



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE COMPUTACIÓN

**TEMA:
ELABORACIÓN DE UN PROTOTIPO DE RECONOCIMIENTO
DE VOZ PARA AUTOMATIZACIÓN MEDIANTE
MICROCONTROLADORES ARM.**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Ciencias de la Computación

AUTOR: BRYAN DAVID VELASCO MEDINA

TUTOR: MANUEL RAFAEL JAYA DUCHE

Quito – Ecuador

2023

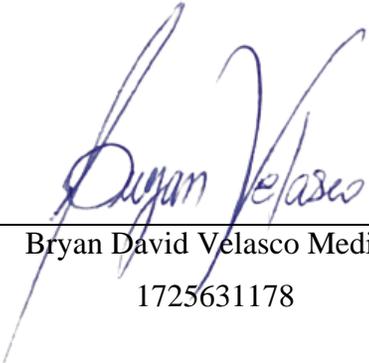
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Bryan David Velasco Medina con documento de identificación N° 1725631178 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 14 de agosto del año 2023

Atentamente,



Bryan David Velasco Medina
1725631178

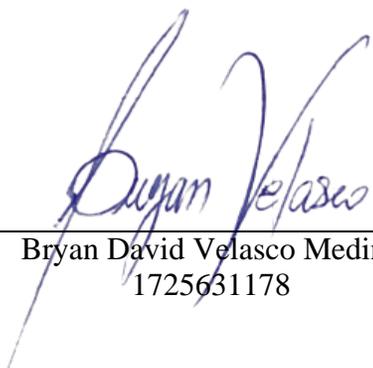
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Yo, Bryan David Velasco Medina con documento de identificación N° 1725631178, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto Técnico: “Elaboración de un prototipo de reconocimiento de voz para automatización mediante microcontroladores ARM.”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 14 de agosto del año 2023

Atentamente,



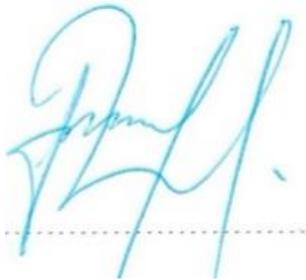
Bryan David Velasco Medina
1725631178

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Manuel Rafael Jaya Duche con documento de identificación N° 1710631035, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: ELABORACIÓN DE UN PROTOTIPO DE RECONOCIMIENTO DE VOZ PARA AUTOMATIZACIÓN MEDIANTE MICROCONTROLADORES ARM, realizado por Bryan David Velasco Medina con documento de identificación N° 1725631178, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 14 de agosto del año 2023

Atentamente,

A handwritten signature in blue ink, appearing to be 'Manuel R. Jaya Duche', written over a horizontal dashed line.

Ing. Manuel Rafael Jaya Duche, MSc.
1710631035

DEDICATORIA

El resultado de este trabajo va dedicado a mis padres que, con su amor, paciencia, esfuerzo y trabajo me han permitido culminar mi etapa educativa, así como su apoyo incondicional y motivación en los momentos difíciles de mi vida. Gracias por estar conmigo en mis momentos más difíciles y enseñarme a afrontar los problemas sin perder la esperanza.

A la memoria de mi abuelos, quiénes estuvieron conmigo y me vieron crecer, su apoyo y aliento a lo largo de mi vida me motivaron a cumplir con mis sueños y metas.

-Bryan Velasco Medina.

AGRADECIMIENTO

En primer lugar expresar mi gratitud a mis padres, ya que gracias a su esfuerzo, constancia y preocupación me enseñaron a nunca rendirme. Su apoyo incondicional me permitió cumplir con mis objetivos personales, su ayuda me impulso a no rendirme y siempre perseguir una meta más en la vida, así como no rendirse ante las adversidades y abandonarlas. Gracias a que me han brindado soporte económico pude estudiar una carrera universitaria, cumpliendo con uno de mis sueños planeados en mi vida, su apoyo fue vital en la culminación de mi etapa educativa.

Agradezco a mi padres por darme el ejemplo de humildad, honradez y trabajo duro, también por enseñarme a no conformarme y siempre buscar un camino de superación, así como darme fuerzas y ánimos para afrontar cualquier adversidad.

A las personas que formaron parte de mi desarrollo profesional, gracias por las horas fraternizadas, los trabajos realizados y por las historias de vida compartidas.

-Bryan Velasco Medina.

ÍNDICE

| | |
|---|----|
| INTRODUCCIÓN | 1 |
| ANTECEDENTES | 2 |
| PROBLEMA..... | 4 |
| JUSTIFICACIÓN | 7 |
| OBJETIVOS GENERAL Y ESPECÍFICO | 8 |
| OBJETIVO GENERAL:..... | 8 |
| OBJETIVOS ESPECÍFICOS: | 8 |
| METODOLOGÍA | 9 |
| CAPÍTULO 1 | 10 |
| 1. MARCO TEÓRICO/ESTADO DEL ARTE | 10 |
| 1.1. ¿Qué es la voz?..... | 11 |
| 1.2. ¿Qué es digitalizar la voz? | 12 |
| 1.3. Como se graba la voz para procesarla..... | 12 |
| 1.4. Procesamiento de la señal de voz..... | 13 |
| 1.5. Grabación digital..... | 13 |
| 1.6. Conversión analógico-digital | 13 |
| 1.7. Muestreo..... | 14 |
| 1.8. Cuantificación | 14 |

| | | |
|-----------------|---|----|
| 1.9. | Codificación | 14 |
| 1.10. | ¿Cómo funciona el reconocimiento de voz? | 14 |
| 1.11. | Comparación de patrones | 15 |
| 1.12. | Microcontroladores ARM | 15 |
| 1.13. | Tarjeta Audeme Movi Shield | 15 |
| 1.14. | Allwinner A13..... | 17 |
| 1.15. | Arduino UNO R3 | 18 |
| 1.16. | Automatización | 20 |
| CAPÍTULO 2..... | | 21 |
| 2. | DISEÑO Y DESARROLLO..... | 21 |
| 2.1. | DISEÑO..... | 21 |
| 2.1.1.1. | Ensamblaje de la tarjeta Movi con la placa Arduino | 21 |
| 2.1.1.2. | Audeme Movi Speech Recognizer and Synthesizer Shield for Arduino .. | 22 |
| 2.1.1.3. | Análisis hardware..... | 23 |
| 2.1.1.4. | Diagrama de bloques del sistema de reconocimiento de voz..... | 23 |
| 2.1.1.5. | Reconocimiento MOVI..... | 24 |
| 2.1.1.6. | Análisis software | 25 |
| 2.1.1.7. | Librería Movi | 26 |
| 2.1.1.8. | Proceso de grabado..... | 27 |
| 2.1.1.9. | Proceso de ejecución | 28 |

| | | |
|-----------------|-----------------------------------|----|
| 2.2. | Desarrollo del prototipo..... | 32 |
| 2.2.1. | Anillo LED RGB..... | 32 |
| 2.2.2. | Modelado del circuito..... | 33 |
| 2.2.3. | Motor reductor de doble eje | 34 |
| 2.2.4. | Controlador puente L298n | 35 |
| 2.2.5. | Modelado del circuito..... | 36 |
| 2.2.6. | Ventilador DC | 36 |
| 2.2.7. | Modulo relé | 37 |
| 2.2.8. | Modelado del circuito..... | 38 |
| 2.2.9. | Bombilla..... | 39 |
| 2.2.10. | Modelado del circuito..... | 40 |
| CAPÍTULO 3..... | | 41 |
| 3. | PRUEBAS Y RESULTADOS | 41 |
| 3.1. | Componentes | 41 |
| 3.1.1. | Led Neopixel RGB..... | 41 |
| 3.1.2. | Motor reductor DC | 52 |
| 3.1.3. | Ventilador..... | 60 |
| 3.1.4. | Bombilla..... | 65 |
| 3.2. | Comparación con Alexa | 71 |
| 3.3. | Análisis de resultados | 73 |

| | |
|---------------------------------|----|
| CONCLUSIONES | 74 |
| RECOMENDACIONES..... | 75 |
| REFERENCIAS BIBLIOGRÁFICAS..... | 76 |
| ANEXOS | 80 |

ÍNDICE DE FIGURAS

| | |
|-----------------|----|
| Figura 1 | 14 |
| Figura 2 | 16 |
| Figura 3 | 18 |
| Figura 4 | 18 |
| Figura 5 | 21 |
| Figura 6 | 21 |
| Figura 7 | 24 |
| Figura 8 | 26 |
| Figura 9 | 26 |
| Figura 10 | 27 |
| Figura 11 | 28 |
| Figura 12 | 28 |
| Figura 13 | 28 |
| Figura 14 | 28 |
| Figura 15 | 29 |
| Figura 16 | 29 |
| Figura 17 | 31 |
| Figura 18 | 31 |
| Figura 19 | 33 |

| | |
|----------------|----|
| Figura 20..... | 33 |
| Figura 21..... | 35 |
| Figura 22..... | 35 |
| Figura 23..... | 36 |
| Figura 24..... | 37 |
| Figura 25..... | 37 |
| Figura 26..... | 38 |
| Figura 27..... | 39 |
| Figura 28..... | 40 |
| Figura 29..... | 41 |
| Figura 30..... | 42 |
| Figura 31..... | 42 |
| Figura 32..... | 43 |
| Figura 33..... | 43 |
| Figura 34..... | 44 |
| Figura 35..... | 45 |
| Figura 36..... | 46 |
| Figura 37..... | 47 |
| Figura 38..... | 50 |
| Figura 39..... | 51 |

| | |
|-----------------|----|
| Figura 40 | 52 |
| Figura 41 | 52 |
| Figura 42 | 52 |
| Figura 43 | 53 |
| Figura 44 | 54 |
| Figura 45 | 55 |
| Figura 46 | 58 |
| Figura 47 | 59 |
| Figura 48 | 60 |
| Figura 49 | 60 |
| Figura 50 | 60 |
| Figura 51 | 61 |
| Figura 52 | 63 |
| Figura 53 | 64 |
| Figura 54 | 65 |
| Figura 55 | 65 |
| Figura 56 | 65 |
| Figura 57 | 66 |
| Figura 58 | 66 |
| Figura 59 | 67 |

Figura 60 69

Figura 61 70

Figura 62 72

ÍNDICE DE TABLAS

| | |
|----------------|----|
| Tabla 1 | 17 |
| Tabla 2 | 19 |
| Tabla 3 | 23 |
| Tabla 4 | 32 |
| Tabla 5 | 33 |
| Tabla 6 | 34 |
| Tabla 7 | 35 |
| Tabla 8 | 35 |
| Tabla 9 | 36 |
| Tabla 10 | 37 |
| Tabla 11 | 38 |
| Tabla 12 | 39 |
| Tabla 13 | 40 |
| Tabla 14 | 48 |
| Tabla 15 | 50 |
| Tabla 16 | 56 |
| Tabla 17 | 58 |
| Tabla 18 | 62 |
| Tabla 19 | 63 |

| | |
|----------------|----|
| Tabla 20 | 67 |
| Tabla 21 | 69 |
| Tabla 22 | 71 |

RESUMEN

El presente proyecto técnico tiene como objetivo la elaboración de un prototipo de reconocimiento de comandos de voz por medio del uso de hardware de microcontroladores ARM, especializado en el reconocimiento de voz y el desarrollo de su software con el cual permitir al usuario utilizar este tipo de tecnologías a su favor, con la fin de buscar la automatización de actividades o tareas enfocado a cualquier tipo de entorno o sector, ya sea laboral o doméstico.

El análisis del tipo de tecnología que se utilizará para el desarrollo del prototipo de reconocimiento será detallado, su información servirá de guía para su funcionamiento y para una adecuada implementación, además, se define cada componente utilizado en su elaboración, a su vez, la utilización de hardware (Placa Arduino) y software (IDE Arduino) de Arduino para su desarrollo, ya que este tipo de entorno de desarrollo es una herramienta de fácil manejo.

El resultado final es un prototipo de reconocimiento desarrollado en base a microcontroladores ARM que permitirá el reconocimiento de ordenes o comandos de voz, el cual analizará la información recibida y de acuerdo al proceso, realizará la acción esperada por el usuario, con el fin de generar resultados que cumplan las aspiraciones del usuario a través de una mínima intervención humana, promoviendo su uso a cualquier sector en el cual las actividades sean repetitivas, obteniendo la automatización de las tareas.

El desarrollo de tecnologías de reconocimiento de voz simplifica las tareas realizadas por humanos, aprovechando que las maquinas ejecuta mejores acciones.

Palabras clave Reconocimiento de voz, automatización, microcontroladores, Arduino.

ABSTRACT

The objective of this technical project is the development of a voice command recognition prototype through the use of ARM microcontroller hardware, specialized in voice recognition and the development of its software with which to allow the user to use this type of technologies in their favor, in order to seek the automation of activities or tasks focused on any type of environment or sector, whether at work or at home.

The analysis of the type of technology that will be used for the development of the recognition prototype will be detailed, its information will serve as a guide for its operation and for an adequate implementation, in addition, each component used in its elaboration is defined, in turn, the use hardware (Arduino Board) and Arduino software (Arduino IDE) for its development, since this type of development environment is an easy-to-use tool.

The final result is a recognition prototype developed based on ARM microcontrollers that will allow the recognition of orders or voice commands, which will analyze the information received and, according to the process, will perform the action expected by the user, in order to generate results that meet the user's aspirations through minimal human intervention, promoting its use in any sector in which activities are repetitive, obtaining the automation of tasks.

The development of voice recognition technologies simplifies the tasks performed by humans, taking advantage of the fact that machines perform better actions.

Keywords *Speech recognition, automation, microcontrollers, Arduino.*

INTRODUCCIÓN

Las tecnologías de reconocimiento de voz ayudan a que las actividades rutinarias de los usuarios se reduzcan otorgándole el control a una máquina, la cual se encargara de realizar la actividad indicada por medio de la voz de manera inmediata, suprimiendo la interacción habitual del usuario con dicha tarea.

La voz es el medio de comunicación que al ser emitida lleva la información que será procesada a través de señales, decodificándola y convirtiéndola en lenguaje entendible para la máquina, de modo que cumpla en ejecutar las órdenes que fueron emitidas por el usuario.

El desarrollo de estas tecnologías actualmente se encuentra en auge, debido a que, su uso beneficia a un amplio entorno de trabajo, de manera que se puede encontrar en el hogar con la domótica, asistentes de voz, en el área de salud, la automatización de maquinaria, robótica o empleado en la seguridad biométrica.

El uso de reconocimiento de voz ayuda en el desempeño del estilo de vida tecnológico en el que actualmente la humanidad se encuentra mediante comandos u ordenes de voz, asistiendo en el control de diversos actuadores o componentes, generando sistemas inteligentes.

ANTECEDENTES

El estudio elaborado por (SANTISTEVAN ALVARADO, 2021) tuvo como objetivo llevar a cabo un sistema de iluminación domótico en laboratorios de electrónica y robótica en la Universidad Estatal del Sur de Manabí utilizando reconocimiento de voz, la investigación se enfocó al área de domótica, donde se utilizó metodologías de investigación, analítico y descriptivo con el fin de examinar puntos relevantes para su proyecto como el conocimiento de equipos y su funcionamiento, además de los módulos y sensores necesarios en la elaboración de proyectos con estas tecnologías de reconocimiento de voz. Lo que dio como resultado un sistema de control inteligente, donde los estudiantes y docentes se benefician de ello, ya que tiene una actualización de luces controladas por voz. Lo que concluye que estudiantes muestran interés en el área de la domótica y a un futuro puedan investigar e incorporar estas tecnologías al ámbito profesional.

En cuanto (Varela Tapia & García Torres, 2020) su objeto de estudio consiste en una cama de hospital para personas con problemas de movilidad, por medio de un sistema electrónico donde las ordenes serán emitidas por comandos de voz, el objetivo es automatizar esta actividad por medio del uso de tecnologías como Arduino y comandos de voz, donde, la cama pueda cambiar posiciones diferentes según la comodidad del usuario o paciente, la metodología que se utilizó consistió en la investigación de herramientas tecnológicas que permitan el reconocimiento de voz, así como su funcionamiento e implementación. El proyecto concluye que, el prototipo de un sistema dirigido por comandos de voz para automatizar una cama hospitalaria tiene un uso fácil al usuario, donde personas que presentan problemas de movilidad se auxilien de tecnologías que automatizan estas tareas por medio de comandos de voz, cuando existe poco personal en el área de salud.

Para (Álvarez Puluchi & Cajas Fuela, 2016) su objetivo de estudio busca el desarrollo de un sistema de control del hogar por medio de comandos de reconocimiento de voz, su investigación detalla el análisis, diseño y la implementación del sistema domótico, concluyendo en el desarrollo de aplicaciones domóticas, como el control de encendido y apagado de luces, motorización de persianas, ajuste de temperatura, apertura de puertas y ventanas. Estas aplicaciones automatizan las tareas que se realizan en el hogar.

(GARNICA CARDENAS, 2018) su estudio propone el desarrollo de un dispositivo controlado por comandos de voz enfocados a personas con discapacidad motriz, la metodología utilizada en el desarrollo de este estudio consta de la elaboración por medio de un cronograma planeado en etapas en el cual se recopila información para elaborar el dispositivo de control desarrollando un prototipo. Concluyendo, el uso de estas tecnologías ayuda a una solución de problemas que personas con limitaciones físicas y dando confort a los usuarios ya que es de fácil manejo.

PROBLEMA

El uso de reconocimiento de voz se encuentra aplicado en diversos campos de desarrollo e investigación como seguridad, bienestar, salud, confort, en especial, enfocada a la domótica, debido a que su implementación favorece a un estilo de vida cómodo al usuario. Estas tecnologías son aplicadas a la automatización de tareas o actividades que por lo general los usuarios ejecutan de manera rutinaria. Los microcontroladores ARM (Advanced RISC Machine) conforman un conjunto de instrucciones simples, pero de alta velocidad de ejecución a bajo consumo, esta arquitectura es utilizada en la automatización (Dpto. Técnico de CAPEL, 2008). Su empleo va dirigido a nivel general ya que se encuentra presente en áreas desde el hogar hasta la industria, con lo cual, se pueden encontrar proyectos con reconocimiento de voz sencillos hasta proyectos más elaborados.

