8. Prueba 1 con el sistema de comandos de voz EasyVR

NOTA:

El usuario uno tuvo dificultad con el comando “para” en 5 de 10 rondas.

Usuario 2											
	1	2	3	4	5	6	7	8	9	10	TAC
Mover adelante	1	1	1	1	1	1	1	1	1	1	10
Mover atrás	1	1	1	1	1	0	1	1	1	1	9
Mover a la derecha	1	1	1	1	1	0	1	1	1	1	9
Mover a la izquierda	1	1	1	1	1	0	0	1	1	1	8
Para	1	1	1	1	1	1	1	1	1	1	10
TAR	5	5	5	5	5	2	4	5	5	5	

Tabla 9. Prueba 2 con el sistema de comandos de voz EasyVR

NOTA:

El usuario 2 presentaba problemas nasales a causa de gripa. Repitió muchas veces el comando "MOVER" pues se le dificultaba pronunciar la letra “ere”.

Usuario 3											
	1	2	3	4	5	6	7	8	9	10	TAC
Mover adelante	1	1	1	1	1	1	1	1	1	1	10
Mover atrás	1	1	1	1	1	1	1	1	1	1	10
Mover a la derecha	1	1	1	1	1	1	1	1	1	1	10
Mover a la izquierda	1	1	1	1	1	1	1	1	1	1	10
Para	1	1	1	1	0	1	1	0	1	1	8
TAR	5	5	5	5	4	5	5	4	5	5	

Tabla 10. Prueba 3 con el sistema de comandos de voz EasyVR

NOTA:

Este usuario tuvo problemas con el comando “para” en 2 de 10 rondas y con el comando “mover” en aproximadamente 3 de cada 10 pronunciaciones.

Usuario 4											
	1	2	3	4	5	6	7	8	9	10	TAC
Mover adelante	1	1	1	1	1	1	1	1	1	1	10
Mover atrás	1	1	1	1	1	1	1	1	1	1	10
Mover a la derecha	1	1	1	1	1	1	1	1	1	1	10
Mover a la izquierda	1	1	1	1	0	1	1	1	1	1	9
Para	1	1	1	1	1	1	1	1	1	1	10
TAR	5	5	5	5	4	5	5	5	5	5	

Tabla 11. Prueba 4 con el sistema de comandos de voz EasyVR

NOTA:

Este usuario tuvo una logró el buen manejo de la silla debido a la buena pronunciación de los comandos. En todo el proceso, el comando “mover a la izquierda” tuvo 1 fallo de 10 posibles.

Usuario 5											
	1	2	3	4	5	6	7	8	9	10	TAC
Mover adelante	1	1	1	1	1	1	1	1	1	1	10
Mover atrás	1	1	1	1	1	1	1	1	1	1	10
Mover a la derecha	1	1	1	1	1	1	1	1	1	1	10
Mover a la izquierda	0	0	1	1	1	1	1	1	1	0	7
Para	1	1	1	1	1	1	1	1	1	1	10
TAR	4	4	5	5	5	5	5	5	5	4	

Tabla 12. Prueba 5 con el sistema de comandos de voz EasyVR

NOTA:

En esta prueba hubo problemas con el comando inicial “mover” 2 de 10 pronunciaciones y con el comando “mover a la izquierda” 3 de 10 pronunciaciones. Se notó dificultad con la letra “ere”.

4.5.6. Prueba con comandos personalizados del sistema de reconocimiento de voz EasyVR

Se realizó una prueba con un estudiante de Ingeniería Civil de la Universidad Pontificia Bolivariana, quien como discapacitado logró darle funcionamiento a la silla de ruedas. Partiendo de que el sistema de reconocimiento de voz puede ser configurado para cada persona, este usuario le asignó las siguientes voces a los movimientos predeterminados para el manejo de esta:

Comando inicial: Hache

Movimiento adelante: Adelante

Movimiento atrás: Atrás

Movimiento a la izquierda: izquierda

Movimiento a la derecha: derecha

Aumentar velocidad: aumentar

Disminuir velocidad: disminuir

El funcionamiento de la silla fue correcto, evitando colisiones con obstáculos y poniendo en marcha los motores a partir de los comando de voz dictados por el estudiante, el cual sugirió modificar el chasis de la misma debido a sus imperfecciones.

4.6. COMUNICACIÓN ENTRE EL ARDUINO MEGA Y EL ANTDUINO

Se estableció comunicación serial entre estos, siendo el arduino Mega el encargado de procesar los comandos de voz y enviarlos al Antduino, desde el cual se le envía una señal PWM correspondiente a la ejecución de cada comando de voz al driver RoboClaw 2x30A. Ver la Tabla 6. El Antduino también tiene la función de procesar la información captada por los sensores SRF02 conectados alrededor de la silla de rueda y evitar posibles colisiones.

