



*“Responsabilidad con pensamiento positivo”*

**UNIVERSIDAD TECNOLÓGICA ISRAEL**

**TRABAJO DE TITULACIÓN**

**CARRERA: ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES**

**TEMA: ESTUDIO, DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA  
ELECTRÓNICO PARA AUTOMATIZACIÓN DE UNA CAMA DE HOSPITAL  
CONTROLADA POR VOZ PARA MOVILIZACIÓN DE PERSONAS CON  
CAPACIDADES ESPECIALES.**

**AUTOR: CARLOS ADRIAN RODRÍGUEZ ATIENCIA**

**TUTOR:ING. MAURICIO ALMINATI**

**AÑO 2014**

## UNIVERSIDAD TECNOLÓGICA ISRAEL

### APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Graduación certifico:

Que el trabajo de graduación "**ESTUDIO, DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA ELECTRÓNICO PARA AUTOMATIZACIÓN DE UNA CAMA DE HOSPITAL CONTROLADA POR VOZ PARA MOVILIZACIÓN DE PERSONAS CON CAPACIDADES ESPECIALES**", presentado por Carlos Adrián Rodríguez Atiencia, estudiante de la Carrera de Electrónica y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D. M., Noviembre de 2014

TUTOR

---

ING. MAURICIO ALMINATI V.

# UNIVERSIDAD TECNOLÓGICA ISRAEL

## AUTORÍA DEL PROYECTO DE TITULACIÓN

El abajo firmante, en calidad de estudiante de la Carrera de Electrónica y Telecomunicaciones, declaro que los contenidos de este Trabajo de Graduación, requisito previo a la obtención del Grado de Ingeniería en Electrónica y Telecomunicaciones, son absolutamente originales, auténticos y de exclusiva responsabilidad legal y académica del autor.

Quito D.M., Noviembre de 2014

---

Carlos Adrián Rodríguez Atiencia

CC: 1719927475

# UNIVERSIDAD TECNOLÓGICA ISRAEL

## APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de Grado, aprueban la tesis de graduación de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica Israel para títulos de pregrado.

Quito D.M., Noviembre 2014

Para constancia firman:

### TRIBUNAL DE GRADO

.....

PRESIDENTE

.....

MIEMBRO 1

.....

MIEMBRO 2

## **AGRADECIMIENTO**

A Dios por haberme dado la vida, a mi patria, a mis padres y todas las personas que me brindaron su apoyo y colaboración cuando más lo necesitaba.

## **DEDICATORIA**

Este trabajo realizado con esfuerzo, está dedicado a mis padres, abuela y familiares.

## ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	II
AUTORÍA DEL PROYECTO DE TITULACIÓN.....	III
APROBACIÓN DEL TRIBUNAL DE GRADO.....	IV
AGRADECIMIENTO .....	V
DEDICATORIA.....	VI
ÍNDICE GENERAL.....	VII
ÍNDICE DE TABLAS .....	XI
ÍNDICE DE GRÁFICOS .....	XII
RESUMEN .....	XIV
ABSTRACT .....	XVI
CAPÍTULO 1 .....	1
PROBLEMATIZACIÓN .....	1
1.1 Antecedentes.....	1
1.2 Problema Investigado.....	2
1.3 Problema Principal .....	3
1.4 Problemas Secundarios.....	3
1.5 Justificación .....	4
1.6 Objetivos.....	4
1.6.1 Objetivo Principal .....	4
1.6.2 Objetivos Específicos.....	5
1.7 Metodología.....	5
CAPÍTULO 2 .....	6
MARCO TEÓRICO Y CONCEPTUAL.....	6
Introducción.....	6
2.1 Marco Teórico.....	6
2.1.1 Reconocimiento de la voz.....	6
2.1.2 Microcontroladores .....	7

2.1.3.1 Arduino Mega .....	8
2.1.4 EasyVR.....	9
2.1.5 Motores de Corriente Alterna .....	9
2.1.6 Fritzing .....	9
2.1.7 Audacity .....	10
2.1.8 Balabolka .....	11
2.2.1.1 Aplicaciones .....	13
2.2.1.2 Características.....	13
2.2.2 EasyVR Commander: INTERFACE GRÁFICA DE USUARIO (GUI) ....	14
2.2.2.1 Configuración .....	15
2.2.2 Arduino Mega.....	19
2.1.3.1 Potencia .....	22
2.2.3 Cama Hospitalaria Eléctrica - Carroll Healthcare - Echo.....	22
CAPÍTULO 3 .....	25
ESTUDIO, DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA ELECTRÓNICO PARA AUTOMATIZACIÓN DE UNA CAMA DE HOSPITAL CONTROLADA POR VOZ PARA MOVILIZACIÓN DE PERSONAS CON CAPACIDADES ESPECIALES .....	25
Introducción.....	25
3.1 Estudio sobre la implantación de sistemas de reconocimiento de voz para controlar sistemas electrónicos y electromecánicos.....	25
3.1.1 EasyVR.....	26
3.1.1.1 Micrófono.....	27
3.1.1.2 Salida de Audio .....	27
3.1.1.3 Construcción de conjuntos de comandos .....	28
3.1.1.3 Tablas de sonido .....	31
3.1.1.4 Módulo de 8 relés para Arduino.....	33
3.1.1.5 Motores .....	34
3.2 Diseño de una interface de control automático, utilizando reconocimiento de voz, añadiendo al programa del microcontrolador comandos que posibiliten el cumplimiento de los movimientos de una cama de hospital.....	36
3.2.1 Alimentación .....	37
3.2.2 Entrada y Salida.....	38
3.2.3 Arduino 3.2 UI .....	38

3.2.4 Configuración.....	39
3.2.5 Programación.....	39
3.2.6 Hardware .....	40
3.2.7 Integración de Arduino Mega con conexiones Eléctricas y de Control.	40
3.2.8 Software.....	41
3.3 Construcción de un sistema automatizado capaz de obedecer a comandos de voz para movilizar una persona en una cama de hospital.....	43
3.3.1 Montaje Hardware.....	43
3.3.1.1 Conexión del Micrófono en el módulo EasyVR .....	43
3.3.1.2 Conexión de Salida de Audio en el módulo EasyVR.....	44
3.3.1.3 Conexión del módulo EasyVR en Arduino Mega.....	45
3.3.1.4 Conexión del módulo de 8 Relés.....	45
3.3.1.5 Conexiones Eléctricas y de Control.....	47
3.3.2 Montaje de Software .....	47
3.3.2.1 Codificación Software.....	48
3.3.2.2 Tabla de Sonidos.....	49
3.3.3 Implementación del sistema.....	50
CAPÍTULO 4 .....	54
RESULTADOS Y COSTOS .....	54
4.1 Evaluación Técnica .....	54
4.2 Pruebas de funcionamiento.....	54
4.2.1 Pruebas de funcionamiento del EasyVR.....	55
4.2.2 Pruebas de funcionamiento de Arduino Mega .....	56
4.2.3 Pruebas de funcionamiento de EasyVR GUI 2.1.8 .....	56
4.2.5 Pruebas de funcionamiento de los Motores .....	57
4.2.6 Pruebas de funcionamiento de la Cama Automatizada por Voz .....	58
4.3 Análisis de resultados.....	59
4.3.1 Análisis de la Evaluación técnica. ....	59
4.3.2 Análisis del funcionamiento del EasyVR.....	59
4.3.3 Análisis del funcionamiento de GUI EasyVR .....	60
4.3.4 Análisis del funcionamiento Arduino Mega .....	60
4.3.5 Análisis del funcionamiento Análisis Arduino UI .....	60
4.3.6 Análisis del funcionamiento de la Integración Arduino Mega con EasyVR .....	60

4.3.7 Análisis del funcionamiento de los Motores .....	61
4.3.8 Análisis del funcionamiento de la Cama Automatizada.....	61
4.4 Análisis del Proyecto Implementado.....	61
4.4.1 Características positivas .....	61
4.4.1 Características negativas .....	62
4.5 Costos del proyecto .....	62
CONCLUSIONES Y RECOMENDACIONES .....	64
5.1 CONCLUSIONES.....	64
5.2 RECOMENDACIONES.....	66
5.3 BIBLIOGRAFÍA .....	67
ANEXOS.....	69