A nivel mundial, el desarrollo de tecnologías enfocadas en el reconocimiento de voz está en auge, ya que se ve reflejado el uso de este tipo de tecnologías como herramientas de apoyo. A partir del 2020, el crecimiento anual de tecnologías de reconocimiento de voz aplicadas a la domótica en el mundo será de un 26% , Estados Unidos y Canadá son los mercados en los que el crecimiento de la domótica está siendo más utilizado (HOGARTEC Expertos en Sistemas Domóticos).

En Latinoamérica el uso de tecnologías de reconocimiento de voz está siendo utilizada con mayor frecuencia en los campos de investigación, lo que demuestra una gran recepción de este tipo de tecnologías. Se estima que América Latina ocupa entre un 8% y 10% del mercado mundial referente al uso de tecnologías de automatización. Expertos en países de América Latina concuerdan que el sector de la automatización ha crecido, además mencionan que regiones como

el Caribe hay un gran mercado, así como en los países grandes de la región, entre ellos Brasil y México y otros que vienen creciendo de manera importante, como Colombia y Chile (Santa, 2021).

En Ecuador se ofertan sistemas de automatización desde 2005, en la actualidad es una actividad que gana terreno, el desarrollo de temas relacionados con el reconocimiento de voz ha dado un avance para que en un futuro la investigación de este tipo de tecnología permita su desarrollo e investigación, según estadísticas del Instituto Nacional de Estadística y Censos (INEC), el acceso y uso de la tecnología en Ecuador es del 60,7% (Flores, 2019).

El presente proyecto busca responder la siguiente problemática: ¿En qué medida ayudaría la elaboración de un prototipo de reconocimiento de voz utilizando microcontroladores?.

Los microcontroladores ARM han sido mayormente utilizados en el desarrollo de ambientes laborales y a nivel general debido a que su implementación ha dado un buen rendimiento para tareas de automatización puesto que su arquitectura trabaja bajo comandos de instrucciones simples, pero de alta velocidad de ejecución con bajo consumo. En la actualidad existen tareas contraladas automáticamente por máquinas ya que se busca la manera de omitir tareas rutinarias usando la tecnología, la automatización ha sido implementada en amplios entornos ya que beneficia el tiempo de respuesta, costos y mano de obra, debido que las maquinas tienen una mejor eficiencia en el trabajo.

Las tecnológicas aplicadas en el reconocimiento de voz han sido implementadas en varios países alrededor del mundo, estas siguen en una investigación constante, ya que, su uso genera sistemas inteligentes, facilitando y automatizando sus actividades reduciendo la carga de actividades que los usuarios generalmente se ven reflejados a realizar. Los proyectos de reconocimiento de voz abarcan un área de trabajo extenso ya que las mismas facilitan y benefician al usuario.

Entre las diferentes causas que pueden originar este problema, se ha detectado: i) obtención de datos, ii) desconocimiento al extraer información, iii) datos corrompidos, la causa primordial elegida fue la obtención de datos, debido a que el microcontrolador recibirá la información de entrada como señales análogas (comandos de voz), la cual se codificará en un patrón o plantilla de voz.

Los efectos que pueden traer este problema son diversos, entre los que se han detectado están: i) interpretación de datos, ii) ineficiencia, iii) malinterpretación de datos, con la cual se eligió como consecuencia a la interpretación de datos, ya que, el programa debe analizar los datos de entrada y decodificarla en una señal digital por medio del lenguaje de máquina desarrollado.

El presente proyecto pretende la elaboración de un prototipo que tiene como finalidad el reconocer comandos de voz aplicado automatizar una tarea ejecutando la acción solicitada por el usuario, con la intención de optimizar el tiempo que el usuario consume en realizar dicha actividad.

JUSTIFICACIÓN

El uso de reconocimiento de voz en los últimos años ha sido tendencia en la tecnológica que usualmente son utilizados para realizar actividades mediante comandos de voz con el fin de satisfacer las necesidades de las personas, el uso de estas tecnologías simplifica y reducen las actividades cotidianas de los usuarios.

Este tipo de tecnologías enfocadas al reconocimiento de voz tiene un incremento en su investigación, desarrollo y su utilización.

Algunos países han dedicado estudios enfocados a investigar y desarrollar proyectos que permiten este tipo de interacción, estos pueden ejecutar una orden la cual será interpretada por medio de comandos de voz.

Estas tecnologías se encuentran mundialmente desarrolladas ya que su uso beneficia no solo a una reducida área de trabajo pudiendo encontrarlas en el domótica, salud, seguridad biométrica, asistentes de voz, automatización de maquinaria, entre otros. La globalización de esta tecnología ha abierto las puertas a su investigación global ya que su beneficio económico es alto. A su vez la utilidad de implementar estas tecnologías es amplio ya que personas con algún tipo de inconveniente físico pueden realizar las tareas simples como por ejemplo encender un foco o actividades presentes en su vida, mediante comandos de voz.

Por lo tanto, se pretende implementar un módulo de reconocimiento de voz que permita reconocer los comandos y en base a ello realice una acción o tarea.

El proyecto servirá de apoyo a beneficiarios como: i) *directos*, investigadores que promueven la investigación y el desarrollo de nuevas tecnologías enfocadas al reconocimiento de voz; ii) *indirectos*, estudiantes de carreras técnicas que buscan información con el cual poder desarrollar prototipos con estas tecnologías.

OBJETIVOS GENERAL Y ESPECÍFICO

OBJETIVO GENERAL:

Elaborar un prototipo de reconocimiento de voz para automatización mediante microcontroladores ARM.

OBJETIVOS ESPECÍFICOS:

- Investigar estados del arte referente al área de trabajo.
- Analizar herramientas de hardware y software.
- Diseñar el prototipo de reconocimiento de voz.
- Desarrollar el software de reconocimiento de voz a través de la IDE de Arduino.
- Observar los resultados obtenidos y analizarlos.

METODOLOGÍA

El reconocimiento de comandos de voz por medio de un prototipo se realizará de manera independiente. Para el desarrollo de este proyecto se pretende ejecutarlo mediante la metodología de desarrollo ágil, ya que esta metodología divide las solicitudes en funciones y desarrolla de manera incremental, además colabora para entregar de forma iterativa cualquier trabajo, esta metodología ofrece retroalimentación ayudando a encontrar y corregir defectos a medida que se despliega la funcionalidad (IBM), se llevara a cabo las siguientes actividades:

- Desarrollo del prototipo de reconocimiento de voz
- Análisis de datos de entrada (comandos de voz) al módulo de reconocimiento
- Interpretación a lenguaje de máquina de los datos de entrada
- Generación de respuestas
- Resultados

CAPÍTULO 1

El primer capítulo abarca un análisis sobre la voz, su definición y uso en la tecnología, así como sus representaciones e interpretaciones en base a la tecnología de reconocimiento, a su vez, se detalla información sobre los componentes hardware, sus características técnicas y funcionamiento.

1. MARCO TEÓRICO/ESTADO DEL ARTE

El objetivo de estudio de (Salazar Aldaz, 2012) fue el de realizar un sistema que, en base al reconocimiento de voz, controle funciones multimedia básicas en el hogar como controlar la televisión y reproducir música, este estudio empleó una metodología de desarrollo incremental documentando el ciclo de etapas de progreso, a su vez analizó las tecnologías disponibles como hardware, sistemas operativos y lenguajes de programación, necesarios en el desarrollo de proyectos con estas características, además, menciona que las tecnologías de reconocimiento de voz tienen un enorme potencial en la automatización del hogar. El estudio concluyó que, el control de dispositivos multimedia del hogar por medio de tecnologías de comandos de voz es realizable ya que existe la facilidad de acceso tanto en software como en hardware.

En el trabajo “Prototipo de un sistema de control y monitoreo para hospitales mediante comandos de voz”, realizado por (CORTEZ MORALES & CARCELEN PIZA, 2018), su estudio tuvo como objetivo implementar un sistema de monitoreo y control utilizando comandos de reconocimiento de voz dentro del área de medicina, la metodología que utilizaron fue explicativa proporcionando investigación cualitativa, estadísticas y metodología de investigación, en el que se detalla la implementación de los diferentes sensores y módulos, además de desarrollar los comandos de reconocimiento de voz y el uso en el cual estos se llevarán a cabo. Concluyendo que el desarrollo de tecnologías con estas características favorece a una mejora en el desempeño ya

que se pueden omitir el realizar tareas repetitivas, en este caso el de nivelar las camas de hospitales, ya que antes el personal médico era el encargado de utilizar controles para ello, automatizando este tipo de tareas.

Mientras que, para (RUIZ SIERRA & VERGARA MARTINEZ, 2019) su estudio se enfocó en el diseño de reconocimiento de voz para el control de sillas de ruedas utilizando microcontrolador Arduino, obteniendo una señal que permita identificar y ejecutar la orden por medio de un micrófono que procesara esta información por medio de frecuencias entre el usuario y el sistema de reconocimiento de voz, la metodología que fue empleada consiste en realizar un análisis y estudio preliminar de los sistemas electrónicos como microcontroladores, módulos y lenguajes de programación. El estudio concluyó que, la implementación de un sistema con estas características facilitó la movilidad a personas con discapacidad física, incluso mencionan que se puede dar un uso en el mercado médico u otros ámbitos.

El uso de sistemas de navegación controlados por la voz busca mejorar la movilidad de personas con incapacidades físicas. El uso de redes neuronales y un estudio de características mecánicas fueron utilizadas para el desarrollo de sillas motorizadas (Farro Herrera & Villegas Malca, 2017).

1.1. ¿Qué es la voz?

Los seres humanos tienen la capacidad de producir sonidos articulados debido a su aparato fonético, el cual genera y amplía el sonido que se produce al momento de articular un sonido en específico. Su funcionamiento consiste en el cual, el aire que sale de los pulmones pasa por la tráquea y luego por la faringe, este aire presiona y hace vibrar las cuerdas vocales generando la voz.

Las características que definen la voz constan de elementos como intensidad, duración, timbre y tono.

- Intensidad: potencia en la que el sonido es emitido, su medición es en decibeles y su uso determina los umbrales auditivos.
- Duración: tiempo que el sonido se tarda en producir, su uso determina la velocidad en la que las palabras son emitidas.
- Timbre: permite diferenciar voces con similitud de intensidad y tono.
- Tono: cantidad de vibraciones, frecuencia de sonido producida por una onda acústica (Apolo Díaz & Coba Castillo, 2017).

1.2. ¿Qué es digitalizar la voz?

La voz humana es recopilada a través de un micrófono, convirtiéndola en señales eléctricas, estas señales son digitalizadas mediante el uso de un conversor, de forma resumida, es la acción de convertir en digital (valores numéricos) la información de entrada de señal continua (señal analógica), ya que el computador procesa únicamente señales digitales, generalmente en lenguaje binario. Esta señal digitalizada puede ser manipulada mediante el procesamiento de la señal de voz, la conversión de la señal analógica a digital consta de dos etapas, muestreo (número de muestras recolectadas de la señal de entrada) y la codificación (valor otorgado a las muestras), en la actualidad la tecnología para la conversión de señales analógicas-digitales ha dado grandes avances con los cuales se pueden obtener información y resultados más precisos.

1.3. Como se graba la voz para procesarla

La voz por medio del micrófono se envía como señal analógica a la computadora, el software de reconocimiento de voz, configurado en el idioma que se esté utilizando, convierte esa

señal analógica en una señal digital y la procesa, luego se eligen las palabras que se adaptan mejor a los que se dijo.

1.4. Procesamiento de la señal de voz

El procesamiento digital forma parte de un conjunto de técnicas matemáticas utilizadas en el uso de señales digitales. Este proceso se realiza por medio de hardware digital el cual debe ser provisto por el software.

El reconocimiento de voz es un proceso por el cual un sistema electrónico interactúa con el usuario por medio del habla, lo que permite reconocer lo que se está hablando con el fin de realizar una acción o tarea. Por medio de un micrófono la voz humana es convertida de señales acústicas a señales eléctricas, estas señales se digitalizan a través de un conversor sigma-delta de 16 bits o 32 bits de resolución o a su vez usando una grabadora digital.

El software de reconocimiento de voz requiere de un proceso conocido como conversión analógico-digital, en el cual el audio analógico se convierte en señales digitales (Beltrán Vásquez, 2003).

1.5. Grabación digital

La voz, que es una señal analógica, se somete a las etapas de muestreo y se transforma en lenguaje binario, estos números serán los que se guarden posteriormente.

1.6. Conversión analógico-digital

La conversión A/D (analógico-digital) consiste en realizar de forma periódica medidas de amplitud de una señal, para el caso del sonido registra la tensión que proviene del micrófono, redondea los valores a un conjunto finito de niveles, conocido como cuantificación, y los registra como números enteros.

Existen tres etapas para convertir la señal analógica en digital, estas etapas son muestreo, cuantificación y codificación.

1.7. Muestreo

Consiste en tomar muestras de la amplitud de onda de manera periódica por medio de la frecuencia de muestreo, esta frecuencia consiste en la velocidad con la que se toma la muestra, es decir, el número de muestras por segundo.

1.8. Cuantificación

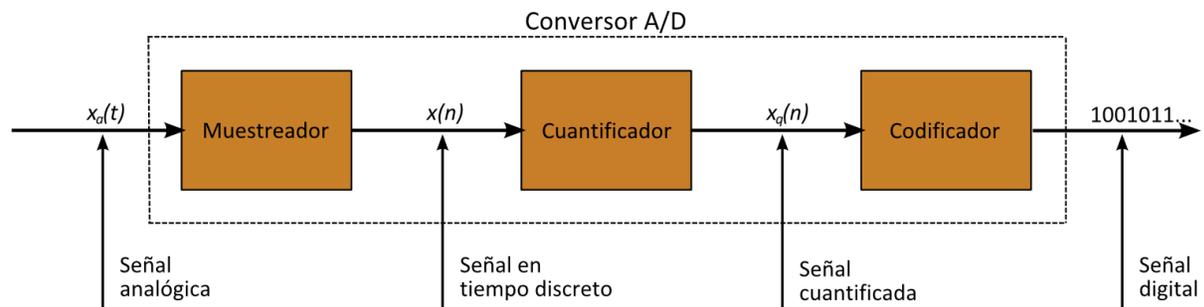
Mide el nivel de voltaje de cada muestra, asigna un valor de una señal analizada a un nivel de salida.

1.9. Codificación

Traduce los valores obtenidos de la cuantificación a código binario.

Figura 1

Conversión Analógico-Digital



Nota. Proceso de conversión A/D. Fuente: (COMEBACKSTUDIOS, 2022)

1.10. ¿Cómo funciona el reconocimiento de voz?

El software de reconocimiento de voz necesita que el audio analógico se transforme en señales digitales, proceso que se conoce como conversión analógico-digital, para que el software descifre con precisión la señal se requiere de una base digital de datos de vocabulario, palabras, sílabas y oraciones, además de un sistema de comparación de datos de señales. Se compara los

patrones de habla almacenadas en memoria con la información recibida por el conversor analógico-digital, lo que se conoce como reconocimiento de patrones.

1.11. Comparación de patrones

El reconocimiento de voz usa distintos métodos con los cuales reconocer ordenes, una de las técnicas básicas es la comparación de patrones, donde la señal es parametrizada, la secuencia de vectores extraída se almacena y se etiqueta como patrón de referencia de cada palabra. Para la fase de reconocimiento, la palabra parametrizada pronunciada por el hablante es comparada con todos los patrones de referencia utilizando una medida de similitud espectral. La fase de decisión se apoya en determinar el patrón con más similitud a la palabra proporcionada, es decir, la menor distancia en la etapa de comparación (Hernando Pericas, 1993).

1.12. Microcontroladores ARM

Advanced RISC Machine por sus siglas en inglés ARM o Máquina Avanzada de Conjunto de Instrucciones Reducido, hace referencia a una arquitectura de microprocesadores que emplea un conjunto reducido de instrucciones volviéndose más eficiente en el consumo de energía.

Su arquitectura tiene un conjunto de instrucciones simples pero eficiente ofreciendo una alta velocidad de ejecución con bajo consumo. Cuentan con una amplia gama de herramientas de desarrollo, su arquitectura trabaja bajo comandos de 32-bits (Dpto. Técnico de CAPEL, 2008).

El microcontrolador es un sistema que dentro de un circuito integrado contiene una unidad de procesamiento, memoria, pines de entrada-salida de datos, similar a un computador de tamaño reducido, en la actualidad existen una gran variedad de microcontroladores con características propias desarrolladas por sus fabricantes, también existen variedad en arquitectura, lenguaje de programación y el tipo de funcionalidad (Panta Martínez, 2012).