A continuación se muestra la conexión para establecer comunicación serial entre dos dispositivos:

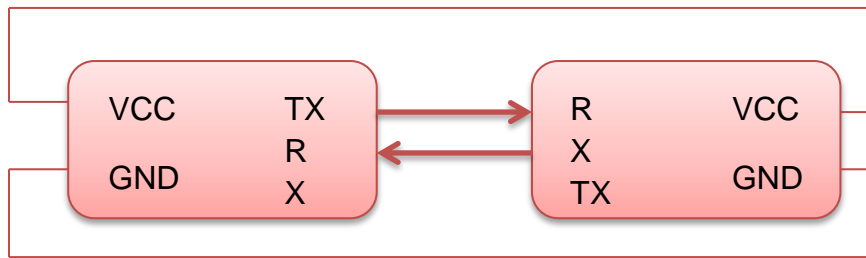


Figura 42. Conexión para establecer comunicación serial entre dos dispositivos, imagen hecha en Word.

5. CONCLUSIONES

- ▶ Con los resultados de la caracterización de los sensores ultrasónicos SRF02, se decidió emplear siete sensores para hacer una evaluación aceptable del entorno.
- ▶ Al implementar una variación escalonada de PWM para el manejo de los motores a través del driver RoboClaw, se mejoró el arranque y frenado de la silla con respecto a los proyectos anteriores, brindándole al usuario mayor comodidad al tener movimientos suaves.
- ▶ El sistema de comandos de voz implementado puede ser personalizado o configurado para cada persona, la cual le puede asignar la etiqueta de voz que desee a los movimientos predispuestos para el manejo de la silla de ruedas: adelante, atrás, derecha, izquierda, parar, aumentar, disminuir.
- ▶ El sistema de reconocimiento de voz seleccionado es efectivo, y su desempeño mejora progresivamente a partir de la cantidad de experiencias que tenga el usuario, al ir memorizando la pronunciación correcta de los comandos.

6. RECOMENDACIONES

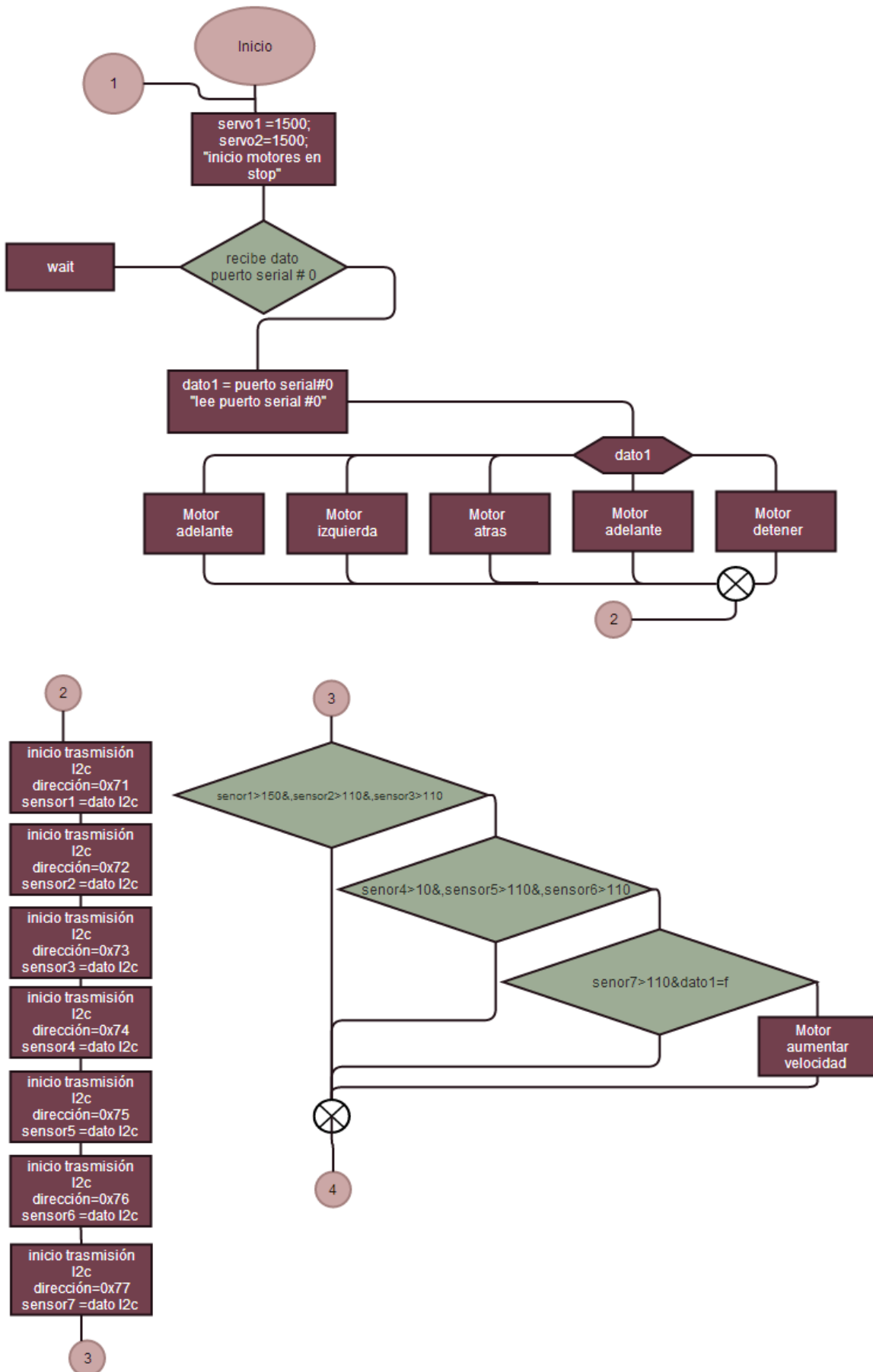
- ▶ Los sensores SRF02 ubicados en la silla captan obstáculos presentes y no ausentes, tales como escaleras o huecos. Se recomienda implementar un sistema que permita solucionar esta falencia.
- ▶ Mejorar el chasis de la silla de ruedas para obtener un mejor desempeño en su funcionamiento.
- ▶ Mantener el micrófono estático a una distancia no tan próxima a la boca para no generar resonancia, pronunciando pausadamente los comandos de voz.
- ▶ Continuar mejorando las características y funciones de la silla de ruedas, en busca de un dispositivo que pueda ser comercializado para brindarle herramientas de movilidad a personas discapacitadas.