## ÍNDICE DE TABLAS

Tabla 2.2: Condiciones de Trabajo Recomendadas.....	17
Tabla 2.3: Características eléctricas .....	18
Tabla 2.4: Consumo de corriente Eléctrica.....	18
Tabla 2.5: Tabla comparativa de de Productos VEER.....	19
Tabla 2.6: Especificaciones Arduino Mega .....	20
Tabla 2.7 Especificaciones de la Cama Eléctrica.....	23
Tabla 2.8 Tabla de medidas .....	23
Tabla 2.9 Tabla de costos del proyecto .....	24
Tabla 3.1 Resumen de comandos y voces predefinidos.....	30
Tabla 3.2 Tabla de comandos del Wordset del EasyVR.....	41
Tabla 3.3 Sonidos Generados por Balabolka (TTS).....	49
Tabla 4.1 Evaluación técnica .....	54
Tabla 4.2 Tabla Pruebas EasyVR .....	55
Tabla 4.3 Tiempos de respuesta y asertividad .....	55
Tabla 4.4 Tabla de prueba de Arduino Mega .....	56
Tabla 4.5 Tabla de prueba de Arduino UI.....	56
Tabla 4.6 Tabla de prueba de Arduino UI.....	57
Tabla 4.7 Pruebas Tabla de Motores.....	58
Tabla 4.8 Tabla de pruebas simultaneas .....	59
Tabla 4.10 Tabla de Costos del proyecto.....	63

## ÍNDICE DE GRÁFICOS

Figura 2.1 Diagrama Conexión Motores .....	9
Figura 2.2 Captura de pantalla de Fritzing .....	10
Figura 2.3 Captura de pantalla de Audacy .....	11
Figura 2.4 Captura de pantalla de Balabolka .....	12
Figura 2.5 El módulo de reconocimiento EasyVR .....	13
Figura 2.6 El área de trabajo EasyVR GLL.....	15
Figura 2.7 Ventana de Grabación del comando.....	16
Figura 2.8 Distribución de pines del EasyVR .....	16
Figura 2.9: Circuito Arduino Mega .....	21
Figura 2.10: Diagrama Electrónico de Arduino Mega .....	21
Figura 2.11: Cama Hospitalaria Electrica .....	22
Figura 2.12 Movimientos de la cama .....	24
Figura 3.1 Conexión entre los módulos EasyVR y Arduino .....	27
Figura 3.2 Comandos para llamar un wordset predefinido en EasyVR .....	29
Figura 3.3 QuickSynthesis™ herramienta externa para la creación de la Tabla de Sonidos .....	33
Figura 3.4 Módulo de 8 Relés para Arduino .....	33
Figura 3.5 Esquema de conexión de un Relé. ....	34
Figura 3.6 Distribución de Pines Conector Motor.....	35
Figura 3.7 Dos Motores de Inversión de giro.....	35
Figura 3.9 Diagrama de funcionamiento por etapas. ....	37
Figura 3.12 Flujograma de comandos de voz. ....	42
Figura 3.13 Diagrama de flujo de sistema de control por voz .....	43
Figura 3.14 Conexión del Micrófono Módulo EasyVR. ....	44
Figura 3.15 Conexión Parlante .....	44
Figura 3.16 Conexión de EasyVR en Arduino Mega .....	45
Figura 3.17 Conexión de Módulo de Relés. ....	46
Figura 3.18 Montaje de Módulos.....	46

Figura 3.19 Conexiones de las cargas a los relés. ....	47
Figura 3.20 Arduino UI con el Archivo prueba00.ino.....	48
Figura 3.21 Conexiones de los motores.....	50
Figura 3.22 Sistema implementado en la Cama .....	51
Figura 3.23 Visión General de la Cama .....	52
Figura 3.24 Parte de elevación del espaldar. ....	53
Figura 3.25 Proyecto terminado .....	53

## RESUMEN

El proyecto consiste en la investigación e integración de la infraestructura de una cama de hospital con un sistema electrónico con el objetivo de automatizar la cama de hospital a través de órdenes de voz, sin necesidad de utilizar ninguna interfaz física, como pueda ser el teclado, ratón o pedales.

El empleo de la tecnología de reconocimiento de voz a través del módulo EasyVR para comunicarse con otros módulos como un microcontrolador Arduino ATmega328 que permita controlar los motores que funcionan en el modelo de la cama del hospital.

En este informe técnico se detalla por medio de capítulos la forma en la que fue realizada la investigación y desarrollo de proyecto como se detalla a continuación:

En el capítulo 1 se detalla la información más relevante sobre el hospital, los problemas que tienen las personas tetrapléjicas y las personas de la tercera edad para realizar movimientos dentro de las camas de hospital como antecedentes y problema investigado. De igual manera se ha incluido en este capítulo los objetivos trazados para este proyecto y la metodología utilizada en el mismo.

En el capítulo 2 se encuentra la información del marco referencial dividida en el marco teórico que trata sobre los conceptos necesarios que sustentan la información para entender de mejor manera sobre el funcionamiento del proyecto, y el marco conceptual que básicamente se encuentra detallado las especificaciones técnicas de cada elemento electrónico y el motivo por el cual fue elegido aquel modelo entre varios.

El capítulo 3 trata sobre el diseño, montaje e implementación del prototipo y cada fase que se subdivide en tres etapas que son las de diseño electrónico, de igual manera el diseño de software y como última etapa el diseño mecánico. La siguiente fase es de montaje donde se realizan el montaje del diseño electrónico en los módulos de Arduino independientemente para proceder al desarrollo del software y a su vez el montaje del software en el microcontrolador y efectuar las primeras pruebas de funcionamiento y finalmente el montaje mecánico que es el procedimiento con el cual se integra las placas electrónicas en la cama de

hospital. La última fase es la de implementación de hardware en la cual en base al diseño electrónico se consolida las placas electrónicas y módulos, en la implementación del sistema se observa la instalación de cada uno de los dispositivos que conforman el prototipo, en una cama de hospital para verificar el funcionamiento.

En el capítulo 4 se detallan las pruebas a las que fue sometido el prototipo para ser evaluado y realizar correcciones si fuera el caso, así mismo el análisis de las pruebas efectuadas al prototipo y finalmente el costo del proyecto.

El capítulo 5 contiene el análisis del desarrollo del proyecto simplificado en conclusiones y recomendaciones.

ANEXOS, donde se presentan las fichas técnicas de cada microcontrolador, chips y módulos que usaron.

## **ABSTRACT**

The project involves research and integration of the infrastructure of a hospital bed with an electronic system in order to automate the hospital bed through voice commands, without using any physical interface, as can be the keyboard, mouse or pedals.

The use of voice recognition technology through EasyVR module to communicate with other modules like an Arduino ATmega328 microcontroller that controls your engine operating in the model of the hospital bed.

In this technical report detailed chapters by the way in which the investigation was conducted and project development as follows:

In Chapter 1 the most relevant information about the hospital, the problems faced by people and people tetraplegic seniors to perform moves within the hospital beds as background and detailed research problem. Similarly has been included in this chapter the objectives for this project and the methodology used in it.

In Chapter 2 of the framework information divided into the theoretical framework that addresses the concepts underlying the information needed to better understand the functioning of the project and the conceptual framework that is basically detailed technical specifications is each electronic item and why that model was chosen among several.

Chapter 3 deals with the design, installation and implementation of the prototype and each phase is divided into three stages which are the electronic design, just as the design of software and as the last step mechanical design. The next phase is where the mounting assembly is made in electronic design modules for Arduino independently proceed with the development of software and in turn the assembly into the microcontroller software and perform the first test runs and finally the mechanical assembly is the process by which integrates electronic boards in the hospital bed. The last phase is the implementation of hardware in which based on the electronic design electronic modules and plates is consolidated in the

implementation of the system the installation of each of the devices that make up the prototype in a hospital bed is observed to verify operation.