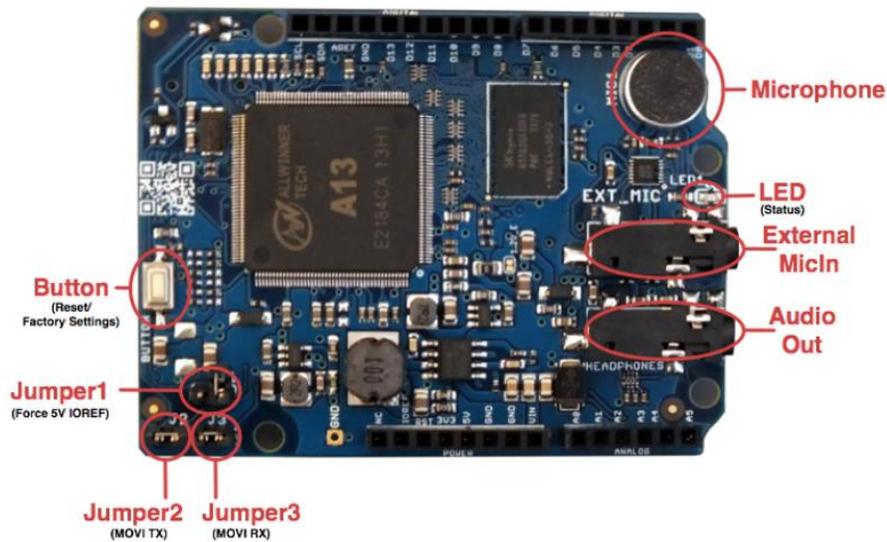
1.13. Tarjeta Audeme Movi Shield

El módulo de reconocimiento de voz MOVI por sus siglas en inglés My Own Voice Interface, desarrollado por Audeme recurrió a Kickstarter, el cuál es una corporación que patrocina y contribuye en proyectos y procesos creativos, para su lanzamiento. Este módulo trabaja en Debian sobre una placa Allwinner A13, MOVI brinda síntesis y reconocimiento de voz de hasta 200 oraciones, independiente del hablante MOVI no tiene un entrenamiento de muestras de voz involucrado, su diccionario integrado de 2GB permite el reconocimiento de cualquier oración, así como también crear nuestro propio vocabulario, su programación se la realiza desde la IDE de Arduino, por lo que no necesita conexión a internet, lo que la hace fuera de línea, además no necesita del almacenamiento en la nube para su funcionamiento.

El módulo de reconocimiento detecta los comandos de voz hasta 3 metros de distancia, MOVI consta de un micrófono integrado, además, posee un conector que admite micrófonos externos, independiente del hablante responderá a las oraciones sin importar quién las esté diciendo, posee una ranura SD que tiene almacenado la compilación de Debian, las oraciones, datos de configuración y el comando de llamada, incluida en la tarjeta SD se encuentran paquetes de código abierto (Brown, 2015).

Figura 2

Componentes de Movi Arduino Shield desarrollado por Audeme



Nota. Componentes de Movi Arduino Shield. Fuente: (AUDEME, 2017)

Tabla 1
Especificaciones técnicas Movi Arduino Shield

| | |
|--------------------------------|--|
| Placa Arduino | UNO R3, Mega 2560, otros |
| Fuente de alimentación externa | 7V – 16V 500 mA (alimenta a Movi a través de Arduino) |
| Auriculares/Altavoces | Conectado a la toma jack permite escuchar el sintetizador de voz |
| Micrófono interno | Incorporado en la placa para detectar la voz hasta 3 metros (10 pies) en ambientes silenciosos |
| Micrófono externo | Opcional, conectado a la entrada MIC cuando el entorno sea ruidoso |
| Soporte de idiomas | Inglés, español, alemán |
| Programación | Arduino IDE, otros |

Nota. Características técnicas Movi Arduino Shield. Elaborado por: Bryan Velasco M.

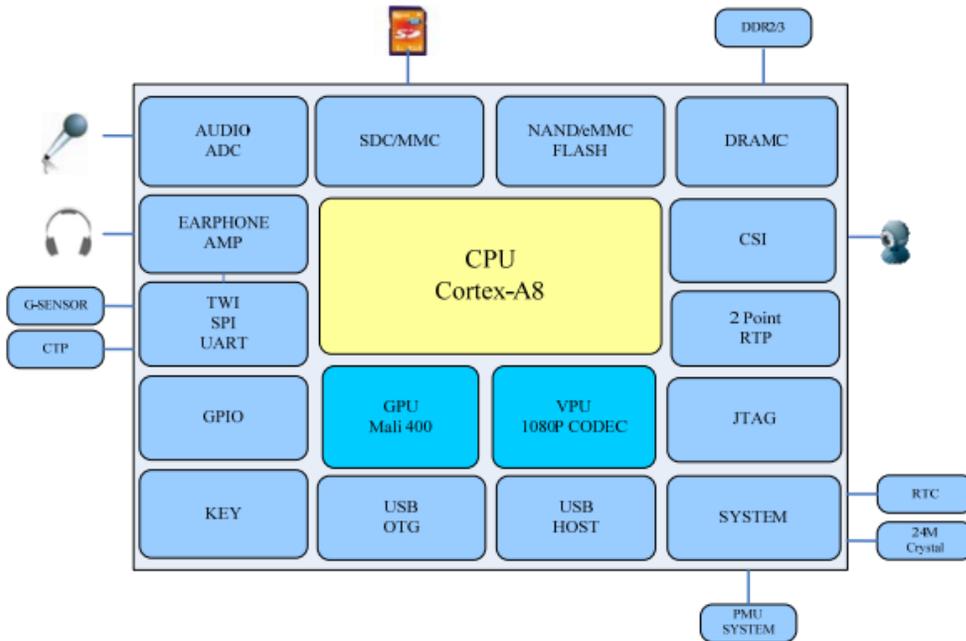
1.14. Allwinner A13

Allwinner A13 es un SoC por sus siglas en inglés (System on a chip), es decir, un sistema en un chip, donde convergen una gran cantidad de componentes que en conjunto conforman un sistema completo, A13 posee un CPU ARM Cortex A8 integrada GPU ARM Mali 400 y VPU Allwinner Cedar Engine, que ahora también se llama R8. El R8 es un cambio de nombre de A13 y está dirigido principalmente a dispositivos IoT de bajo presupuesto, por su capacidad para

desarrollar multimedia y procesamiento óptimo, tiene un menor consumo de energía y tiene una alta integración (Allwinner Technology CO., Ltd., 2012).

Figura 3

Diagrama en bloques del chip Allwinner A13



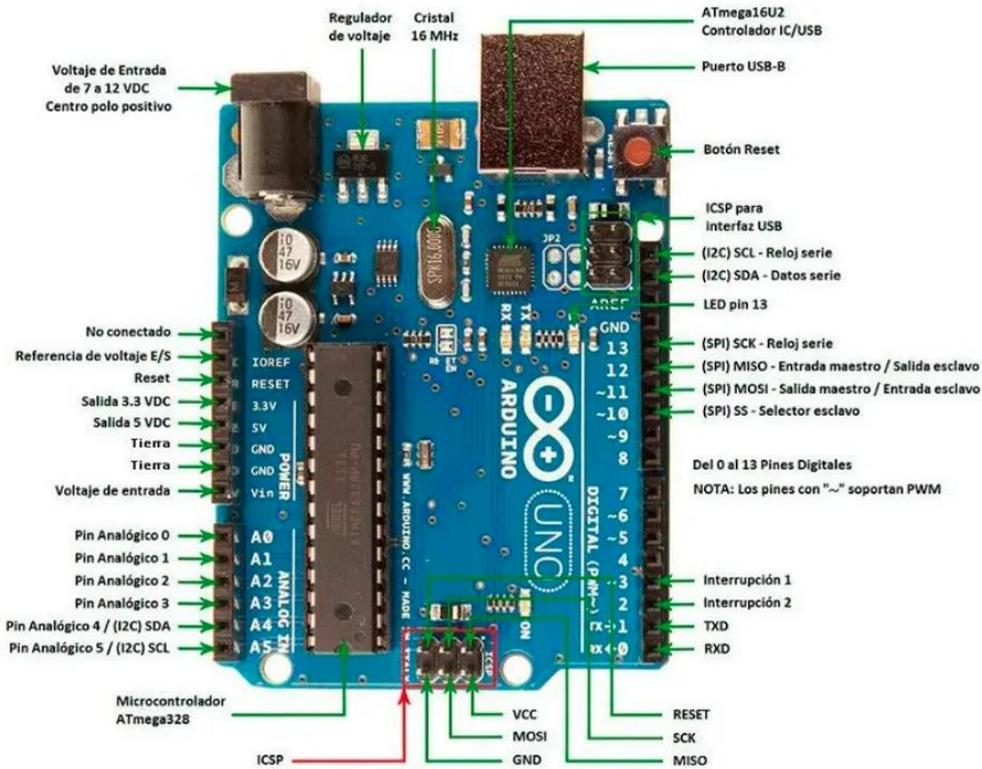
Nota. Diagrama en bloques del chip A13. Fuente: (Allwinner Technology CO., Ltd., 2012)

1.15. *Arduino UNO R3*

Es una placa de desarrollo basada en el microcontrolador ATmega328P, dentro de la extensa familia de placas Arduino Uno es una de las más conocidas y usadas, siendo una mejora de un diseño anterior (Arduino Duemilanove). Arduino funciona en plataforma abierta, además, al ser multiplataforma resulta funcional en los sistemas operativos que existen en el mercado, además, Arduino posee un software propio para su programación que ocupa un lenguaje de alto nivel (SOLEDISPA PILLAJO, 2022).

Figura 4

Componentes de un Arduino UNO



Nota. Componentes de un Arduino. Fuente: (Lukianova, 2021)

Tabla 2

Especificaciones técnicas de Arduino Uno

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

Nota. Características técnicas Arduino UNO. Elaborado por: Bryan Velasco M.

1.16. Automatización

La automatización mejora la eficiencia generando resultados más rápidos, consiste en utilizar tecnología para realizar tareas con una mínima intervención humana, puede ser implementando en cualquier sector en el que se llevan a cabo tareas repetitivas (Red Hat, 2022).

La automatización cambia la forma con las personas trabajan y viven, optimizando la eficiencia en tareas tanto en el hogar como en el ámbito empresarial, asiste en el uso de tecnologías para crear entornos inteligentes con el fin de mejorar las operaciones y la eficiencia.

CAPÍTULO 2

En el segundo capítulo se realiza el diseño y desarrollo del prototipo, especificando el hardware y software, así como detallado cada componente que será utilizado para la elaboración del prototipo. El capítulo también describe las herramientas empleadas para su funcionamiento.

2. DISEÑO Y DESARROLLO

2.1.DISEÑO

2.1.1.1.Ensamblaje de la tarjeta Movi con la placa Arduino

La placa de reconocimiento Movi es compatible con Arduino Uno, su ensamblaje es directo con la placa Arduino por medio de la colocación de sus pines, Movi se comunica a través de los pines digitales 10 y 11 con la placa Arduino, no obstante las versiones anteriores de Arduino UNO no cuentan con el PIN de comunicación IOREF.

Figura 5

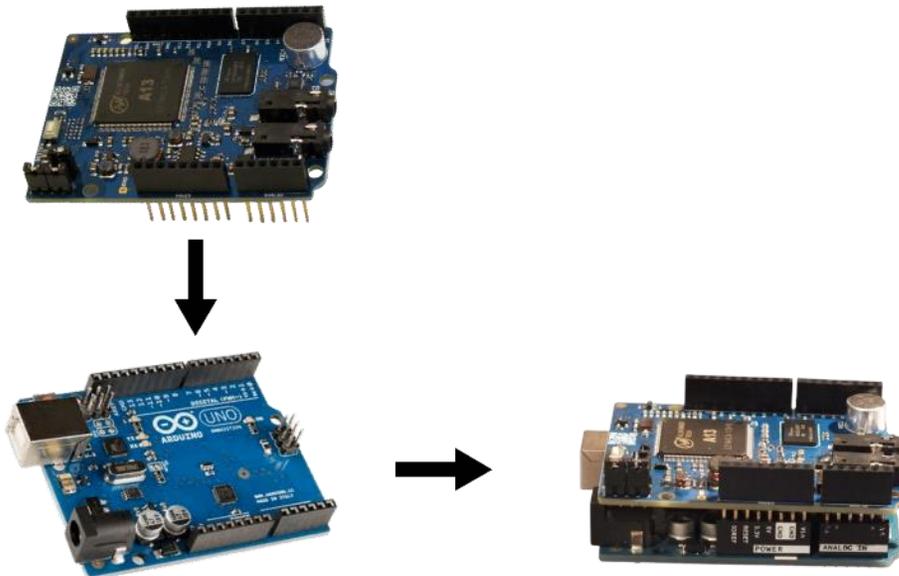
Cabecera Arduino Uno



Nota. Cabecera de Arduino UNO PIN IOREF. Fuente: (AUDEME, 2017)

Figura 6

Ensamblaje Movi Arduino



Nota. Ensamblaje Movi Arduino Shield. Elaborado por: Bryan Velasco M.

2.1.1.2. Audeme Movi Speech Recognizer and Synthesizer Shield for Arduino

Según los desarrolladores (AUDEME, 2017), MOVI es compatible con la mayoría de las placas Arduino, además, se puede usar con cualquier sistema con una interfaz serial a través de una API de código abierto. El reconocedor de voz de MOVI puede aprender hasta 150 oraciones completas. Se programa desde el IDE de Arduino y no requiere de algún software adicional ni muestra de voz para el entrenamiento. El procesamiento de Movi se realiza por medio de un ARM de 1Ghz.

(Brown, 2015), menciona que el dispositivo MOVI se comunica a través de los pines 10 y 11 con Arduino y debe alimentarse por medio una fuente de alimentación externa ya que el puerto USB no suministra suficiente corriente, cuando Movi se encuentra en funcionamiento se podrá observar encendida una luz led indicadora de estado, además, por medio de una altavoz se podrá escuchar a Movi inicializando cuando su sintetizador de voz enuncie “Movi is booting”.

La tarjeta Movi utilizara los siguientes pines para su funcionamiento:

Tabla 3

Pines utilizados por MOVI

| PIN | Detalle |
|-----------|---|
| VIN y GND | Consumo de energía |
| D10 y D11 | Comunicación en serie |
| IOREF | Determina el voltaje para la comunicación |

Nota. Pines utilizados por MOVI. Elaborado por: Bryan Velasco M.

En detalle, el funcionamiento de cada pin consiste en:

- Fuente de alimentación: debe haber suficiente corriente en VIN para alimentar a MOVI de 5.20V a 16V, en la mayoría de los casos se requiere una fuente externa de alimentación ya que por medio de USB se tiene una corriente por debajo de los 5V.
- Pin D10 y D11: deben estar libres para su comunicación en serie.
- Pin IOREF: indica el voltaje utilizados para la comunicación en serie.

2.1.1.3.Análisis hardware

Audeme Movi al ser una herramienta publicada recientemente no está extensa de retos, según los fabricantes esta plataforma está dirigida aquellos que desean ingresar al desarrollo de software y hardware en búsqueda de desarrollar dispositivos inteligentes, pero la incorporación de comandos de voz en este tipo de proyectos no está exenta de desafíos. MOVI es una unidad independiente de sintetizador y reconocimiento de voz con capacidad para oraciones completas que tiene como objetivo simplificar tareas, además según Audeme, quienes son los fabricantes, mencionan que es fácil de manipular y programar.

2.1.1.4.Diagrama de bloques del sistema de reconocimiento de voz

El sistema de reconocimiento de voz puede resumirse con el siguiente diagrama de bloques, donde, la tarjeta Movi por medio de su micrófono integrado recibe y procesa el comando de voz, Movi trabaja automáticamente el reconocimiento debido a que contiene una librería con palabras comúnmente utilizadas con las que realiza una comparación, obteniendo así la detección del comando, seguido de la programación en la IDE de Arduino, se ejecutará la orden señalada en las líneas de código, dando como resultado la acción requerida por el usuario.

Figura 7

Diagrama en bloques reconocimiento de voz



Nota. Diagrama en bloques reconocimiento de voz. Elaborado por: Bryan Velasco M.

El funcionamiento de la tarjeta Movi consiste en que, a través de la IDE de Arduino se programan las ordenes que deben ser reconocidas a través del micrófono, una vez cargado y compilado el programa, Movi indica que está listo para su función de reconocimiento, la tarjeta tiene un método de alertar al usuario a través de un zumbido o silbidos con los cuales lograr informar del estado al usuario.

2.1.1.5.Reconocimiento MOVI

MOVI posee un micrófono integrado con el cual consigue el reconocimiento por medio de una onda que entra por el micrófono dividiéndose en regiones de habla y no habla.

Las regiones de habla pasan a un clasificador, descompone la forma de onda en posibles secuencias de fonemas donde se favorece a los fonemas con mayor probabilidad con los cuales

poder reconocer las palabras que forman parte de la oración, donde se asigna las palabras a las oraciones más probables.

La librería de Movi envía oraciones de entrenamiento en forma de texto a la tarjeta, las palabras de cada oración son pronunciadas por la tarjeta por medio del uso de un diccionario incorporado de 2GB, que conoce reglas gramaticales. Utiliza una secuencia de fonemas con las cuales crea un modelo temporal que reconoce las palabras que fueron parte de las oraciones de entrenamiento, un segundo modelo temporal favorece secuencias de palabras que son parte de las oraciones sobre secuencias que no lo son, asignando una mayor probabilidad a las secuencias de fonemas que formaron parte de las oraciones de entrenamiento.

Una forma de onda entra por el micrófono a las regiones de habla y no habla, mediante un algoritmo monitorea la señal entrante durante un periodo corto de tiempo y lo compara con un umbral.

Las regiones del habla de la señal pasan a un clasificador, descomponiendo la forma de onda en secuencias de fonemas, por medio del modelo temporal del entrenamiento las secuencias de fonemas se emparejan con las palabras entrenadas.

Por último, una fase de corrección asigna las palabras al conjunto de oraciones más probables (AUDEME, 2017).

2.1.1.6. Análisis software

La IDE Arduino es un entorno de multiplataforma de programación basado en Java, posee un editor de código de recursos como realce de sintaxis, paréntesis e identidad automática, es capaz de compilar y cargar programas en la placa.

Se define dos funciones para lograr un programa funcional como:

setup(): se inserta en el inicio, su utiliza para inicializar configuración.

loop(): llamado con el cual se repite un bloque de comandos.