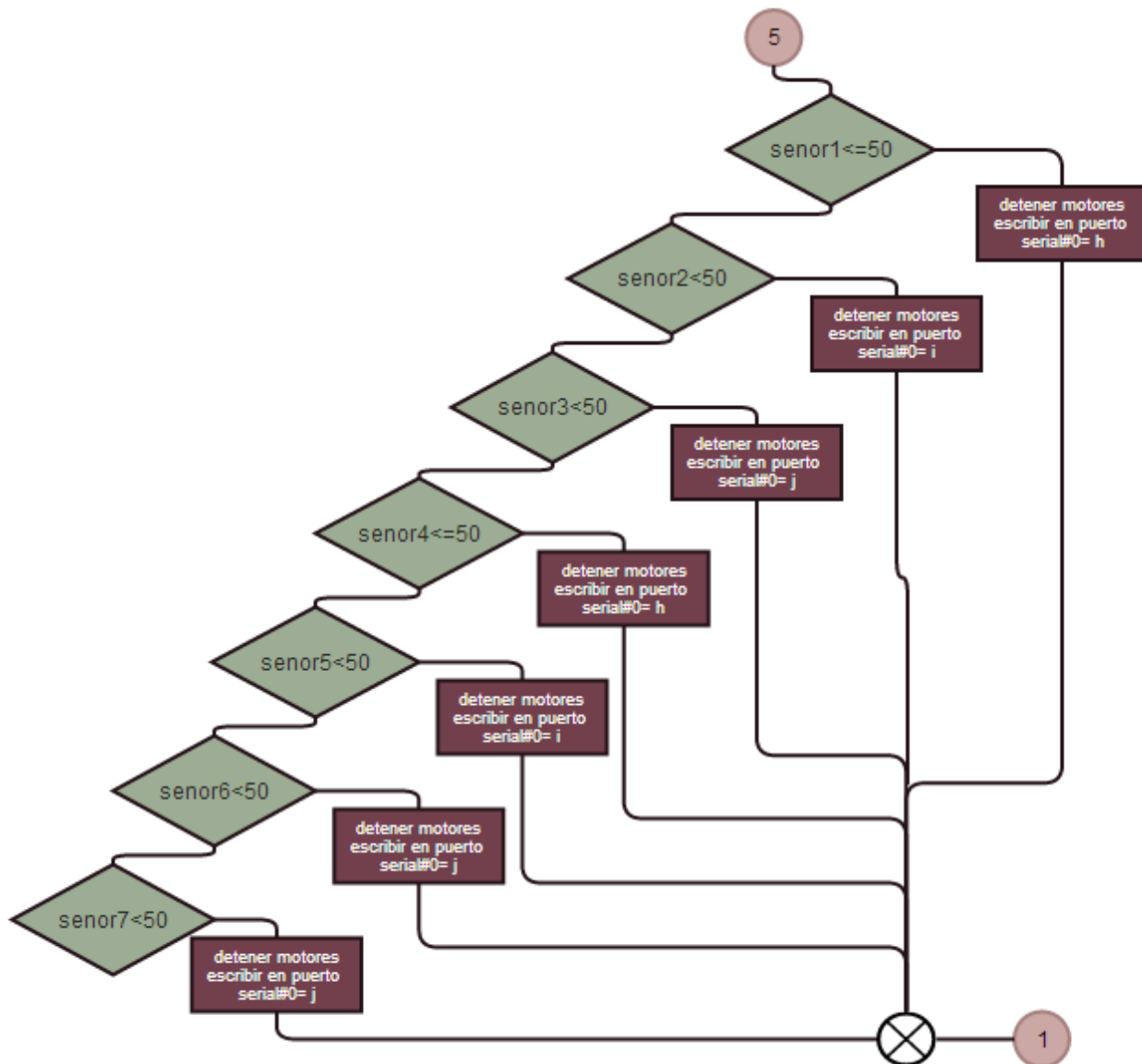
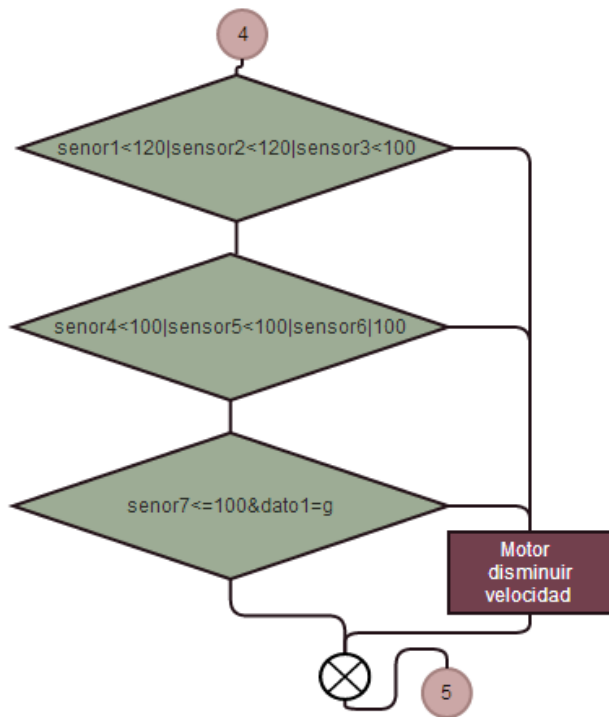
7. BIBLIOGRAFÍA