In chapter 4 the tests he underwent the prototype for evaluation and make corrections if necessary analysis of the prototype tests carried out and finally the cost of the project are listed, also.

Chapter 5 contains the analysis of the development of simplified project findings and recommendations.

ANNEXES where the technical specifications of each microcontroller chips and modules they used are presented.

## CAPÍTULO 1

### PROBLEMATIZACIÓN

#### 1.1 Antecedentes

El sueño del Padre José Carolo fue dar una oportunidad de una vida digna a todos los habitantes del sur de Quito. A pesar de las dificultades siempre ponía fe, esperanza y amor, acompañados de mucho trabajo, se dio el tiempo para construir iglesias, casas parroquiales, centros médicos, aulas, viviendas familiares, etc.,

La Fundación Tierra Nueva tiene a su cargo la gestión de todos los proyectos y sistemas creados por el Padre Carolo entre ellos el más importante el Hospital Un Canto a la Vida.

La Fundación Tierra Nueva es una organización social sin fines de lucro, fue creada por el Padre José Carolo para servir en Salud, Educación, y Protección Social a grupos de atención prioritaria del sur de Quito.

Las obras del Padre empiezan desde 1971. En el año 1992 toma el nombre de Fundación Tierra Nueva emprendiendo su trabajo con visitas de casa en casa buscando construir el Centro Médico de Especialidades.

Actualmente, la Fundación tiene varios proyectos con el fin de mejorar la vida de los habitantes del sur de Quito y sectores cercanos, la fundación se sostiene a través de organizaciones privadas, públicas, nacionales, extranjeras, donantes y voluntarios de diferentes países: España, Italia, Estados Unidos, Suiza, Alemania, Canadá, etc.

El Hospital tiene las siguientes especialidades:

- Medicina general
- Pediatría
- Ortopedia
- Traumatología

- Cirugías de prótesis de cadera prótesis de rodilla, pie equinovaro, luxación de cadera)
- Cirugía plástica facial (labio leporino, paladar hendido, reconstrucción de orejas, quemaduras)
- Ginecología
- Odontología
- Oftalmología
- Fisioterapia

En el año se benefician 8000 usuarios que reciben los servicios médicos quirúrgicos, de varios profesionales nacionales y extranjeros. El Hospital Padre Carolo “Un Canto a la Vida” se caracteriza por ser docente, de investigación. Se cuenta con 13 misiones conformadas por médicos especialistas, enfermeras, anesthesiólogos, voluntarios y traductores, quienes vienen de los diferentes países para hacer las operaciones complejas.

## **1.2 Problema Investigado**

No sólo los adultos, jóvenes y niños asisten a los hospitales, también están las personas con discapacidades visuales, personas parapléjicas, personas con dificultades motrices o personas de la tercera edad que por su condición o discapacidad tienen dificultades en el movimiento, para subir, sentarse, llamar a las enfermeras o enfermeros, para encender luces o un ventilador dentro del hospital; por lo que para estas personas el tiempo y estancia en el hospital no es muy placentero, además de que se interrumpirá las labores de los empleados y de los usuarios que visitan el hospital al tratar de ayudar a estas personas.

Las camas hospitalarias actualmente son eléctricas, mecánicas e hidráulicas, teniendo en el hospital el mayor porcentaje de camas mecánicas, dificultando la autonomía del

paciente y haciéndola dependiente de otra persona para poder realizar el mínimo movimiento dentro de la cama.

Se ha investigado que existe un proveedor brasileño de camas hospitalarias automatizadas por voz, las cuales necesitan de un computador para la implementación del sistema, lo cual tiene un costo adicional, lo que hace que una cama con estas características sea muy costosa.

### **1.3 Problema Principal**

Las personas con capacidades especiales o de la tercera edad cuando se encuentran hospitalizados tienen la dificultad de movilización en la cama hospitalaria.

### **1.4 Problemas Secundarios**

1. No existe un sistema tecnológico sobre reconocimiento de voz para la automatización de camas de hospital.
2. No se conoce diseños económicos de fácil implementación y que no utilicen computador, para un sistema interface utilizando reconocimiento de voz, para la automatización de una cama de hospital.
3. Hace falta la construcción de un proyecto innovador en el país, integrando sistemas y circuitos capaces de obedecer a comandos de voz para movilizar una persona en una cama de hospital.

## **1.5 Justificación**

El proyecto servirá para estudiar la forma de cómo comunicar, e interconectar diferentes tipos de tecnologías. Se emplea la tecnología de reconocimiento de voz a través del módulo EasyVR que utiliza la interfaz de serie RS-232 para comunicarse con otros módulos como un microcontrolador Arduino ATmega2560 que controlará los motores del sistema electromecánico.

El tema del proyecto es muy actual porque a través este sistema existirá varios aplicativos, utilizando tecnología Mando por Voz y Arduino que es una interface muy usada últimamente.

Ayudará a resolver muchos problemas de las personas con capacidades especiales en lo que refiere a su estancia dentro del hospital de una manera automatizada, utilizando el mando de voz para controlar el movimiento de una cama de hospital.

La implementación de este sistema ayudará a las personas que por su condición o discapacidad tiene problemas para la manipulación de botones, o manipulación de palancas o en el peor de los casos una cama mecánica que tenga que estar constantemente dependiente de otra persona.

## **1.6 Objetivos**

### **1.6.1 Objetivo Principal**

Estudio, diseño y construcción de un sistema electrónico para automatización de una cama de hospital controlada por voz para movilización de personas con capacidades especiales.

### **1.6.2 Objetivos Específicos**

- 1.-Estudiar sobre la implantación de sistemas de reconocimiento de voz para controlar sistemas electrónicos y electromecánicos.
- 2.- Diseñar una interface de control automático, utilizando reconocimiento de voz, añadiendo al programa del microcontrolador comandos que posibiliten el cumplimiento de los movimientos de una cama de hospital.
- 3.- Construir un sistema automatizado capaz de obedecer comandos de voz para movilizar una persona en una cama de hospital.

### **1.7 Metodología**

#### **Análisis**

El análisis se lo utilizó en la investigación después de tener la compilación de toda la información analizando cuál es útil y cuál no. Relacionando información y volviendo a investigar el funcionamiento de los sistemas.

#### **Síntesis**

La síntesis de todo lo investigado sistematizando todo en un sólo documento para el diseño e implementación.

#### **Inducción**

Se utilizó este método para la realización de pruebas conexiones, configuraciones y diseño de las tarjetas.

## **CAPÍTULO 2**

### **MARCO TEÓRICO Y CONCEPTUAL**

#### **Introducción**

En este capítulo se presentan las características, conceptos en los cuales se basan el diseño del prototipo, así también como tablas y gráficos que aportan al desarrollo del proyecto, además se hace una descripción detallada de los componentes del sistema.

#### **2.1 Marco Teórico**

##### **2.1.1 Reconocimiento de la voz**

El reconocimiento automático del habla o reconocimiento automático de voz es una disciplina de inteligencia artificial que tiene como objetivo permitir la comunicación hablada entre seres humanos y los computadores. La dificultad que se plantea en este tipo de sistema es el hacer asistir un conjunto de informaciones que provienen de diversas fuentes de conocimiento (acústica, fonética, fonológica, léxica, sintáctica, semántica y pragmática), en presencia de ambigüedades, incertidumbres y errores inevitables para lograr obtener una interpretación aceptable del mensaje acústico recibido. (Habla, 2014)

El reconocimiento de voz es una herramienta que sirve para procesar señales de voz emitidas por los seres humanos, y reconocer y comparar esta información convirtiéndola en texto o enviando órdenes que realicen alguna acción.

Existen dos formas de reconocer el habla: el reconocimiento dependiente del usuario, y el reconocimiento independiente del usuario los cuales explicamos a continuación .

Reconocimiento dependiente: el sistema sólo reconoce el voz de una persona determinada.