Figura 8

IDE Arduino (Entorno de desarrollo integrado)

```
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

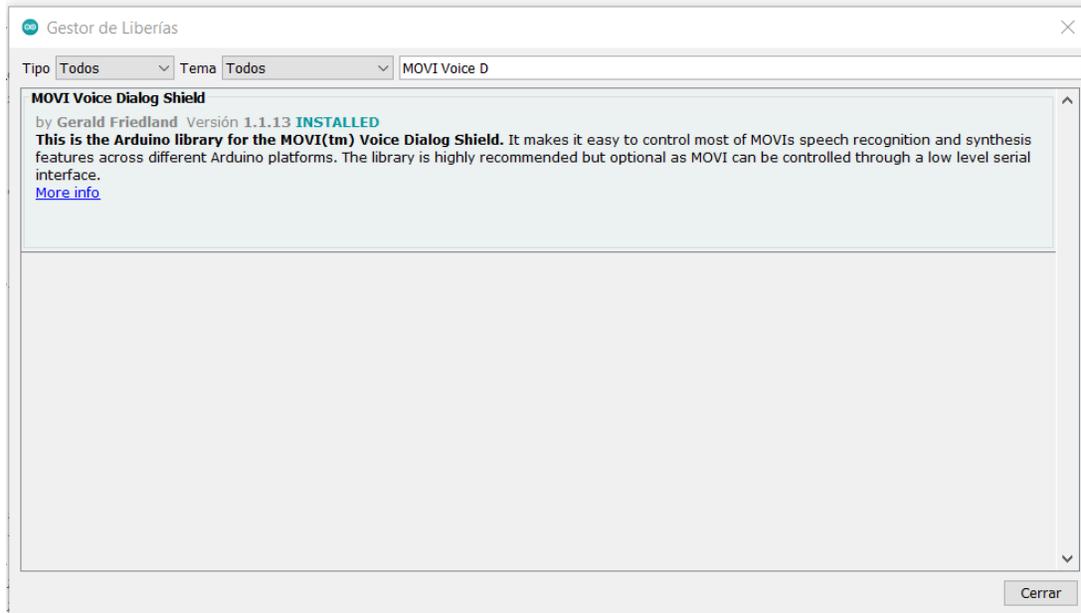
Nota. IDE Arduino (Entorno de desarrollo integrado). Elaborado por: Bryan Velasco M.

2.1.1.7.Librería Movi

Movi posee una librería open source descargable desde su página web oficial, así como descarga directa desde las librerías de IDE Arduino.

Figura 9

Librería MOVI



Nota. Librería MOVI en el Gestor de librerías de IDE Arduino. Elaborado por: Bryan Velasco M.

2.1.1.8. Proceso de grabado

Movi cuenta con un diccionario interno de 2GB, lo que permite que los usuarios puedan programar cualquier oración, así como, cambiar su vocabulario creando sus propios diálogos, ya que no requiere de entrenamiento ni muestras de voz.

El método de llamada a Movi se lo realiza con la siguiente línea de código:

Figura 10

Llamada de Movi

```
recognizer.callSign("Arduino"); // Train callsign Arduino (may take 20 seconds)
```

Nota. Señal de llamada a Movi. Elaborador por: Bryan Velasco M.

El método de llamado para la tarjeta puede ser configurado dentro de Arduino con diferentes nombres como "Computer" o "Movi", la señal de llamada que por lo general es utilizada es "Arduino".

Al momento de agregar oraciones a la tarjeta Movi, se lo realiza a través de la interfaz de programación de Arduino con las siguientes líneas de código:

Figura 11

Oraciones de Movi

```
recognizer.addSentence("Sentence 1"); // Add sentence 1
recognizer.train(); // Train (may take 20seconds)
```

Nota. Línea de código para oraciones de Movi. Elaborado por: Bryan Velasco M.

Una vez subido y compilado las líneas de código a Movi, se consigue que la tarjeta haya aprendido la oración.

2.1.1.9. Proceso de ejecución

Incluimos la librería de Movi:

Figura 12

Librería Movi

```
#include "MOVIShield.h" //
```

Nota. Librería Movi. Elaborado por: Bryan Velasco M.

Esta línea de código habilita los pines de comunicación de Movi:

Figura 13

Comunicación con Movi

```
MOVI recognizer(true); //
```

Nota. Línea de comunicación con Movi. Elaborado por: Bryan Velasco M.

Este método espera a que Movi se inicie y reinicia ajustes:

Figura 14

Inicialización de Movi

```
recognizer.init(); // Initialize MOVI (waits for it to boot)
```

Nota. Método de inicialización Movi. Elaborador por: Bryan Velasco M.

La configuración del umbral ayuda a Movi reconocer ordenes en ambientes ruidosos:

Figura 15

Umbral de Movi

```
recognizer.setThreshold(5); // set to a higher value (valid range 2-95) if you have a problems due to a noisy environment.
```

Nota. Umbral de Movi. Elaborado por: Bryan Velasco M.

Funcionamiento del algoritmo

El método `MOVI::loop()` extraído de la librería `MOVI`, indica una serie de eventos con los cuales se logran obtener los resultados.

La función “`poll()`” provee a las aplicaciones un mecanismo para realizar operaciones de entrada/salida sobre un conjunto de descriptors de archivos examinando cada evento, `poll()` espera eventos dentro del descriptor de archivos.

Figura 16

Método `poll()` de la librería Movi

```

signed int MOVI::poll()
{
    firstsentence=false; // We assume loop() and we can't train in loop.
    intraining=false;

    int curchar;
    int eventno;
    while (mySerial->available()) {
        curchar=mySerial->read();
        if (curchar=='\n') {
            if (debug) {
                Serial.println(response);
            }
            if (response.lastIndexOf("MOVIEvent[")>=0) { // Dylan-suggested fix that makes sure buffer junk is not interpreted
                eventno=response.substring(response.indexOf("[")+1,response.indexOf("):").toInt();
                result=response.substring(response.indexOf("")+1);
                if (eventno<100) { // then it's a user-read-only event
                    response="";
                    return SHIELD_IDLE;
                }
                if (eventno==202) {
                    result=response.substring(response.indexOf("#")+1);
                    response="";
                    return result.toInt()+1; // Sentences returned start at 0,
                    // we make it easier for non-programmers and start at 1.
                }
                if (eventno==203) { // this is a password event
                    response="";
                    result.trim();
                    if (passstring.equals(result)) {
                        return PASSWORD_ACCEPT;
                    } else {
                        return PASSWORD_REJECT;
                    }
                }
                response="";
                return -eventno;
            } else {
                // other jibberish not belonging to MOVI
            }
        } else {
            response+=(char) curchar;
        }
    }

    if (debug) {
        while (Serial.available()) {
            mySerial->write(Serial.read());
        }
    }

    return SHIELD_IDLE;
}

```

Nota. Método poll() obtenido de librería Movi. Elaborado por: Bryan Velasco M.

A través del modelo temporal creado en la parte del entrenamiento, se comparan las secuencias de fonemas con las palabras previamente entrenadas, a su vez favorece a las secuencias

de palabras que forman parte de las oraciones de entrenamiento. Luego, se asigna las palabras a la oración más probable dentro de un conjunto de entrenamiento (resultado de poll()).

Figura 17

Variable asignada al resultado

```
signed int res = recognizer.poll(); // Get result from MOVI, 0 denotes nothing happened, negative values denote events (see docs)
```

Nota. Variable asignada a método de resultado. Elaborado por: Bryan Velasco M.

La función de agregado de oraciones extraído de la librería MOVI, indica que la condición “if (firstsentence)” verificará si la variable “firstsentence” se encuentra vacía o no, si el campo se encuentra lleno entrará al condicional “if”.

La variable “intraining” almacena el resultado de la función “sendCommand”, volviendo a su estado “false” a la variable con el fin de no ingresar nuevamente al “if”.

Figura 18

Función agregación de oraciones

```
bool MOVI::addSentence(const __FlashStringHelper* sentence)
{
    if (firstsentence) {
        intraining=sendCommand(F("NEWSENTENCES"),F(""),"210");
        firstsentence=false;
    }
    if (!intraining) return false;
    intraining=sendCommand(F("ADDSSENTENCE"),sentence,"211");
    return intraining;
}

bool MOVI::addSentence(String sentence)
{
    // Needs a new MOVI instance (typically restart Arduino). This avoids training only part of the sentence set.
    if (firstsentence) {
        intraining=sendCommand("NEWSENTENCES","", "210");
        firstsentence=false;
    }
    if (!intraining) return false;
    intraining=sendCommand("ADDSSENTENCE",sentence,"211");
    return intraining;
}
```

Nota. Funciones de agregado de oraciones de MOVI. Elaborado por: Bryan Velasco M.

La función “if (!intraining) return false”, indica si la variable “intraining” está vacía retornando un false, es decir, no se ha hecho ningún entrenamiento ya que no entro a la condición “if” de la sentencia de código de arriba, por lo cual, no hace la función "sendCommand" y retorna un false porque no se hizo nada. Si el condicional “if (firstsentence)” se encuentra vacía y no entra a la primera condición “if”, es decir, no existe una oración, guarda en la variable “intraining” lo que la función “sendCommand” le envíe.

2.2.Desarrollo del prototipo

El desarrollo del prototipo de reconocimiento de voz posee componentes electrónicos, los cuales serán detallados a continuación, que servirán para pruebas de funcionalidad, esto aplicado a un escenario de demostración, no obstante, este escenario puede ser aplicado a las actividades o situaciones reales de la vida cotidiana con el fin de obtener sistemas inteligentes que ayuden al usuario.

2.2.1. Anillo LED RGB

LED RGB en forma de anillo compatible con Arduino, cada luz puede ser programada individualmente, se tiene el control de cada pixel y color por medio de un solo cable. Se basa en el chip controlador WS2812 elaborado por Neopixel, se puede encadenar a otros led conectando su salida a la a entrada de otro, obtenido una cadena larga de luces.

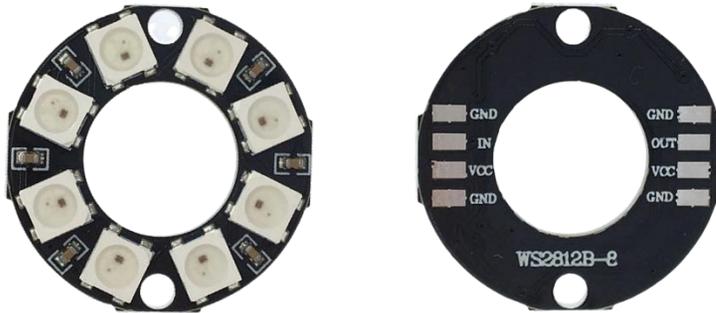
Tabla 4
Pines del LED RGB

| | |
|----------|-------------------------------------|
| GND | Alimentación de bajo nivel (tierra) |
| VCC | Alimentación de alto nivel |
| DIN (IN) | Recibe información del color |
| DO (OUT) | Entrega información del color |

Nota. Pines que posee cada Led. Elaborado por: Bryan Velasco M.

Figura 19

Anillo LED RGB WS2812B-8bits



Nota. Anillo Led RGB 8bits. Elaborado por: Bryan Velasco M.

Tabla 5

Características Led WS2812

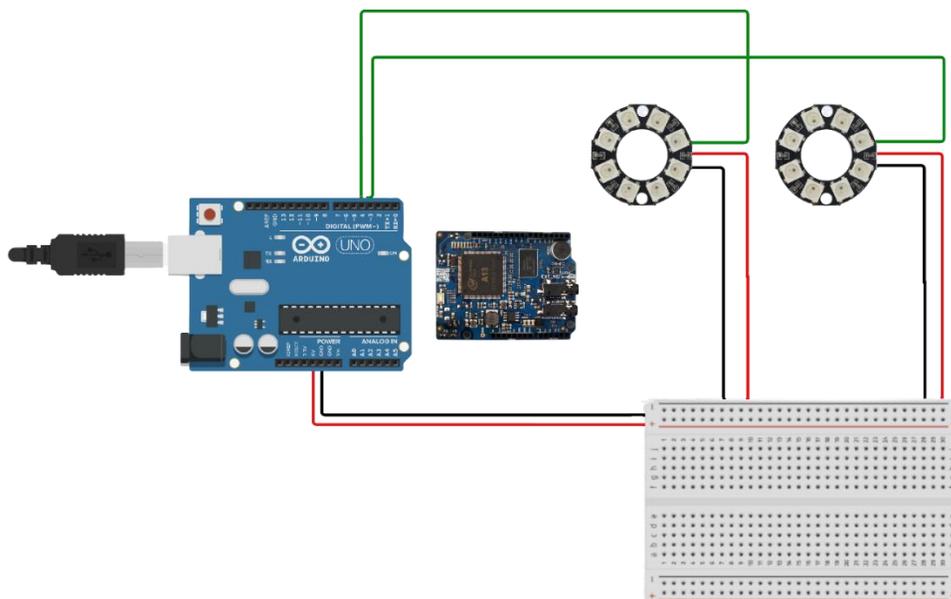
| | |
|---------------------------|---------------|
| Modelo LED | WS2812 |
| Voltaje | CC 4 – 7V |
| Alimentación Arduino | 5V |
| Consumo de corriente a 5V | 60 mA por LED |
| Comunicación | 1 cable (IN) |

Nota. Características técnicas anillo LED RGB. Elaborado por: Bryan Velasco M.

2.2.2. Modelado del circuito

Figura 20

Modelado LED RGB



Nota. Modelado luces RGB. Elaborado por: Bryan Velasco M.

En la Figura 20, se observa la conexión de las luces LED RGB con la placa de reconocimiento, está colocada a un lado de Arduino con el propósito de observar los pines asignados.

Tabla 6
Pines asignados a Led

| PIN | Detalle |
|-----|---------|
| 3 | LED RGB |
| 4 | LED RGB |
| 5 | LED RGB |
| 6 | LED RGB |

Nota. Pines asignados a LED RGB. Elaborado por: Bryan Velasco M.

2.2.3. Motor reductor de doble eje

El motor DC es un mecanismo capaz de regular la velocidad de giro de la máquina haciéndola funcionar a un ritmo determinado, se componen de engranajes que conforman una cadena cinemática, lo que hace funcionar piezas giratorias.

Figura 21

Motor reductor doble eje



Nota. Motor reductor doble eje. Elaborado por: Bryan Velasco M.

Tabla 7

Características

| | |
|-----------------------------|--------|
| Voltaje de operación | 3 – 6V |
| Consumo de corriente máxima | 150 mA |

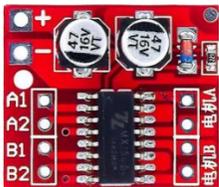
Nota. Características técnicas motor doble eje. Elaborado por: Bryan Velasco M.

2.2.4. Controlador puente L298n

La placa controladora de motor de corriente continua DC permite manipular el sentido de giro, la placa controlador permite el uso de hasta dos motores.

Figura 22

Driver L298n



Nota. Mini driver L298n. Elaborado por: Bryan Velasco M.

Tabla 8

Características

| | |
|----------------------|---------|
| Voltaje de operación | 3 – 10V |
|----------------------|---------|

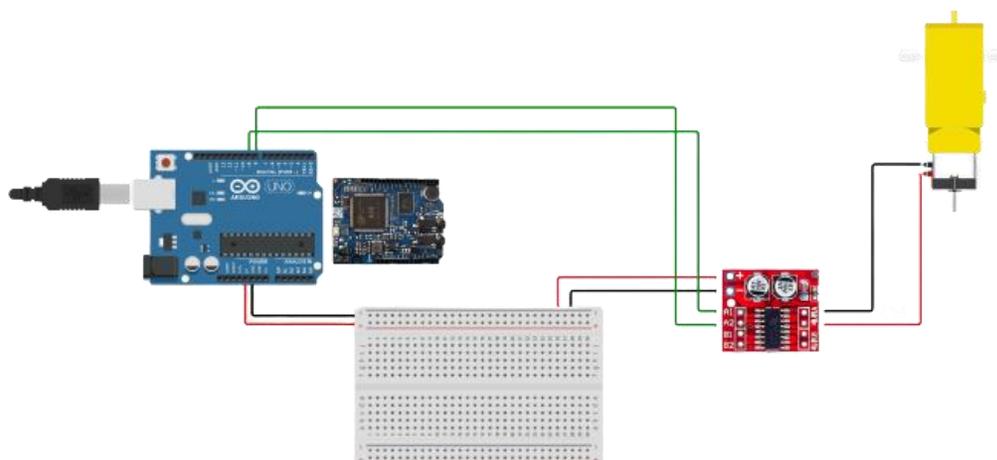
| | |
|-----------------------------|----------|
| Voltaje de entrada | 1.8 – 7V |
| Corriente máxima | 2.5 A |
| Corriente de trabajo simple | 1.5 A |

Nota. Características técnicas driver L298n. Elaborado por: Bryan Velasco M.

2.2.5. Modelado del circuito

Figura 23

Modelado motor doble eje



Nota. Modelado motor doble eje. Elaborado por: Bryan Velasco M.

En la figura 23, se observa la conexión del driver controlador de giro y el motor reductor con la placa de reconocimiento, está colocada a un lado de Arduino para observa los pines que fueron asignados.

Tabla 9

Pines asignados a motor reductor

| PIN | Detalle |
|-----|----------------|
| 8 | Motor reductor |
| 9 | Motor reductor |

Nota. Pines asignados a Motor reductor. Elaborado por: Bryan Velasco M.

2.2.6. Ventilador DC

El ventilador axial permite el flujo de aire al dispositivo de refrigeración, es utilizado comúnmente para enfriar y ventilar evitando sobrecalentamientos.

Figura 24

Ventilador DC



Nota. Ventilador DC. Elaborado por: Bryan Velasco M.

Tabla 10

Características

| | |
|---------------------|------------|
| Voltaje nominal | 5 – 12V |
| Corriente nominal | 0.09 A |
| Velocidad de viento | 1.5 m/s |
| Dimensiones | 40x40x10mm |

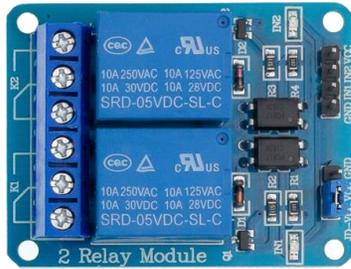
Nota. Características técnicas ventilador DC. Elaborado por: Bryan Velasco M.

2.2.7. Modulo relé

El módulo relé es utilizado para controlar el encendido o apagado de equipos con altas cargas de voltajes o equipos con alta potencia de amperaje, consta de una aislamiento electrónico, activa una salida normalmente abierta (NO) al recibir un 0 voltios (“0” lógico) y desactiva la salida con un 5 voltios (“1” lógico).

Figura 25

Modulo relé 2 canales



Nota. Módulo relé 2 canales. Elaborado por: Bryan Velasco M.

Tabla 11
Características

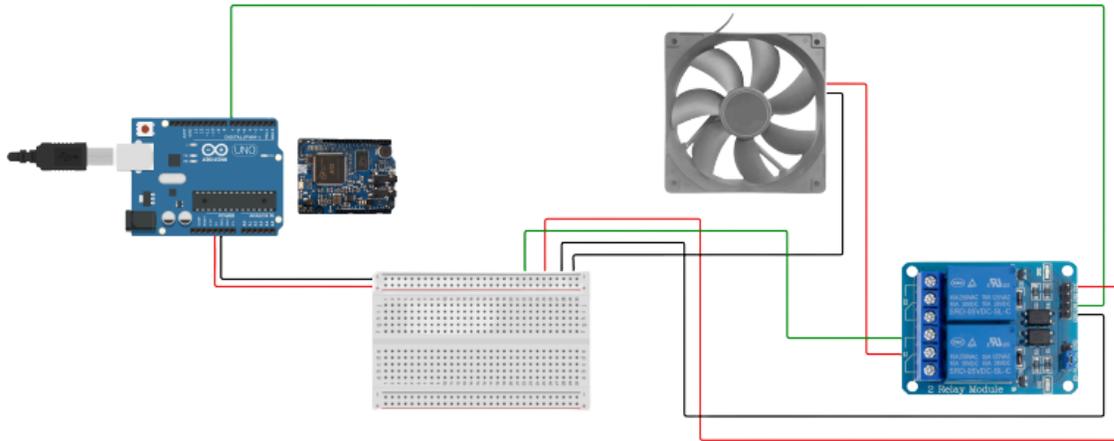
| | |
|----------------------|-----------------------|
| Voltaje de operación | 5V |
| Señal de control TTL | 3.3V o 5V |
| Numero de canales | 2 |
| Capacidad máxima | 10A/250VAC, 10A/30VDC |
| Corriente máxima | 10A (NO), 5A (NC) |
| Tiempo de acción | 10 ms / 5 ms |
| Activar salida NO | 0V |

Nota. Características técnicas modulo relé. Elaborado por: Bryan Velasco M.

2.2.8. Modelado del circuito

Figura 26

Modelado ventilador DC



Nota. Modelado ventilador DC y módulo relé. Elaborado por: Bryan Velasco M.

En la figura 26, se observa la conexión del ventilador DC por medio de un módulo de suministro de corriente (módulo relé) con la placa de reconocimiento, está colocada a un lado de Arduino para observar los pines que fueron asignados.

Tabla 12
Pines asignados a Ventilador DC

| PIN | Detalle |
|-----|---------------|
| 7 | Ventilador DC |

Nota. Pines asignados a Ventilador DC. Elaborado por: Bryan Velasco M.

2.2.9. Bombilla

Es una fuente que produce luz artificial por medio de conductores eléctricos, calientan un filamento hasta producir una fuente de luz por incandescencia.

Figura 27

Bombilla

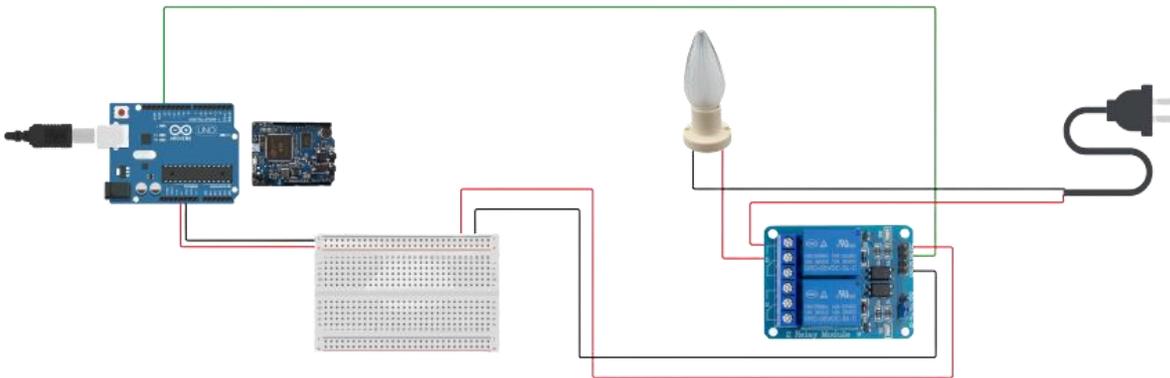


Nota. Bombilla. Elaborado por: Bryan Velasco M.

2.2.10. Modelado del circuito

Figura 28

Modelado bombilla



Nota. Modelado bombilla y módulo relé. Elaborado por: Bryan Velasco M.

En la figura 28, se observa la conexión de la bombilla por medio de un módulo de suministro de corriente (módulo relé) con la placa de reconocimiento, está colocada a un lado de Arduino para observar los pines que fueron asignados.

Tabla 13

Pines asignados a Bombilla

| PIN | Detalle |
|-----|----------|
| 11 | Bombilla |

Nota. Pines asignados a bombilla. Elaborado por: Bryan Velasco M.

CAPÍTULO 3

En el tercer capítulo se realizan escenarios de pruebas por medio del uso de los componentes mencionados previamente, detallando su funcionamiento dentro de la IDE de Arduino, además de analizar las pruebas obtenidas como resultado.

3. PRUEBAS Y RESULTADOS

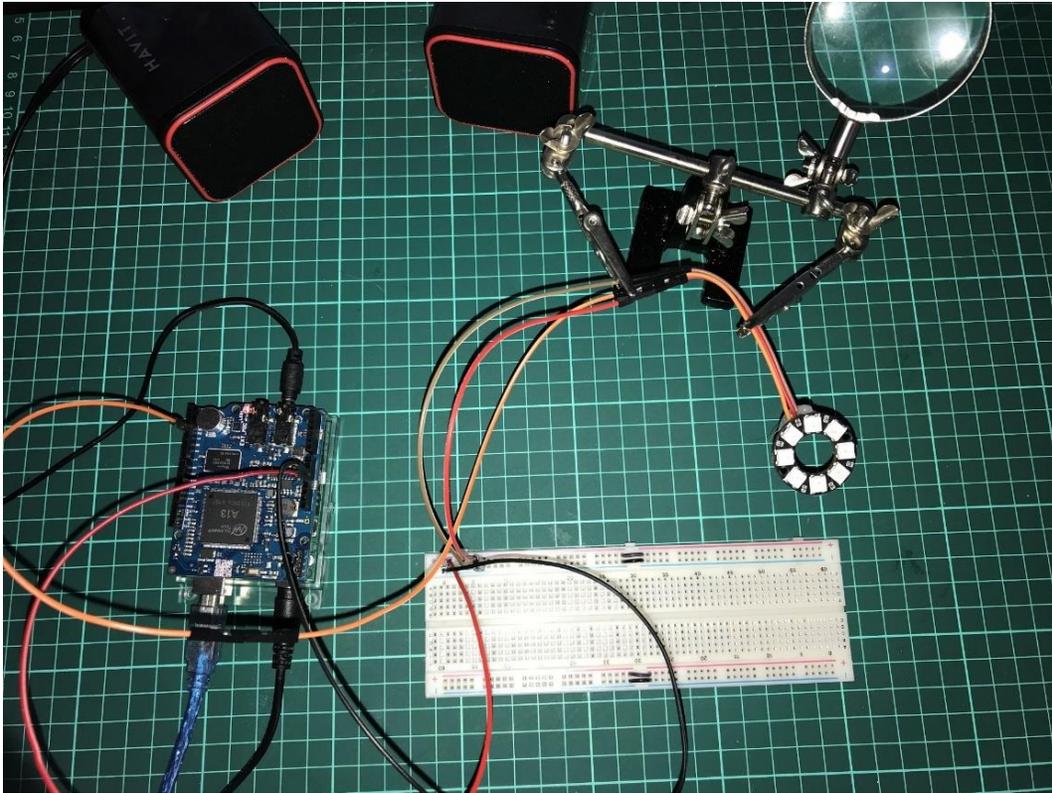
3.1. Componentes

3.1.1. *Led Neopixel RGB*

La prueba de encendido del anillo RGB se realiza mediante una orden previamente ingresada en la función de agregación de oraciones dentro de la IDE de Arduino y su programación correspondiente, este led tiene la capacidad de programar cualquier tipo de color. Se utilizará el modelo Neopixel creado por Adafruit Industries, este led puede controlarse por medio de un pin digital (DIN).

Figura 29

Led RGB en estado apagado



Nota. Funcionalidad del prototipo con el Led RGB. Elaborado por: Bryan Velasco M.

Para controlar el Led es necesario implementar la librería <Adafruit_NeoPixel.h>, de manera que, su librería puede instalarse desde la IDE de Arduino.

Figura 30

Librería de luces Led RGB desarrollado por Adafruit