- [1] SORO-CAMATS, Emili. La escuela y los alumnos con discapacidad motriz. CL & E: Comunicación, lenguaje y educación, 1994, no 22, p. 23-36.
- [2] Discapacidad motriz, Guía didáctica para la inclusión en educación inicial y básica, Consejo Nacional de Fomento Educativo, CONAFE, México, 2010.
- [3] Página web: <http://www.arduino.cc/>
- [4] Guía rápida del módulo de reconocimiento de voz y reproducción, EasyVR, página web: <http://www.msebilbao.com/notas/downloads/EasyVR%20Guia%20rapida.pdf>.
- [5] Sensores de proximidad, página web: <http://sensoresdeproximidad.blogspot.com/>.
- [6,10] Pololu. Descripción de productos. Página web: <http://www.pololu.com/catalog/product/1497>. Citado 11 diciembre 2012.
- [7] Datasheet sensor ultrasónico SRF02. Página web: <http://www.robot-electronics.co.uk/htm/srf02tech.htm>, 2014.
- [8] Comunicación I2C sensor ultrasónico SRF02. Página web: <http://www.robot-electronics.co.uk/htm/srf02techI2C.htm>, 2014.
- [9] Librería Wire de Arduino. Página web: <http://arduino.cc/es/Reference/Wire>.
- [11] EasyVR User Manual. Página web: <http://www.veear.eu/products/easyvr/>

8. ANEXOS

8.1. DIAGRAMA DE FLUJO, TARJETA ANTDUINO





8.2. DIAGRAMA DE FLUJO, TARJETA ARDUINO MEGA