Reconocimiento independiente: un sistema puede reconocer el voz de cualquier

persona.

Los sistemas de reconocimiento del habla se están utilizando en aplicaciones telefónicas: atención al cliente, agencias de viajes, información etc. esto es gracias a que los sistemas de reconocimiento de voz han ido mejorando los últimos años.

### **2.1.2 Microcontroladores**

El microcontrolador es un sistema que se forma en un propio circuito integrado, una memoria, unidad de procesamiento y pines para la salida y entrada de datos. Se podría pensar que un microcontrolador es como un pequeño computador.

El primer microcontrolador fue inventado por Texas Instruments en el año 1971. El nombre designado al microcontrolador fue TMS 1000, este trabajaba a 4 bits y tenía un tamaño de  $3 \times 3,6$  mm (Stan Augarten, 1998 - 2009).

Actualmente hay una gran cantidad de fabricantes de microcontroladores. En la variedad de productores que existe cada uno de estos fabricantes toma sus decisiones en cuanto las características de los microcontroladores que fabrican. Estas decisiones vienen dadas según el sector o propósito al que estén orientados. Por lo que, existe una gran variedad en todo lo que a las arquitecturas, los lenguajes en los que se programan y el tipo de funcionalidades que se incluyen dentro del integrados. Al presente, hay microcontroladores que trabajan en desde 4 a 64 bits. (Toshiba, 2014).

### **2.1.3 Arduino**

Arduino es una plataforma de prototipo de código abierto que nació en Italia en el año 2006. Arduino se forma de una placa que se fundamenta en un microcontrolador, y una interface de desarrollo para escribir el software libre para

esta placa. Tanto los diseños del hardware como el código de su IDE están disponibles bajo licencias de tipo código abierto.

Arduino describe muy bien su propósito:

"Arduino es una herramienta para hacer ordenadores que puedan sentir y controlar el mundo físico a su alrededor [...]". (Arduino, Arduino, 2014)

Arduino oculta y simplifican los detalles de programación de los microcontroladores, los que se basan, al mismo tiempo proporcionan interfaces de entrada /salida comunes como puertos serie o puertos USB. De esta forma, estas plataformas hacen más fácil y asequible el comenzar en el uso de los microcontroladores.

Existen otras interfaces de salida y entrada que las placas de Arduino poseen. Una de las características más importantes de Arduino es la disposición estándar de estos pines. Esta disposición estándar permite conectarlo a otros módulos que aportan nuevas características al controlador. A los módulos que integran esta disposición estándar se les llama "Shields".

### **2.1.3.1 Arduino Mega**

Se trata de una placa electrónica basada en el microcontrolador ATmeg1280, cuenta con 54 entradas digitales, 4 UARTS, un cristal oscilador de 16MHz, conexión USB, conexión ICSP y botón de reset. El Mega es compatible con la mayoría de shields diseñados para el Arduino Duemilanove o Diecimila.

### 2.1.4 EasyVR

Es un potente módulo de reconocimiento de voz diseñado para ser utilizado en cualquier aplicación que se requiere el reconocimiento de voz. Se puede utilizar con cualquier entrada por lo que tiene un puerto serie RS-232.

### 2.1.5 Motores de Corriente Alterna

Son motores monofásicos con inversión de giro, estos funcionan a 110 [V] y tienen un consumo de 6 [A], estos motores ya viene con cuatro pines de conexión los cuales son el pin de fase, neutro, tierra y el de inversión de giro, las conexiones de estos se los realizará a través de un mando general mismo que se conectara en el sistema que se va a implementar.

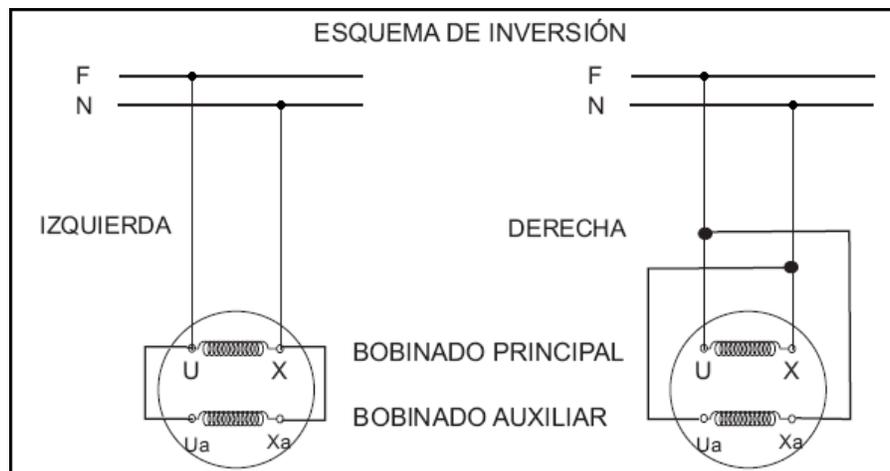


Figura 2.1 Diagrama Conexión Motores

Fuente: (Yo Reparó , 2014)

### 2.1.6 Fritzing

Es un software de automatización de diseño electrónico libre que ayuda a diseñadores para que se pueda pasar de prototipos a productos finales.

Este programa servirá para el diseño esquemático del proyecto.

Fritzing fue creado bajo los principios de Processing y Arduino, que permite a los diseñadores, artistas e investigadores documentar los prototipos basados en Arduino; crear esquemas de circuitos impresos para su posterior fabricación. Además cuenta con un sitio web complementario que ayuda a compartir y discutir bosquejos y experiencias y a reducir los costos de fabricación.

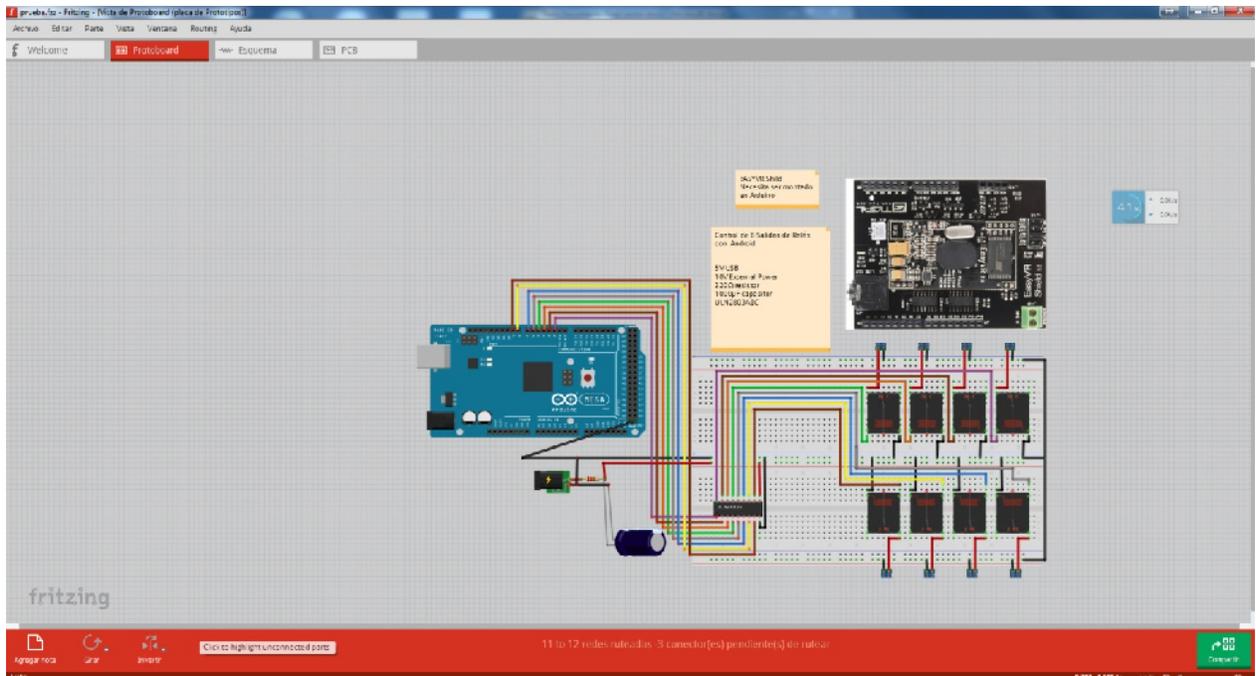


Figura 2.2 Captura de pantalla de Fritzing

Fuente: Autor

### 2.1.7 Audacity

Es un programa multiplataforma libre, que se puede usar para grabación y edición de audio, distribuido bajo la licencia GPL.