```
#include <Adafruit_NeoPixel.h>
```

Nota. Declaración de la librería Adafruit NeoPixel. Elaborado por: Bryan Velasco M.

Figura 31

Definición de variables

```

// LED RGB
#define NUMPIXELS 8
//LED 1
#define PIN      6
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
//LED 2
#define PIN2     5
Adafruit_NeoPixel pixels2(NUMPIXELS, PIN2, NEO_GRB + NEO_KHZ800);
//LED 3
#define PIN3     4
Adafruit_NeoPixel pixels3(NUMPIXELS, PIN3, NEO_GRB + NEO_KHZ800);
//LED 4
#define PIN4     3
Adafruit_NeoPixel pixels4(NUMPIXELS, PIN4, NEO_GRB + NEO_KHZ800);

#define DELAYVAL 500 //tiempo de espera en ms

```

Nota. Definición de variables usadas para Led RGB. Elaborado por: Bryan Velasco M.

Figura 32

Oraciones creadas para el reconocimiento de voz

```

recognizer.addSentence("Bodega encendendida"); // Add Spanish sentence 3
recognizer.addSentence("Bodega apagada"); // Add Spanish sentence 4
recognizer.addSentence("Oficina encendida"); // Add Spanish sentence 5
recognizer.addSentence("Oficina apagada"); // Add Spanish sentence 6
recognizer.addSentence("Habitacion encendendida"); // Add Spanish sentence 7
recognizer.addSentence("Habitacion apagada"); // Add Spanish sentence 8
recognizer.addSentence("Baño encendido"); // Add Spanish sentence 9
recognizer.addSentence("Baño apagado"); // Add Spanish sentence 10

```

Nota. Las oraciones ingresadas en la función serán las que reconocerá Movi. Elaborado por: Bryan Velasco M.

Figura 33

Respuesta al comando de voz funciones 3 y 4

```

if (res==3) {                                // Sentence 3
    pixels.clear();

    for(int i=0; i<NUMPIXELS; i++) { // Para cada pixel...

        // Modificamos el LED #i, encendiendolo con un color blanco moderadamente brillante
        pixels.setPixelColor(i, pixels.Color(255,255,255));

        pixels.show(); // Mandamos todos los colores con la actualización hecha

        delay(DELAYVAL); // Pausa antes de modificar el color del siguiente LED
    }
    recognizer.say("Encendiendo Bodega"); // Respond a sentence in Spanish
    // Do more ...
}

if (res==4) {                                // Sentence 4
    pixels.clear();
    for(int i=0; i<NUMPIXELS; i++) { // Para cada pixel...

        // Modificamos el LED #i, encendiendolo con un color moderadamente brillante
        pixels.setPixelColor(i, pixels.Color(0, 0, 0));

        pixels.show(); // Mandamos todos los colores con la actualización hecha

        delay(DELAYVAL); // Pausa antes de modificar el color del siguiente LED
    }
    recognizer.say("Apagando Bodega"); // Respond a sentence in Spanish
}

```

Nota. Respuesta a los comandos de voz de las oraciones 3 y 4. Elaborado por: Bryan Velasco M.

Figura 34

Respuesta al comando de voz funciones 5 y 6

```

if (res==5) {                                // Sentence 5
    pixels2.clear();

    for(int i=0; i<NUMPIXELS; i++) { // Para cada pixel...

        // Modificamos el LED #i, encendiendolo con un color blanco moderadamente brillante
        pixels2.setPixelColor(i, pixels2.Color(255,255,255));

        pixels2.show(); // Mandamos todos los colores con la actualización hecha

        delay(DELAYVAL); // Pausa antes de modificar el color del siguiente LED
    }
    recognizer.say("Encendiendo Oficina"); // Respond a sentence in Spanish
    // Do more ...
}

if (res==6) {                                // Sentence 6
    pixels2.clear();

    for(int i=0; i<NUMPIXELS; i++) { // Para cada pixel...

        // Modificamos el LED #i, encendiendolo con un color moderadamente brillante
        pixels2.setPixelColor(i, pixels2.Color(0, 0, 0));

        pixels2.show(); // Mandamos todos los colores con la actualización hecha

        delay(DELAYVAL); // Pausa antes de modificar el color del siguiente LED
    }

    recognizer.say("Apagando Oficina"); // Respond a sentence in Spanish
}

```

Nota. Respuesta a los comandos de voz de las oraciones 5 y 6. Elaborado por: Bryan Velasco M.

Figura 35

Respuesta al comando de voz funciones 7 y 8

```

if (res==7) { // Sentence 7
    pixels3.clear();

    for(int i=0; i<NUMPIXELS; i++) { // Para cada pixel...

        // Modificamos el LED #i, encendiendolo con un color blanco moderadamente brillante
        pixels3.setPixelColor(i, pixels3.Color(255,255,255));

        pixels3.show(); // Mandamos todos los colores con la actualización hecha

        delay(DELAYVAL); // Pausa antes de modificar el color del siguiente LED
    }
    recognizer.say("Encendiendo Habitacion"); // Respond a sentence in Spanish
    // Do more ...
}

if (res==8) { // Sentence 8
    pixels3.clear();
    for(int i=0; i<NUMPIXELS; i++) { // Para cada pixel...

        // Modificamos el LED #i, encendiendolo con un color moderadamente brillante
        pixels3.setPixelColor(i, pixels3.Color(0, 0, 0));

        pixels3.show(); // Mandamos todos los colores con la actualización hecha

        delay(DELAYVAL); // Pausa antes de modificar el color del siguiente LED
    }
    recognizer.say("Apagando Habitacion"); // Respond a sentence in Spanish

}

```

Nota. Respuesta a los comandos de voz de las oraciones 7 y 8. Elaborado por: Bryan Velasco M.

Figura 36

Respuesta al comando de voz funciones 9 y 10

```

if (res==9) { // Sentence 9
  pixels4.clear();

  for(int i=0; i<NUMPIXELS; i++) { // Para cada pixel...

    // Modificamos el LED #i, encendiendolo con un color blanco moderadamente brillante
    pixels4.setPixelColor(i, pixels4.Color(255,255,255));

    pixels4.show(); // Mandamos todos los colores con la actualización hecha

    delay(DELAYVAL); // Pausa antes de modificar el color del siguiente LED
  }
  recognizer.say("Encendiendo Baño"); // Respond a sentence in Spanish
  // Do more ...
}

if (res==10) { // Sentence 10
  pixels4.clear();
  for(int i=0; i<NUMPIXELS; i++) { // Para cada pixel...