Es el editor de audio más difundido en los sistemas GNU/Linux.

Este software se utilizará para corregir y convertir el audio, de tal forma que sea compatible para que sea subido al EasyVR.

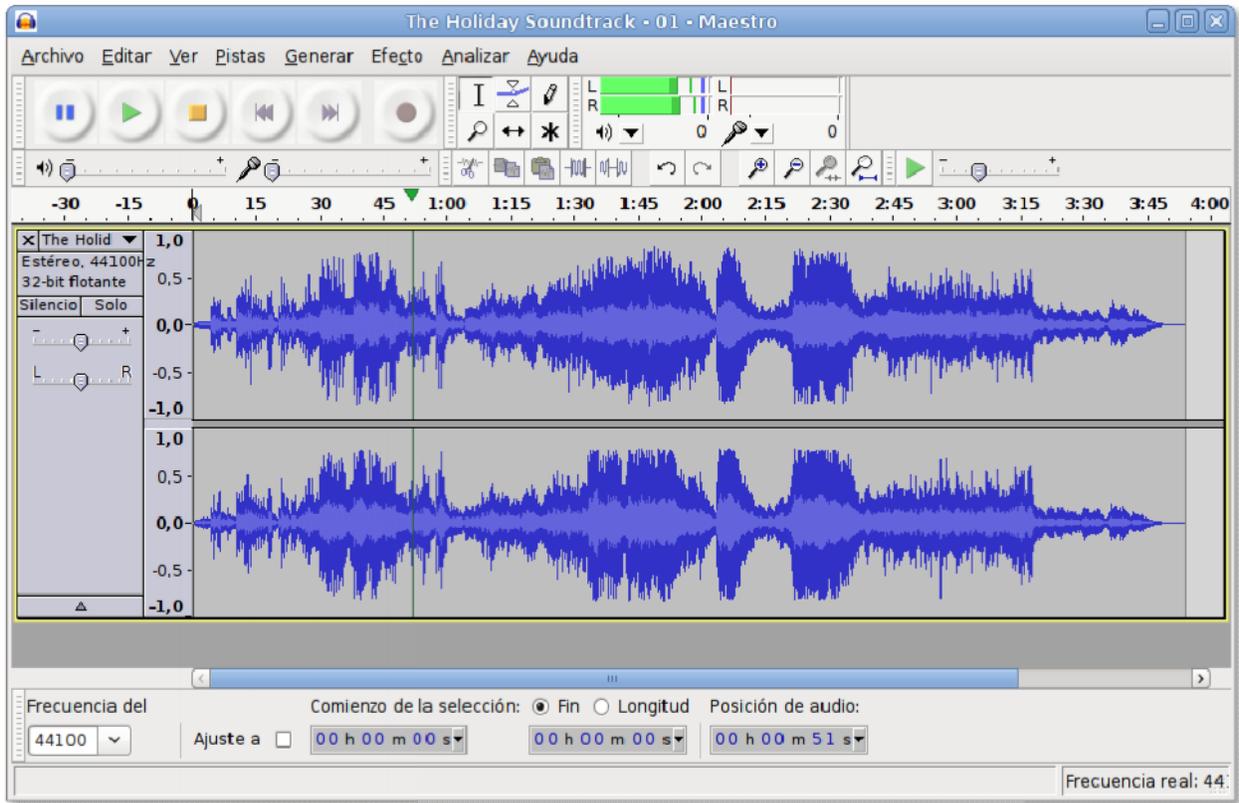


Figura 2.3 Captura de pantalla de Audacity

Fuente: Autor

### 2.1.8 Balabolka

Es un software que pasa de Texto a Voz (Text-To-Speech, TTS). Las voces que se instala en el Sistema Operativo están disponibles para Balabolka. El texto que se escriba en el programa puede ser guardado como un archivo de audio en formato WAV, MP3, OGG (formato de audio libre) o WMA.

Este software fue utilizado para convertir el texto a voz para luego realizar la tabla de sonidos que se implementará en el proyecto.

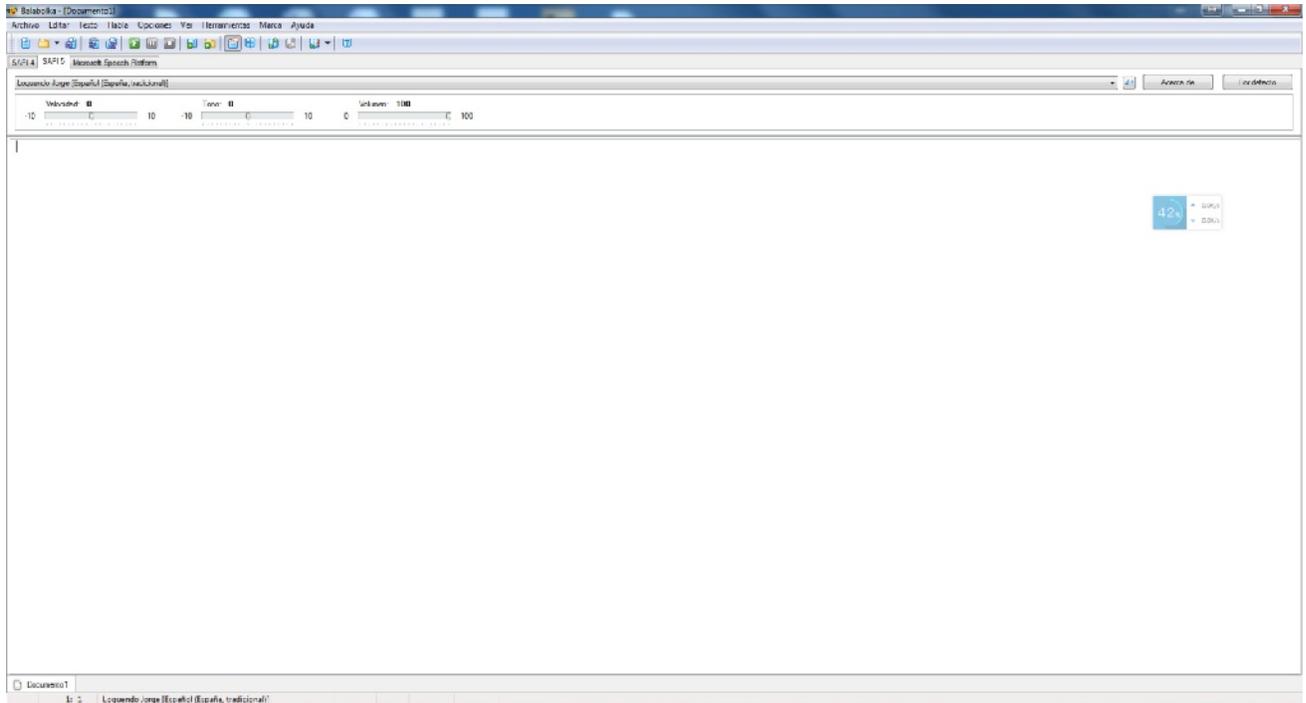


Figura 2.4 Captura de pantalla de Balabolka

Fuente: (Autor)

## 2.2 Marco Conceptual

En esta sección se presenta la descripción de los elementos a utilizarse, tablas de comparaciones y por qué la elección de los elementos que componen el sistema de automatización de cama hospitalaria.

## 2.2.1 EasyVR

El micrófono suministrado con el módulo EasyVR es un micrófono de condensador omnidireccional. Respuesta de sensibilidad de 38dB, 2.2k impedancia de carga, voltaje de funcionamiento de 3,3 [V] y la frecuencia en el rango de 100 Hz a 20 kHz.