    // Modificamos el LED #i, encendiendolo con un color moderadamente brillante
    pixels4.setPixelColor(i, pixels4.Color(0, 0, 0));

    pixels4.show(); // Mandamos todos los colores con la actualización hecha

    delay(DELAYVAL); // Pausa antes de modificar el color del siguiente LED
  }
  recognizer.say("Tira Apagada"); // Respond a sentence in Spanish

  recognizer.say("Apagando Baño"); // Respond a sentence in Spanish
}

```

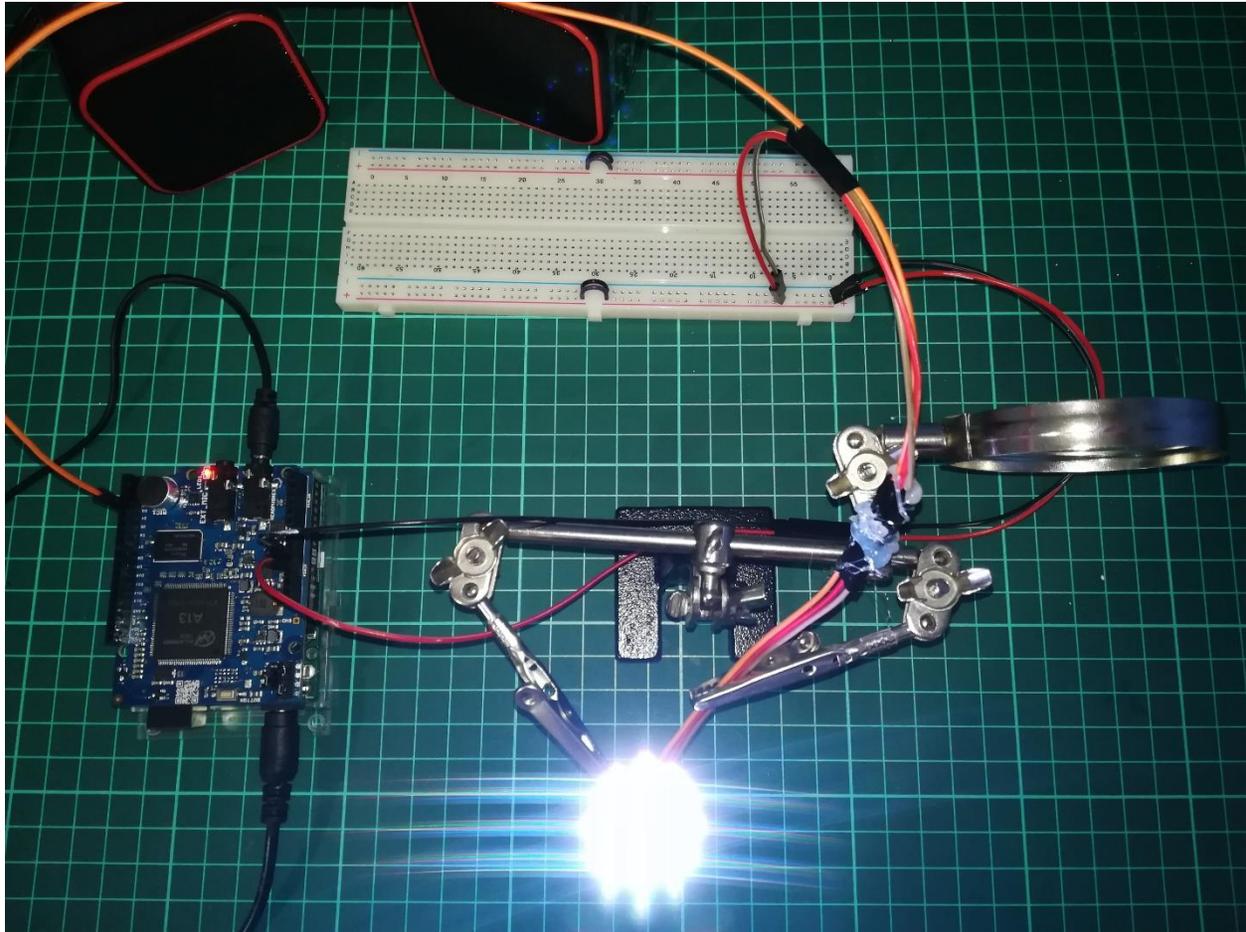
Nota. Respuesta a los comandos de voz de las oraciones 9 y 10. Elaborado por: Bryan Velasco M.

Estas líneas de código la recibir la información de entrada por el comando de voz, reconoce que función se le está dictando, la respuesta a eso es encender el anillo o tira de luces, o a su vez apagar las luces, seguido del sintetizador de voz de Movi que expresará que función se ejecutó.

Se utilizo el color blanco para su ejemplificación, no obstante, existe la posibilidad de cambiar los colores o cambiar uno por uno las luces con diferentes colores dependiendo de la programación del led.

Figura 37

Led RGB estado encendido



Nota. Funcionalidad del prototipo con el Led RGB. Elaborado por: Bryan Velasco M.

Se realizaron pruebas de tiempo de respuesta en segundos con la ayuda de un cronómetro, con el objetivo de analizar su tiempo promedio de respuesta.

Tabla 14
Prueba de encendido/apagado de led RGB

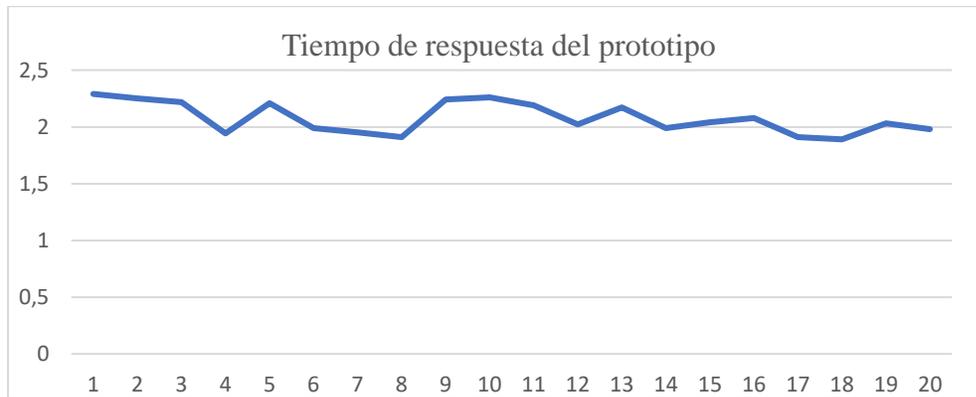
| No. pruebas | Oración | Tiempo de respuesta Movi |
|-------------|--|-----------------------------|
| 1 | recognizer.addSentence("Bodega encendendida"); | 2.29 segundos |
| 2 | recognizer.addSentence("Bodega apagada"); | 2.25 segundos |
| 3 | recognizer.addSentence("Oficina encendida"); | 2.22 segundos |

| | | |
|------------------------------|--|---------------|
| 4 | recognizer.addSentence("Oficina apagada"); | 1.94 segundos |
| 5 | recognizer.addSentence("Habitacion encendendida"); | 2.21 segundos |
| 6 | recognizer.addSentence("Habitacion apagada"); | 1.99 segundos |
| 7 | recognizer.addSentence("Baño encendido"); | 1.95 segundos |
| 8 | recognizer.addSentence("Baño apagado"); | 1.91 segundos |
| 9 | recognizer.addSentence("Bodega encendendida"); | 2.24 segundos |
| 10 | recognizer.addSentence("Bodega apagada"); | 2.26 segundos |
| 11 | recognizer.addSentence("Oficina encendida"); | 2.19 segundos |
| 12 | recognizer.addSentence("Oficina apagada"); | 2.02 segundos |
| 13 | recognizer.addSentence("Habitacion encendendida"); | 2.17 segundos |
| 14 | recognizer.addSentence("Habitacion apagada"); | 1.99 segundos |
| 15 | recognizer.addSentence("Baño encendido"); | 2.04 segundos |
| 16 | recognizer.addSentence("Baño apagado"); | 2.08 segundos |
| 17 | recognizer.addSentence("Bodega encendendida"); | 1.91 segundos |
| 18 | recognizer.addSentence("Bodega apagada"); | 1.89 segundos |
| 19 | recognizer.addSentence("Oficina encendida"); | 2.03 segundos |
| 20 | recognizer.addSentence("Oficina apagada"); | 1.98 segundos |
| Tiempo promedio de respuesta | | 2.08 segundos |

Nota. Pruebas de tiempo realizadas. Elaborado por: Bryan Velasco M.

Figura 38

Gráfica de tiempo de respuesta



Nota. Gráfica que detalla el tiempo de respuesta del prototipo. Elaborado por: Bryan Velasco M.

Tabla 15

Pruebas de funcionamiento del prototipo

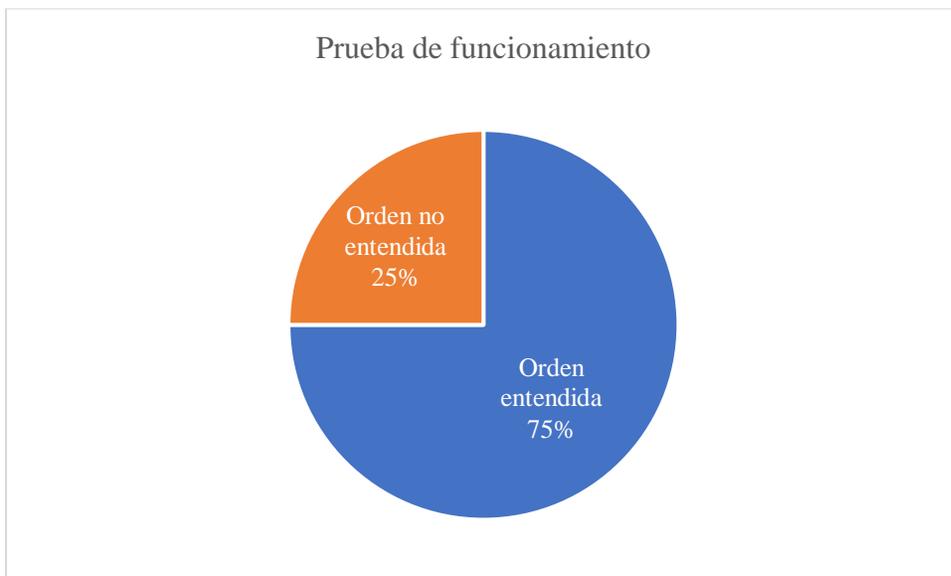
| No. pruebas | Orden entendida | Orden no entendida |
|-------------|-----------------|--------------------|
| 1 | x | |
| 2 | | x |
| 3 | | x |
| 4 | x | |
| 5 | x | |
| 6 | x | |
| 7 | x | |
| 8 | x | |
| 9 | x | |
| 10 | | x |
| 11 | x | |

| | | |
|-------|--------------------------|----------------------------|
| 12 | | x |
| 13 | x | |
| 14 | x | |
| 15 | x | |
| 16 | x | |
| 17 | x | |
| 18 | | x |
| 19 | x | |
| 20 | x | |
| Total | 75 % de orden entendidas | 25 % de orden no entendida |

Nota. Prueba de funcionamiento en porcentajes. Elaborado por: Bryan Velasco M.

Figura 39

Gráfica de funcionalidad



Nota. Grafica en porcentaje de respuestas. Elaborado por: Bryan Velasco M.

La grafica de funcionalidad muestra que hay un porcentaje del 75% momento de reconocer el comando de voz, el 25% fueron comandos no reconocidos por el prototipo.

3.1.2. Motor reductor DC

Figura 40

Oraciones creadas para el reconocimiento de voz

```
recognizer.addSentence("Puerta arriba"); // Add Spanish sentence 11  
recognizer.addSentence("Puerta abajo"); // Add Spanish sentence 12
```

Nota. Las oraciones ingresadas en la función serán las que reconocerá Movi. Elaborado por: Bryan Velasco M.

Figura 41

Definición de variables

```
// motor doble eje  
pinMode(8, OUTPUT); //IN2  
pinMode(9, OUTPUT); //IN1  
Serial.begin(115200);
```

Nota. Definición de variables usadas para el motor. Elaborado por: Bryan Velasco M.

Figura 42

Respuesta al comando de voz función 11

```

if (res==11) {                                     // Sentence 11
  if(i==0){
    digitalWrite(8, HIGH);
    digitalWrite(9, LOW);

    delay(3000); //tiempo de abierto
    i=1;
  }

  if(i==1){
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    i=0;
  }

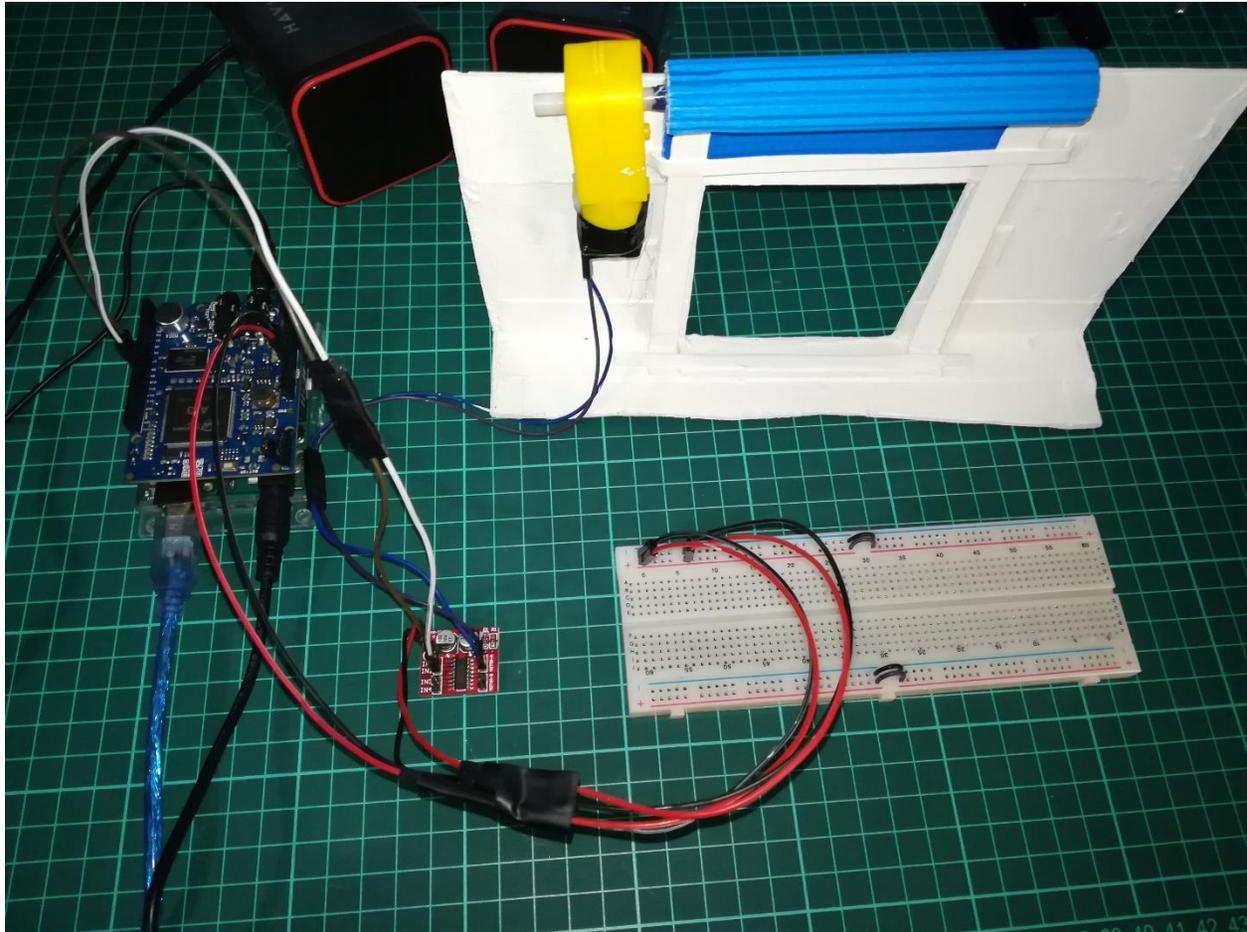
  recognizer.say("Puerta Abierta"); // Respond a sentence in Spanish
}

```

Nota. Respuesta a los comandos de voz de la oración 11. Elaborado por: Bryan Velasco M.

Figura 43

Motor reductor estado puerta abierta



Nota. Funcionalidad del prototipo con motor reductor. Elaborado por: Bryan Velasco M.

El motor reductor con la ayuda del driver girará en sentido que se enrolla una ejemplificación de puerta enrollable en un tiempo limitado en las líneas de código. El sintetizador de voz de Movi expresará la acción realizada al usuario.

Figura 44

Respuesta al comando de voz función 12

```

if (res==12) {                                     // Sentence 12
  if(i2==0){
    digitalWrite(8, LOW);
    digitalWrite(9, HIGH);
    delay(3000); //tiempo de cerrado
    i2=1;
  }

  if(i2==1){
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    i2=0;
  }

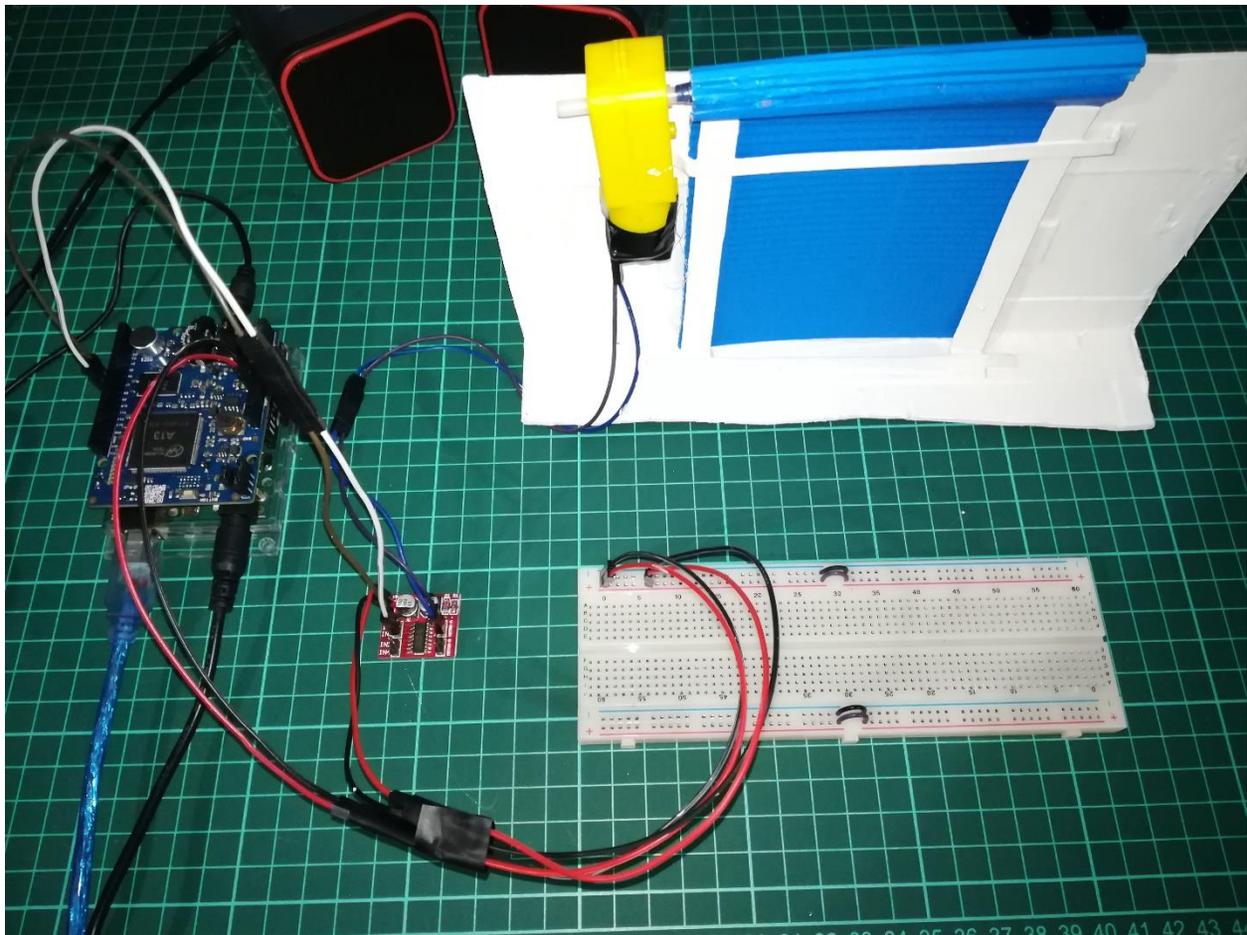
  recognizer.say("Puerta cerrada"); // Respond a sentence in Spanish
}

```

Nota. Respuesta a los comandos de voz de la oración 12. Elaborado por: Bryan Velasco M.

Figura 45

Motor reductor estado puerta cerrada



Nota. Funcionalidad del prototipo con motor reductor. Elaborado por: Bryan Velasco M.

El motor reductor con la ayuda del driver girará en sentido que se desenrolla una ejemplificación de puerta enrollable en un tiempo limitado en las líneas de código. El sintetizador de voz de Movi expresará la acción realizada al usuario.

Se realizaron pruebas de tiempo de respuesta en segundos con la ayuda de un cronómetro, con el objetivo de analizar su tiempo promedio de respuesta.

Tabla 16
Prueba de puerta arriba/abajo del motor reductor

| No. pruebas | Oración | Tiempo de respuesta Movi |
|-------------|---------|-----------------------------|
| | | |

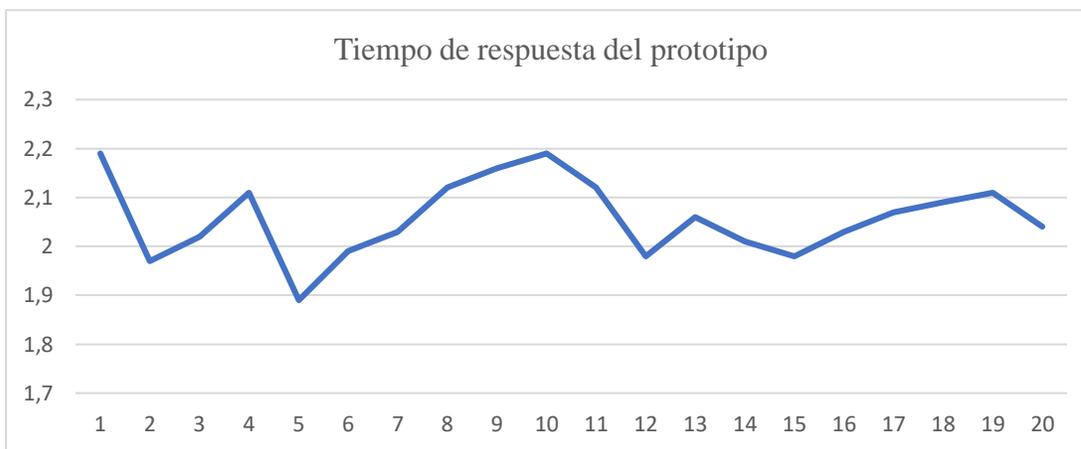
| | | |
|----|--|---------------|
| 1 | recognizer.addSentence("Puerta arriba"); | 2.19 segundos |
| 2 | recognizer.addSentence("Puerta abajo"); | 1.97 segundos |
| 3 | recognizer.addSentence("Puerta arriba"); | 2.02 segundos |
| 4 | recognizer.addSentence("Puerta abajo"); | 2.11 segundos |
| 5 | recognizer.addSentence("Puerta arriba"); | 1.89 segundos |
| 6 | recognizer.addSentence("Puerta abajo"); | 1.99 segundos |
| 7 | recognizer.addSentence("Puerta arriba"); | 2.03 segundos |
| 8 | recognizer.addSentence("Puerta abajo"); | 2.12 segundos |
| 9 | recognizer.addSentence("Puerta arriba"); | 2.16 segundos |
| 10 | recognizer.addSentence("Puerta abajo"); | 2.19 segundos |
| 11 | recognizer.addSentence("Puerta arriba"); | 2.12 segundos |
| 12 | recognizer.addSentence("Puerta abajo"); | 1.98 segundos |
| 13 | recognizer.addSentence("Puerta arriba"); | 2.06 segundos |
| 14 | recognizer.addSentence("Puerta abajo"); | 2.01 segundos |
| 15 | recognizer.addSentence("Puerta arriba"); | 1.98 segundos |
| 16 | recognizer.addSentence("Puerta abajo"); | 2.03 segundos |
| 17 | recognizer.addSentence("Puerta arriba"); | 2.07 segundos |
| 18 | recognizer.addSentence("Puerta abajo"); | 2.09 segundos |

| | | |
|------------------------------|--|---------------|
| 19 | recognizer.addSentence("Puerta arriba"); | 2.11 segundos |
| 20 | recognizer.addSentence("Puerta abajo"); | 2.04 segundos |
| Tiempo promedio de respuesta | | 2.06 segundos |

Nota. Pruebas de tiempo realizadas. Elaborado por: Bryan Velasco M.

Figura 46

Gráfica tiempo de respuesta



Nota. Gráfica que detalla el tiempo de respuesta del prototipo. Elaborado por: Bryan Velasco M.

Tabla 17

Funcionamiento

| No. pruebas | Orden entendida | Orden no entendida |
|-------------|-----------------|--------------------|
| 1 | x | |
| 2 | x | |
| 3 | x | |
| 4 | | x |
| 5 | x | |
| 6 | x | |

| | | |
|-------|--------------------------------|----------------------------------|
| 7 | | x |
| 8 | x | |
| 9 | x | |
| 10 | | x |
| 11 | x | |
| 12 | x | |
| 13 | | x |
| 14 | | x |
| 15 | x | |
| 16 | x | |
| 17 | | x |
| 18 | x | |
| 19 | | x |
| 20 | x | |
| Total | 65 % de orden entendidas | 35 % de orden no entendida |

Nota. Prueba de funcionamiento en porcentajes. Elaborado por: Bryan Velasco M.

Figura 47

Gráfica de funcionalidad



Nota. Grafica en porcentaje de respuestas. Elaborado por: Bryan Velasco M.

La grafica de funcionalidad muestra que hay un porcentaje del 65% momento de reconocer el comando de voz, el 35% fueron comandos no reconocidos por el prototipo.

3.1.3. Ventilador

Figura 48

Oraciones creadas para el reconocimiento de voz

```
recognizer.addSentence("Ventilador apagado"); // Add Spanish sentence 13
recognizer.addSentence("Ventilador encendido"); // Add Spanish sentence 14
```

Nota. Las oraciones ingresadas en la función serán las que reconocerá Movi. Elaborado por: Bryan Velasco M.

Figura 49

Definición de variables

```
const int venti = 7; // Ventilador
```

Nota. Definición de variables usadas para el ventilador. Elaborado por: Bryan Velasco M.

Figura 50

Respuesta al comando de voz función 13 y 14

```
if (res==13) { // Sentence 13
  digitalWrite(venti, HIGH); // Turn on
  recognizer.say("Ventilador Apagado"); // Respond a sentence in Spanish
}

if (res==14) { // Sentence 14
  // wait for a second

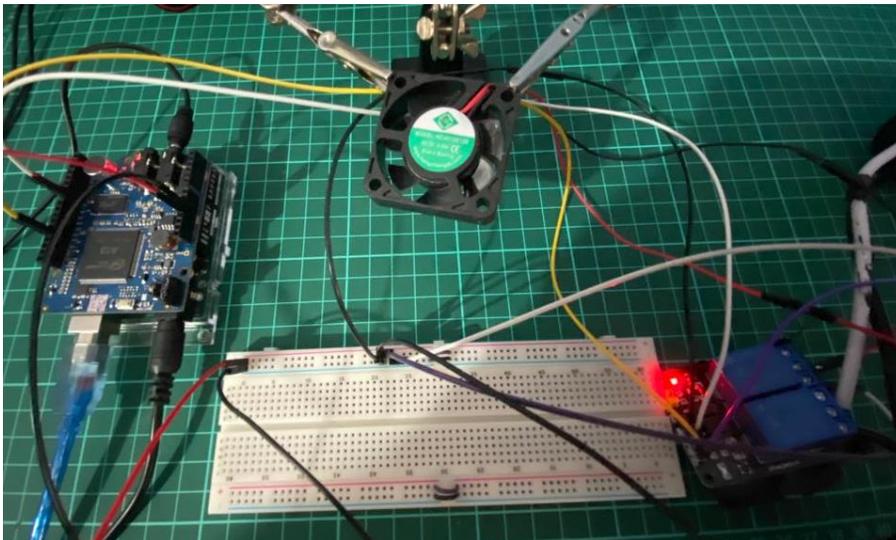
digitalWrite(venti, LOW); // turn the BULB off by making the voltage LOW

delay(1); // wait for a second
  recognizer.say("Ventilador Encendido"); // Respond a sentence in Spanish
}
```

Nota. Respuesta a los comandos de voz de la oración 13 y 14. Elaborado por: Bryan Velasco M.

Figura 51

Ventilador estado apagado



Nota. Funcionalidad del prototipo con motor reductor. Elaborado por: Bryan Velasco M.

El ventilador junto al módulo relé permitirá el paso de corriente en una ejemplificación de ventilador. El sintetizador de voz de Movi expresará la acción realizada al usuario. Este componente podría ser reemplazado con un ventilador real, mediante conexión con capacitadores relé robustos que permitan el flujo de un mayor suministro de energía.

Tabla 18
Pruebas

| No. pruebas | Oración | Tiempo de respuesta Movi |
|-------------|---|-----------------------------|
| 1 | recognizer.addSentence("Ventilador apagado"); | 2.11 segundos |
| 2 | recognizer.addSentence("Ventilador encendido"); | 2.06 segundos |
| 3 | recognizer.addSentence("Ventilador apagado"); | 2.19 segundos |
| 4 | recognizer.addSentence("Ventilador encendido"); | 2.23 segundos |
| 5 | recognizer.addSentence("Ventilador apagado"); | 2.05 segundos |
| 6 | recognizer.addSentence("Ventilador encendido"); | 2.09 segundos |
| 7 | recognizer.addSentence("Ventilador apagado"); | 1.95 segundos |
| 8 | recognizer.addSentence("Ventilador encendido"); | 2.03 segundos |
| 9 | recognizer.addSentence("Ventilador apagado"); | 1.97 segundos |
| 10 | recognizer.addSentence("Ventilador encendido"); | 2.01 segundos |
| 11 | recognizer.addSentence("Ventilador apagado"); | 2.10 segundos |
| 12 | recognizer.addSentence("Ventilador encendido"); | 2.06 segundos |
| 13 | recognizer.addSentence("Ventilador apagado"); | 1.94 segundos |
| 14 | recognizer.addSentence("Ventilador encendido"); | 2.07 segundos |
| 15 | recognizer.addSentence("Ventilador apagado"); | 2.09 segundos |
| 16 | recognizer.addSentence("Ventilador encendido"); | 2.03 segundos |

| | | |
|------------------------------|---|---------------|
| 17 | recognizer.addSentence("Ventilador apagado"); | 1.89 segundos |
| 18 | recognizer.addSentence("Ventilador encendido"); | 1.92 segundos |
| 19 | recognizer.addSentence("Ventilador apagado"); | 2.01 segundos |
| 20 | recognizer.addSentence("Ventilador encendido"); | 1.92 segundos |
| Tiempo promedio de respuesta | | 2.04 segundos |

Nota. Pruebas de tiempo realizadas. Elaborado por: Bryan Velasco M.

Figura 52

Gráfica de tiempo de respuesta



Nota. Gráfica que detalla el tiempo de respuesta del prototipo. Elaborado por: Bryan Velasco M.

Tabla 19

Funcionamiento

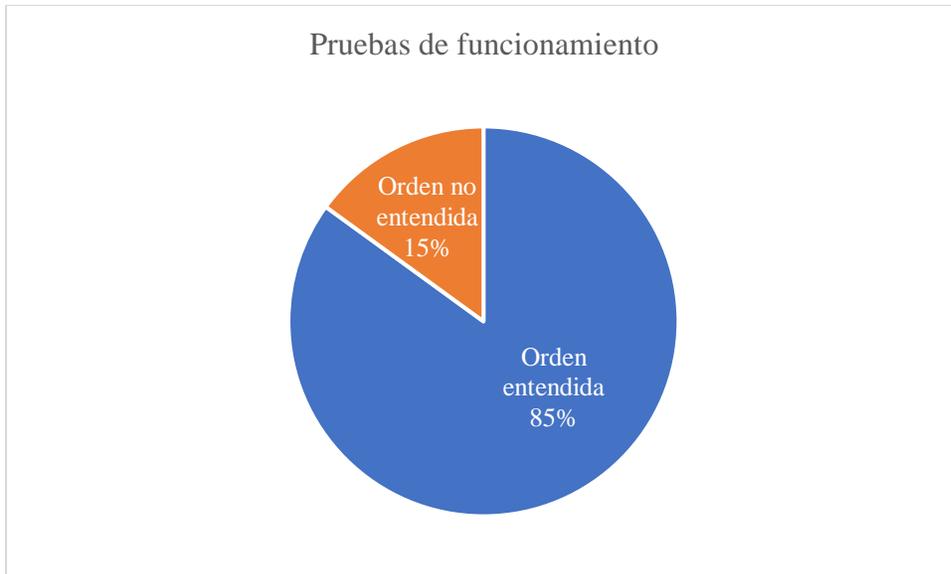
| No. pruebas | Orden entendida | Orden no entendida |
|-------------|-----------------|--------------------|
| 1 | x | |
| 2 | x | |
| 3 | x | |

| | | |
|-------|--------------------------------|----------------------------------|
| 4 | x | |
| 5 | x | |
| 6 | | x |
| 7 | x | |
| 8 | x | |
| 9 | x | |
| 10 | x | |
| 11 | x | |
| 12 | | x |
| 13 | x | |
| 14 | x | |
| 15 | x | |
| 16 | | x |
| 17 | x | |
| 18 | x | |
| 19 | x | |
| 20 | x | |
| Total | 85 % de orden entendidas | 15 % de orden no entendida |

Nota. Prueba de funcionamiento en porcentajes. Elaborado por: Bryan Velasco M.

Figura 53

Gráfica de funcionalidad



Nota. Gráfica en porcentaje de respuestas. Elaborado por: Bryan Velasco M.

La gráfica de funcionalidad muestra que hay un porcentaje del 85% momento de reconocer el comando de voz, el 15% fueron comandos no reconocidos por el prototipo.

3.1.4. Bombilla

Figura 54

Oraciones creadas para el reconocimiento de voz

```
recognizer.addSentence("Foco encendido"); // Add Spanish sentence 15
recognizer.addSentence("Foco apagado"); // Add Spanish sentence 16
```

Nota. Las oraciones ingresadas en la función serán las que reconocerá Movi. Elaborado por: Bryan Velasco M.

Figura 55

Definición de variables

```
const int foco = 13; // Foco
```

Nota. Definición de variables usadas para el ventilador. Elaborado por: Bryan Velasco M.

Figura 56

Respuesta al comando de voz función 15

```
if (res==15) {                                // Sentence 15
    // wait for a second

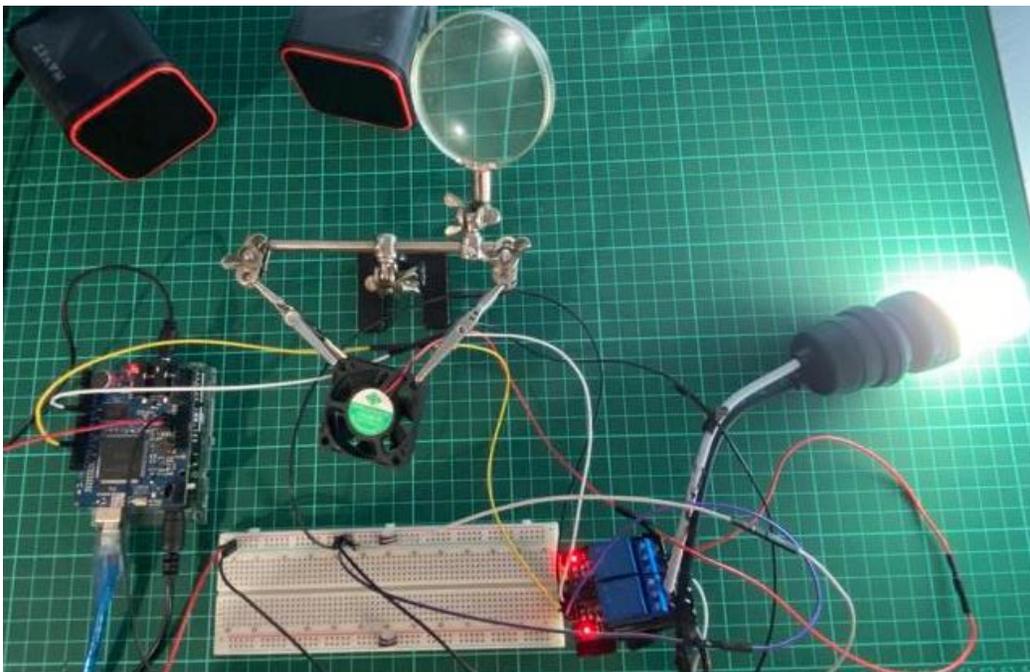
digitalWrite(foco, LOW);    // turn the BULB off by making the voltage LOW

delay(1);                // wait for a second
    recognizer.say("Foco Encendido"); // Respond a sentence in Spanish
}
```

Nota. Respuesta a los comandos de voz de la oración 15. Elaborado por: Bryan Velasco M.

Figura 57

Bombilla encendida



Nota. Funcionalidad del prototipo con bombilla y modulo relé. Elaborado por: Bryan Velasco M.

Figura 58

Respuesta al comando de voz función 16