Este módulo de reconocimiento de voz y reproducción de sonidos tiene un costo no muy alto. El sistema tiene prestaciones con comandos de voz ya predefinidas y grabadas internamente. Este consigue grabar y reconocer voces dadas por el usuario y en varios idiomas. Ver la figura 2.5.

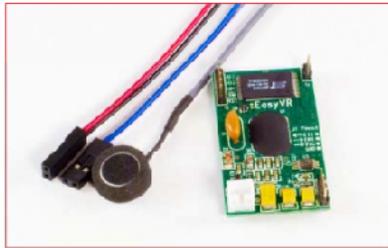


Figura 2.5 El módulo de reconocimiento EasyVR

Fuente: (VEEAR, 2014)

### 2.2.1.1 Aplicaciones

Hay varias aplicaciones del módulo de reconocimiento EasyVR. A continuación se detalla algunas:

- Sistemas de control gobernados mediante voz
- Control de acceso por voz
- Sistemas robóticos controlados por voz

### 2.2.1.2 Características

Se destaca las más relevantes:

- 26 instrucciones y voces pre programados: en 6 idiomas diferentes.
- 32 instrucciones y voces grabadas por el usuario en cualquier idioma.

- Tiene una interface gráfica de usuario (GUI) para Windows
- Se puede conectar con cualquier tipo de controlador mediante comunicación serie.
- Alimentación de 3.3V a 5V
- 3 líneas de Entrada/Salida que se controlan directamente mediante los comandos propios.
- Salida PWM de audio con una conexión para altavoz de 8  $\Omega$
- Reproducción de 9 minutos de sonidos y/o voces dependiendo de la calidad del audio.

### **2.2.2 EasyVR Commander: INTERFACE GRÁFICA DE USUARIO (GUI)**

Se utiliza para configurar los comandos necesarios en el módulo EasyVR para llevar a cabo su configuración. El EasyVR debe estar conectado a la PC a través de la Arduino o utilizando una placa adaptadora.

En el software, el usuario puede definir un nuevo disparo, grupos de comandos, los sonidos y las contraseñas. Basado en el discurso grabado es posible generar cualquier acción como encender un comando.

En el software, el usuario debe definir un nuevo comando, grupos de comandos, los sonidos y las contraseñas. Basado en el comando grabado es posible generar cualquier acción como encender una salida.



Figura 2.6 El área de trabajo EasyVR GLI

Fuente: (Autor)

### 2.2.2.1 Configuración

Se elige el puerto de comunicación y una vez establecida la conexión se puede realizar los ajustes necesarios. Hay cuatro tipos de posibles comandos en el software, que son:

- Trigger - Es un grupo comando especial, sirve como un disparador para realizar una acción. Es un comando dependiente que puede ser cambiado por el usuario, pero se ha registrado previamente con el comando "Robot" independiente.
- Group - son grupos de comandos dependientes utilizadas por el usuario para definir las acciones.
- Password - Se trata de un grupo especial de comandos depende de hasta cinco contraseñas voces.
- Wordset - es un grupo de 22 comandos independientes con acciones predefinidas que se pueden utilizar en 6 idiomas, inglés, italiano, alemán, francés, español y japonés.

Además, el SoundTable es un grupo donde se puede introducir audio [wav] en el módulo y reproducirlo a través del altavoz EasyVR.

Cuando EasyVR interfaz gráfica de usuario se conecta al módulo se leen todos los comandos y los grupos definidos por el usuario, se almacena en la memoria no volátil EasyVR. Un comando debe ser entrenado dos veces con la voz del usuario, como se ve en Figura 2.7.

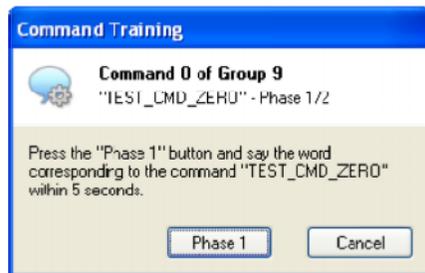


Figura 2.7 Ventana de Grabación del comando.

Fuente: Autor

### 2.2.1.1 Especificaciones técnicas/eléctricas

La figura 2.8 muestra las dimensiones del módulo EasyVR así como las etiquetas de los pines de conexión.

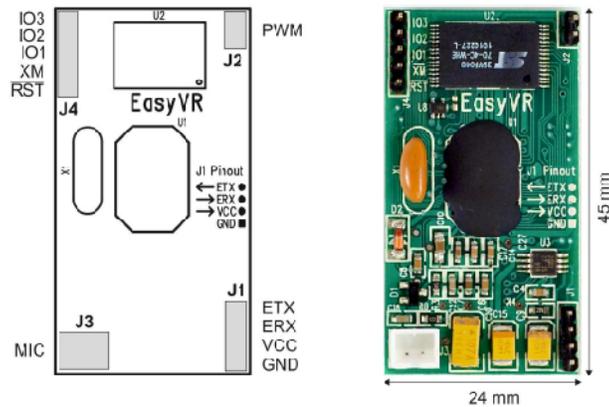


Figura 2.8 Distribución de pines del EasyVR

Fuente: (VEEAR, 2014)

En la Tabla 2.1 se muestra la distribución de pines del EasyVR, en la que se muestra un detalle del nombre del pin, su funcionalidad y la descripción de cada uno.

CONECTOR	Pin N°	NOMBRE	TIPO	DESCRIPCIÓN
J1	1	GND	-	Tierra de alimentación
	2	VCC	E	Entrada de alimentación
	3	ERX	E	Recepción serie de datos (niveles TTL)
	4	ETX	S	Transmisión serie de datos (niveles TTL)
J2	1-2	PWM	S	Salida PWM. Puede conectarse desde una salida directa a un altavoz de 8Ω
J3	1	MIC_RET	-	Tierra para el micrófono
	2	MIC_IN	E	Entrada de señal del micrófono
J4	1	/RST	E	Reset asíncrono del módulo activo por nivel "0". Está conectada internamente con una resistencia Pull - Up de 100 K
	2	/XM	E	Selección del modo de arranque del módulo. Internamente está conectada con una resistencia Pull - Down de 1000
	3	IO1	E/S	Línea de E / S de propósito general (nivel TTL de 3.0 Voltios )
	4	IO2	E/S	Línea de E / S de propósito general (nivel TTL de 3.0 Voltios )
	5	IO3	E/S	Línea de E / S de propósito general ( nivel TTL de 3.0 Voltios )

Tabla 2.1: Distribución de pines módulo de reconocimiento de voz

Fuente: (S.L, 2012)

#### Condiciones de trabajo recomendadas

SIMBOLO	PARAMETRO	Mín.	Típ.	Máx	UNIDAD
VCC	Tensión de alimentación	3.3	5	5.5	[V]
Ta	Temperatura ambiente de trabajo	0	25	70	°C
ERX	Recepción de datos serie	0	-	VCC	[V]
ETX	Transmisión de datos serie	0	-	VCC	[V]

Tabla 2.2: Condiciones de Trabajo Recomendadas

Fuente: (S.L, 2012)

La Tabla 2.3 muestra las características eléctricas en la cuales va a funcionar el módulo del EasyVR, en esta se detallan voltajes máximos y mínimos de funcionamiento:

SIMBOLO	PARAMETRO	Mín.	Típ.	Máx	UNIDAD
VIH	Tensión de entrada del nivel "1"	2.4	3.0	3.3	[V]
VIL	Tensión de entrada del nivel "0"	-0.1	0.0	0.75	[V]
IIL	Corriente de fuga de entrada		<1	10	µA
VOH	Tensión de salida del nivel "1"	2.4			[V]
VOL	Tensión de salida del nivel "0"			0.6	[V]

Tabla 2.3: Características eléctricas  
Fuente: (S.L, 2012)

La Tabla 2.4 detalla los valores de consumo de corriente ya que el módulo EasyVR está funcionando en modo normal, o se encuentre reproduciendo audio.