```

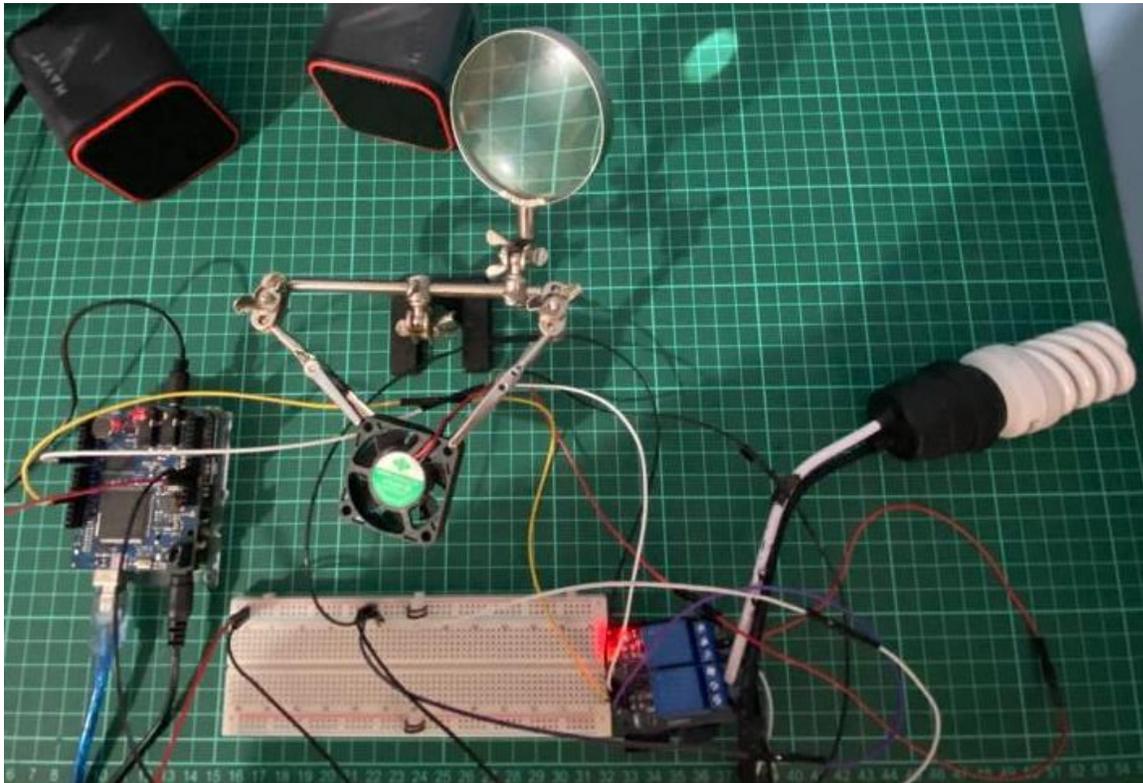
if (res==16) {                                     // Sentence 16
  digitalWrite(foco, HIGH);                       // Turn on
  recognizer.say("Foco Apagado"); // Respond a sentence in Spanish
}

```

Nota. Respuesta a los comandos de voz de la oración 16. Elaborado por: Bryan Velasco M.

Figura 59

Bombilla apagada



Nota. Funcionalidad del prototipo con bombilla y modulo relé. Elaborado por: Bryan Velasco M.

La bombilla junto al módulo relé permitirá el paso de corriente. El sintetizador de voz de Movi expresará la acción realizada al usuario. Este componente necesita conexión con capacitadores relé robustos que permitan el flujo de un mayor suministro de energía.

Tabla 20

Pruebas

| No. pruebas | Oración | Tiempo de respuesta Movi |
|-------------|---|-----------------------------|
| 1 | recognizer.addSentence("Foco encendido"); | 1.89 |
| 2 | recognizer.addSentence("Foco apagado"); | 1.93 |
| 3 | recognizer.addSentence("Foco encendido"); | 2.11 |
| 4 | recognizer.addSentence("Foco apagado"); | 2.19 |
| 5 | recognizer.addSentence("Foco encendido"); | 1.97 |
| 6 | recognizer.addSentence("Foco apagado"); | 2.01 |
| 7 | recognizer.addSentence("Foco encendido"); | 2.05 |
| 8 | recognizer.addSentence("Foco apagado"); | 2.11 |
| 9 | recognizer.addSentence("Foco encendido"); | 2.03 |
| 10 | recognizer.addSentence("Foco apagado"); | 2.07 |
| 11 | recognizer.addSentence("Foco encendido"); | 1.91 |
| 12 | recognizer.addSentence("Foco apagado"); | 1.90 |
| 13 | recognizer.addSentence("Foco encendido"); | 1.99 |
| 14 | recognizer.addSentence("Foco apagado"); | 1.97 |
| 15 | recognizer.addSentence("Foco encendido"); | 1.92 |
| 16 | recognizer.addSentence("Foco apagado"); | 2.03 |
| 17 | recognizer.addSentence("Foco encendido"); | 1.89 |

| | | |
|------------------------------|---|---------------|
| 18 | recognizer.addSentence("Foco apagado"); | 1.91 |
| 19 | recognizer.addSentence("Foco encendido"); | 2.02 |
| 20 | recognizer.addSentence("Foco apagado"); | 1.94 |
| Tiempo promedio de respuesta | | 1.99 segundos |

Nota. Pruebas de tiempo realizadas. Elaborado por: Bryan Velasco M.

Figura 60

Gráfica de tiempo de respuesta



Nota. Gráfica que detalla el tiempo de respuesta del prototipo. Elaborado por: Bryan Velasco M.

Tabla 21

Funcionamiento

| No. pruebas | Orden entendida | Orden no entendida |
|-------------|-----------------|--------------------|
| 1 | x | |
| 2 | x | |
| 3 | x | |
| 4 | x | |
| 5 | x | |
| 6 | | x |

| | | |
|-------|---------------|--------------|
| 7 | x | |
| 8 | x | |
| 9 | x | |
| 10 | x | |
| 11 | x | |
| 12 | x | |
| 13 | x | |
| 14 | x | |
| 15 | x | |
| 16 | x | |
| 17 | x | |
| 18 | x | |
| 19 | x | |
| 20 | x | |
| Total | 95 % de orden | 5 % de orden |

Nota. Prueba de funcionamiento en porcentajes. Elaborado por: Bryan Velasco M.

Figura 61

Gráfica de funcionalidad



Nota. Gráfica en porcentaje de respuestas. Elaborado por: Bryan Velasco M.

La gráfica de funcionalidad muestra que hay un porcentaje del 95% momento de reconocer el comando de voz, el 5% fueron comandos no reconocidos por el prototipo.

3.2.Comparación con Alexa

Para un mejor entendimiento de los resultados, se realizó una comparación con un asistente de voz mundialmente reconocido como Alexa de Amazon, el cual, es trabajado por una empresa tecnológica que cuenta con una gran cantidad de desarrolladores que en conjunto buscan un asisten de voz inteligente y eficiente.

Tabla 22
Tiempos de respuesta asistente de voz Alexa

| Comando de voz | Tiempo de respuesta |
|---------------------|---------------------|
| Encender sala | 0.81 |
| Encender habitación | 0.76 |
| Encender sala | 0.69 |
| Encender habitación | 0.61 |
| Encender sala | 0.59 |
| Encender habitación | 0.61 |
| Encender sala | 0.79 |
| Encender habitación | 0.85 |

| | |
|---------------------|------|
| Encender sala | 0.57 |
| Encender habitación | 0.48 |
| Encender sala | 0.56 |
| Encender habitación | 0.64 |
| Encender sala | 0.71 |
| Encender habitación | 0.65 |
| Encender sala | 0.73 |
| Encender habitación | 0.58 |
| Tiempo promedio | 0.66 |

Nota. Pruebas de tiempo realizadas. Elaborado por: Bryan Velasco M.

Figura 62

Tiempos de respuesta Alexa



Nota. Gráfica que detalla el tiempo de respuesta de Alexa. Elaborado por: Bryan Velasco M.

Los tiempos de respuesta de Alexa son menores en comparación a los tiempos obtenidos con el módulo de Audeme Movi, esto se debe a que Alexa es desarrollada para peticiones robustas por el usuario, entrenada con una gran cantidad de información y su conexión a internet le permite apoyarse de esos datos, en cambio, Movi se basa en una biblioteca de palabras comunes en inglés integrada en su placa, por lo que, su memoria de entrenamiento se ve más limitada, de igual manera su biblioteca de palabras en español no se encuentra tan desarrollada por lo que en ocasiones no consigue el reconocimiento del comando de voz.

3.3.Análisis de resultados

Los resultados obtenidos en el prototipo de reconocimiento de voz indican un alto índice de efectividad en el reconocimiento de los comandos de voz, sin embargo, los resultados en el tiempo de respuesta y procesamiento son lentos en comparación con asistentes de voz comerciales como por ejemplo Alexa, esto se debe a que, ese tipo de asistentes de voz tiene un alto grado de desarrollo y sus programadores se encargan de que el sistema sea eficiente y rápido para el mercado al cual está destinado, además, su entrenamiento en reconocimiento de voz es mucha más avanzado. No obstante, para personas que gustan del desarrollo de aplicaciones o prototipos con estas características este módulo de reconocimiento de voz va a serles de utilidad por su facilidad de manejo, además si se busca un dispositivo inteligente sin algún tipo de seguimiento, interacción o conexión a internet, ya que es un microcontrolador enfocado a la privacidad, no requiere de conexiones a internet para su uso.

Además resalta que los datos cuantitativos fueron realizados por una sola persona, en ambientes de prueba cuyos resultados pueden variar dependiendo del tipo de prueba o el método de medición.

CONCLUSIONES

El desarrollo del prototipo de reconocimiento de voz culminó en escenarios donde se busca automatizar actividades reduciendo la interacción humana, los tiempos de respuesta son óptimos de alrededor de 1 o 2 segundos, siendo superado por asistentes de voz comerciales como Alexa, con menor tiempo en la respuesta a una petición, además, este tipo de herramientas tecnológicas tienen un beneficio para cualquier persona que por algún motivo o inconveniente no pueden realizar una actividad deseada, pudiendo así llevar a cabo dicha acción únicamente utilizando su voz.

Los escenarios a los cuales llevar a cabo estas tecnologías son amplios, desde su uso en la domótica, así como su implementación en el mundo laboral, consiguiendo automatizar áreas y logrando espacios inteligentes, ejemplificado mediante componentes electrónicos a modo demostración, se busca llevarlo a espacios y situaciones de la vida real.

El objetivo del proyecto tuvo como fin el desarrollo de un prototipo que puede adaptarse a proyectos reales utilizando un microcontrolador simple, ya que, el reconocimiento de voz es una herramienta de apoyo al usuario, escucha una orden en tiempo real y da una respuesta en el mismo instante, reduciendo las intervenciones humanas en actividades repetitivas.

RECOMENDACIONES

Para utilizar Movi junto a la placa Arduino es necesario que este desconectado del cable USB, para encenderlo primero se debe conectar a una fuente de alimentación externa, seguido de la conexión USB, al momento de la desconexión se debe retirar la conexión USB y seguido de la alimentación externa, esto para evitar que Movi deje de funcionar de manera correcta. En caso de que exista alguna falla, la memoria SD incluida en Movi posee una copia de seguridad con la cual restablecer la placa.

En cuanto al uso del módulo, es recomendable tratar de hablar de manera clara, esto con el objetivo de que Movi puede entender de mejor manera la oración que debe reconocer.

Al momento de realizar conexiones eléctricas se debe proceder con mucha cautela debido a que el mal manejo puede causar accidentes fatales.

REFERENCIAS BIBLIOGRÁFICAS

- Allwinner Technology CO., Ltd. (29 de Marzo de 2012). *A13 Datasheet*. Allwinner Technology CO., Ltd.: <https://dl.linux-sunxi.org/A13/A13%20Datasheet%20%20v1.12%20%282012-03-29%29.pdf>
- Apolo Díaz, K. B., & Coba Castillo, N. J. (Febrero de 2017). *Implementación de un módulo de reconocimiento de voz para niños mediante el procesamiento de señales aplicado en un caso práctico*. Escuela Superior Politécnica de Chimborazo:
<http://dspace.espoch.edu.ec/handle/123456789/6365>
- AUDEME. (2017). *MOVI Voice Dialog Shield for Arduino boards*. AUDEME:
https://www.audeme.com/uploads/4/3/9/9/43997575/movi_11usermanual111.pdf
- Beltrán Vásquez, L. A. (Mayo de 2003). *SIMULACION DE MODELOS OCULTOS DE MARKOV APLICADOS AL RECONOCIMIENTO DE PALABRAS AISLADAS, UTILIZANDO EL PROGRAMA MATLAB*. Escuela Politécnica Nacional:
<https://bibdigital.epn.edu.ec/bitstream/15000/5467/1/T2201.pdf>
- Brown, E. (6 de Julio de 2015). *Speech recognition and synthesis shield runs Linux*. LinuxGizmos.com: <https://linuxgizmos.com/speech-recognition-and-synthesis-shield-runs-linux/>
- COMEBACKSTUDIOS. (30 de Marzo de 2022). *¿QUÉ ES UN CONVERTIDOR ANALÓGICO-DIGITAL?* COMEBACKSTUDIOS: <https://comebackstudios.es/convertidor-audio-dac-analogico-digital/>
- CORTEZ MORALES, A. A., & CARCELEN PIZA, J. A. (2018). *PROTOTIPO DE UN SISTEMA DE CONTROL Y MONITOREO PARA HOSPITALES MEDIANTE*

COMANDOS DE VOZ. UNIVERSIDAD DE GUAYAQUIL:

<http://repositorio.ug.edu.ec/handle/redug/27217>

Dpto. Técnico de CAPEL. (2008). *Microcontroladores ARM. μCs ARM: Estructura y*

herramientas desarrollo: https://www.redeweb.com/_txt/646/66.pdf

Farro Herrera, V. J., & Villegas Malca, L. M. (2017). *ESCUELA PROFESIONAL DE*

INGENIERÍA ELECTRÓNICA. Sistema de Navegación Automático controlado por voz, maniobrable y seguro adaptable a sillas de ruedas motorizadas:

<https://repositorio.unprg.edu.pe/bitstream/handle/20.500.12893/2436/BC-TES-TMP-1307.pdf?sequence=1&isAllowed=y>

Flores, R. (19 de Octubre de 2019). *La tecnología inteligente crea casas cómodas y con mayores seguridades.* EL TELÉGRAFO:

<https://www.eltelegrafo.com.ec/noticias/tecnologia/1/tecnologia-inteligente-seguridades>

GARNICA CARDENAS, E. O. (2018). *CONTROL DE MANDO POR VOZ PARA PERSONAS*

CON DISCAPACIDAD MOTRIZ. UNIVERSIDAD CATÓLICA DE COLOMBIA:

<https://repository.ucatolica.edu.co/bitstream/10983/18662/1/TESIS%20FINAL.pdf>

Hernando Pericas, F. J. (Mayo de 1993). *TECNICAS DE PROCESADO Y REPRESENTACION*

DE LA SEÑAL DE VOZ PARA EL RECONOCIMIENTO DEL HABLA EN AMBIENTES

RUIDOSOS. UNIVERSIDAD POLITECNICA DE CATALUÑA:

https://www.tdx.cat/bitstream/handle/10803/6911/03_hernandoPericas_capitol_2.pdf?sequence=3

HOGARTEC Expertos en Sistemas Domóticos. (s.f.). *El sostenido crecimiento de la domótica*

en el mundo. HOGARTEC Expertos en Sistemas Domóticos:

<https://hogartec.es/hogartec2/el-sostenido-crecimiento-de-la-domotica-en-el-mundo/>

IBM. (s.f.). *¿Qué es el desarrollo de software?* IBM: <https://www.ibm.com/es-es/topics/software-development>

Lukianova, N. (2021). *Principales partes de un Arduino*. ARDUINO.cl: <https://arduino.cl/principales-partes-de-un-arduino/>

Panta Martínez, J. (27 de Septiembre de 2012). *Control domótico por voz*. Universitat Politècnica de València: <https://riunet.upv.es/bitstream/handle/10251/17631/Memoria.pdf>

Red Hat. (10 de Mayo de 2022). *La automatización*. Red Hat: <https://www.redhat.com/es/topics/automation>

RUIZ SIERRA, A. E., & VERGARA MARTINEZ, J. E. (2019). *DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO DE VOZ PARA SILLAS DE RUEDAS MEDIANTE ARDUINO*. UNIVERSIDAD DE SUCRE: <https://repositorio.unisucre.edu.co/bitstream/handle/001/944/T621.399%20V494.pdf?sequence=1&isAllowed=y>

Salazar Aldaz, J. A. (2012). *Reconocimiento de comandos de voz para control de las funciones multimedia dentro del hogar*. Universidad de las Américas: <https://dspace.udla.edu.ec/handle/33000/2702>

Santa, R. (19 de Abril de 2021). *Domótica crece durante la pandemia*. AVI LATINOAMÉRICA: <https://www.avilatinoamerica.com/202104197300/articulos/integracion-residencial/domotica-crece-durante-la-pandemia.html#:~:text=En%20el%20informe%20Mercado%20de,per%C3%ADodo%20de%20pron%C3%B3stico%202020%2D2025.>

SANTISTEVAN ALVARADO, G. K. (2021). *SISTEMA DE ILUMINACIÓN DOMÓTICO MEDIANTE COMANDO DE VOZ, PARA EL LABORATORIO DE ELECTRÓNICA Y ROBÓTICA DE LA UNIVERSIDAD ESTATAL DEL SUR DE MANABÍ*. UNIVERSIDAD ESTATAL DEL SUR DE MANABÍ:

<http://repositorio.unesum.edu.ec/handle/53000/3134>

SOLEDISPA PILLAJO, V. X. (Febrero de 2022). *IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE ALERTA* . ESCUELA POLITÉCNICA NACIONAL:

<https://bibdigital.epn.edu.ec/handle/15000/22631>

Varela Tapia, E., & García Torres, I. (2020). *Cama automática manipulada con un sistema de voz y Arduino para personas con impedimento físico*. Revista de Ciencias Sociales y Humanísticas MAPA: <https://www.revistamapa.org/index.php/es/article/view/210>

ANEXOS