SIMBOLO	PARAMETRO	Mín.	Típ.	Máx	UNIDAD
ISLEEP	Consumo en el modo Sleep	<1			[mA]
IOPER	Consumo en el modo normal de trabajo	12			[mA]
ISPEAKER	Consumo en reproducción de audio	180			[mA]

Tabla 2.4: Consumo de corriente Eléctrica  
Fuente: (S.L, 2012)

En la Tabla 2.5 se muestra la comparativa de productos VEER de reconocimiento de voz en la cual se puede observar las características técnicas del EasyVR Escudo y el SmartVR, por el diseño de Shield y por muchas otras prestaciones adicionales se escogió el EasyVR Shield 2.0

	EasyVR 2.0 / EasyVR Escudo 2.0	SmartVR <b>OBSOLETO - No se recomienda para nuevos diseños!</b>
Independiente de aplicaciones	No	Sí
Módulo Esclavo	Sí	Sí
Built-in altavoces Comandos Independientes	Sí - incluye 26 incorporado en los comandos independientes de altavoz para robóticos applications.Supported Idiomas: Estados Unidos Inglés Italiano Alemán Francés Español Japonés	No hay comandos integrados incluidos.
Altavoz Dependiente de Apoyo al Mando	Hasta 32 Usuario Formado comandos. Soporta cualquier idioma.	Hasta 256 Comandos de usuario capacitado.Soporta cualquier idioma.

Speaker Verification (contraseñas de voz biométrica)	Hasta 21 contraseñas	Hasta 256 contraseñas
Soporte Sonicnet	Sí	No
DTMF Generación de tono	Sí	Sí
CC externa Voltaje de entrada	3.3V - 5V	2.7V - 3.6V
Corriente del sueño	<1mA	<1mA
Corriente de funcionamiento	típica 12 mA	típica 11 mA
GPIO	3	16
Código / CostFlash	No	512kB
Flash de datos (de serie)	No	512kB
EEPROM de datos	8kB	Ninguno
A bordo de RAM	No	128kB
SPI Interface	No	Sí
Selección de chip Salidas	No	5
Memoria Activa Salidas	No	2
Leer-Escribir bus de memoria	No	8-bit de ancho
Chip / Controlador	RSC-4128	RSC-4128
I2C	No	Sí
Grabación de mensajes de audio	No	Sí
Tiempo de reproducción	Hasta 9 minutos	Hasta 9 minutos
Salida de audio	PWM	DAC, PWM

Tabla 2.5: Tabla comparativa de de Productos VEER  
Fuente: (VEEAR, 2014)

### 2.2.2 Arduino Mega

Es una placa electrónica muy completa con 54 E/S Digitales de las cuales 14 pueden ser PWM, a continuación se muestra las especificaciones más importantes de esta placa:

Microcontroladores	ATmega2560
Tensión de funcionamiento	5 Voltios
Tensión de entrada ( recomendado )	7-12 Voltios
Tensión de entrada (límites)	6-20 Voltios
Digital I / O Pins	54 ( de los que 15 proporcionan salida PWM )
Pines de entrada analógica	16
Intensidad para los pines E / S Pin	40 miliAmperios
Intensidad para los pines 3.3V	50 miliAmperios
Memoria Flash	128 KB de los cuales 4 KB del arranque del módulo Arduino.
Memoria SRAM	8 KB
Memoria EEPROM	4 KB
Velocidad del reloj interno	16 Mega Hertz

Tabla 2.6: Especificaciones Arduino Mega  
Fuente: (Arduino, Arduino, 2014)

La placa de Arduino tiene 54 entradas y salidas de las que no se utilizaran todas, se utilizará los pines de comunicación para conectar en el módulo EasyVR y los pines que se utilizarán como salidas digitales que enviarán las señales a los optoacopladores que activarán los relés, se utilizó este módulo por la capacidad de almacenamiento, velocidad de procesamiento y por lo amigable que es el entorno de Arduino ya sea en software como en hardware.

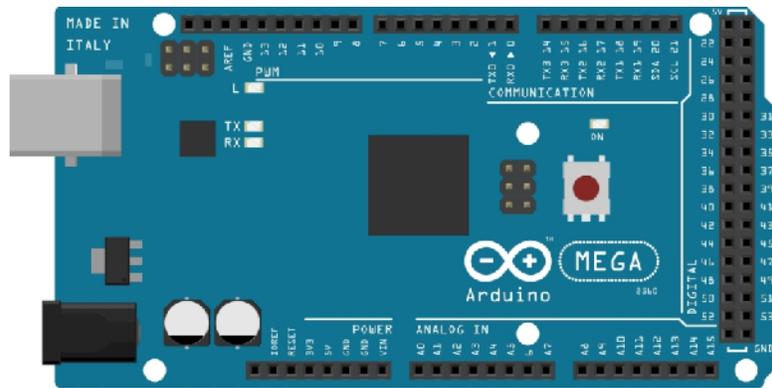


Figura 2.9: Circuito Arduino Mega  
Fuente: (Arduino, Arduino, 2014)

El diagrama circuito electrónico de la Figura 2.7 se observa las conexiones del Microcontrolador Atmega2560 con todos sus elementos, partes y circuitos que hace que función Arduino Mega y que esta sea una plataforma muy útil y fácil de usar.

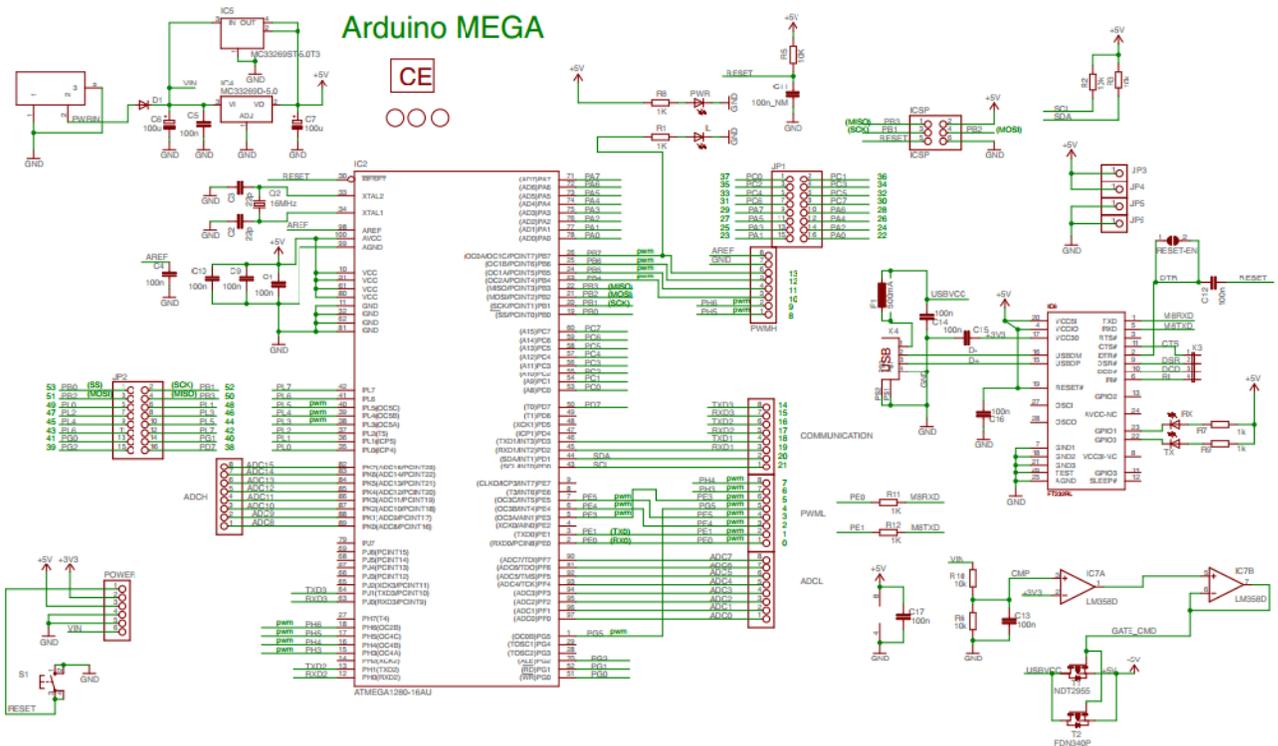


Figura 2.10: Diagrama Electrónico de Arduino Mega  
Fuente: (Arduino, Arduino, 2014)

### **2.1.3.1 Potencia**

Puede ser alimentado a través de la conexión USB o una fuente externa de voltaje DC. La alimentación se selecciona automáticamente. La tarjeta puede funcionar con un voltaje externo de 6 a 20 [V]. Si se ingresa un voltaje menor de 7[V], el pin de 5[V] puede dar menos de 5 [V] y esto puede hacer que el circuito se torne inconsistente. Si se utiliza más de 12 [V], el regulador de voltaje se puede sobrecalentar y dañar la tarjeta Arduino o los módulos acoplados a este. Teniendo un rango recomendado de voltaje 7 a 12 [V].

### **2.2.3 Cama Hospitalaria Eléctrica - Carroll Healthcare - Echo**

Se investigó las especificaciones del modelo de la cama hospitalaria. Esta cama de marca Carroll Healthcare de procedencia Canadiense es una cama eléctrica que funciona con pulsadores, los que se encuentran en la parte posterior, y un interruptor general que enciende todo el circuito de la cama, consta de dos tarjetas, una de fuerza, otra de control y comunicación.



Figura 2.11: Cama Hospitalaria Electrica

Fuente: (Eco Bed Parts, 2014)

Las especificaciones dadas por el fabricante son las que se detalla a continuación en la Tabla 2.7, estas especificaciones son útiles para el momento de la implantación del sistema y para el buen uso de este.

Rango Ángulo de la cabeza	0° a 70°
Capacidad de peso recomendado	450 libras.
Grosor del colchón recomendado	6 "
Diámetro de las Ruedas	3 "
Tamaño del colchón recomendado	36 "x 76" o 36 "x 80"

Tabla 2.7 Especificaciones de la Cama Eléctrica  
Fuente: (Eco Bed Parts, 2014)

Las dimensiones de la cama hospitalaria fueron tomadas con un flexometro y escrita en la tabla 2.8.

<b>Modelo</b>	AD974
<b>Largo Cama</b>	200 cm
<b>Ancho Cama</b>	90 cm
<b>Altura regulable</b>	de 40 a 80 cm

Tabla 2.8 Tabla de medidas  
Fuente: Autor

En la Figura 2.12 se observa los movimientos que puede realizar la cama hospitalaria, tiene una altura mínima de subida de 40 centímetros y una altura máxima de subida de 80 centímetros, el ángulo máximo de subida para el espaldar es de 70° y el mínimo es de 0° .



## CAPÍTULO 3

### **ESTUDIO, DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA ELECTRÓNICO PARA AUTOMATIZACIÓN DE UNA CAMA DE HOSPITAL CONTROLADA POR VOZ PARA MOVILIZACIÓN DE PERSONAS CON CAPACIDADES ESPECIALES**

#### **Introducción**

En este capítulo se presenta el diseño, tanto en hardware y software planteando diagramas circuitales, simulaciones, diagramas de flujo y demás ítems que dan paso al proceso de montaje e implementación del sistema.

#### **3.1 Estudio sobre la implantación de sistemas de reconocimiento de voz para controlar sistemas electrónicos y electromecánicos.**

Se revisó formas de realizar el reconocimiento de voz, para implementación del proyecto con el costo más bajo y que funcione de la mejor manera para el usuario final que en este caso son personas cuadripléjicas o de la tercera edad.

Lo que se pudo destacar es que se puede realizar este reconocimiento de las maneras que se detallan a continuación:

- Módulo EasyVR
- Android
- Computador

El módulo de EasyVR es un potente módulo de reconocimiento de voz el mismo que se comunica vía serie, este tiene un costo relativamente accesible, una desventaja es que tiene la memoria limitada a 32 comandos programables.

El Sistema Android necesitaría la programación para el reconocimiento de voz, y si se utilizara la propia del sistema sería necesaria una conexión de internet, además de que el costo de equipo android (Tablet o Smartphone) agregaría un costo adicional al sistema; otra desventaja es que la persona cada vez que quisiera usar el sistema de reconocimiento de voz tendría que iniciar el sistema Android, además de presionar un botón lo cual si es una persona cuadripléjica no podría hacerlo.

Computador ya sea de escritorio o laptop, el costo se elevaría al de una computadora dedicada al uso del sistema, el consumo de energía eléctrica aumentaría, el tamaño del proyecto aumentaría considerablemente, el software para el diseño de este tipo de programas es costoso.

Por estas razones después del estudio de las diferentes formas en las que se puede realizar el reconocimiento de voz, se optó por el módulo del EasyVR que viene montada en un shield que se conecta directo con el módulo Arduino Mega.

### **3.1.1 EasyVR**

El módulo viene con siete idiomas predefinidos español, italiano, japonés, alemán, francés e inglés, esto no significa que no entienda otros idiomas, ya tienen ciertas palabras, ciertos comandos de voz incluidos dentro del firmware para el uso de los usuarios.

Los puertos que se va a utilizar son los del J3 (ver en Tabla 2.1) que es donde se va a conectar el micrófono y el J1 en el que se va a realizar la comunicación serie con Arduino que se componen en pines de transmisión, recepción y los de alimentación que son Vcc y GND que irán conectados a través del shield.

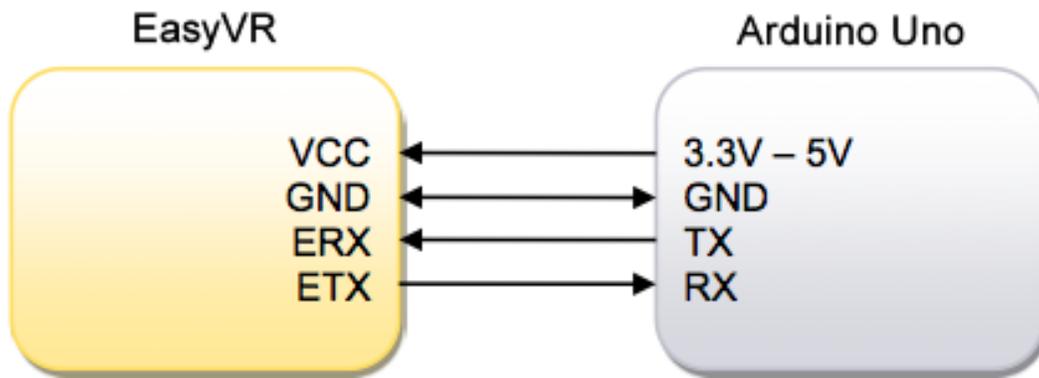


Figura 3.1 Conexión entre los módulos EasyVR y Arduino

Fuente: (VEEAR, 2014)

### 3.1.1.1 Micrófono

El micrófono suministrado con el módulo EasyVR es un micrófono de condensador electret omnidireccional (Horn EM9745P-382) con las siguientes características:

Sensibilidad -38 dB (0 dB = 1 [V] / Pa @ 1 KHz)

Impedancia de carga 2.2 K

Tensión de funcionamiento 3 [V]

Respuesta de frecuencia casi plana en el rango de 100Hz - 20kHz

Si se utiliza un micrófono con diferentes especificaciones en precisión de reconocimiento, puede verse afectada negativamente. Otro tipo de micrófono no es compatible con el módulo EasyVR.

Los comandos vocales se deben dar a una distancia optima de 60 cm desde el micrófono, se puede tratar a mayor distancia si se habla más fuerte.

### 3.1.1.2 Salida de Audio

La interfaz de salida de audio EasyVR es capaz de accionar directamente un altavoz de 8Ω. También se puede conectar a un amplificador de audio externo para conducir altavoces de menor impedancia.

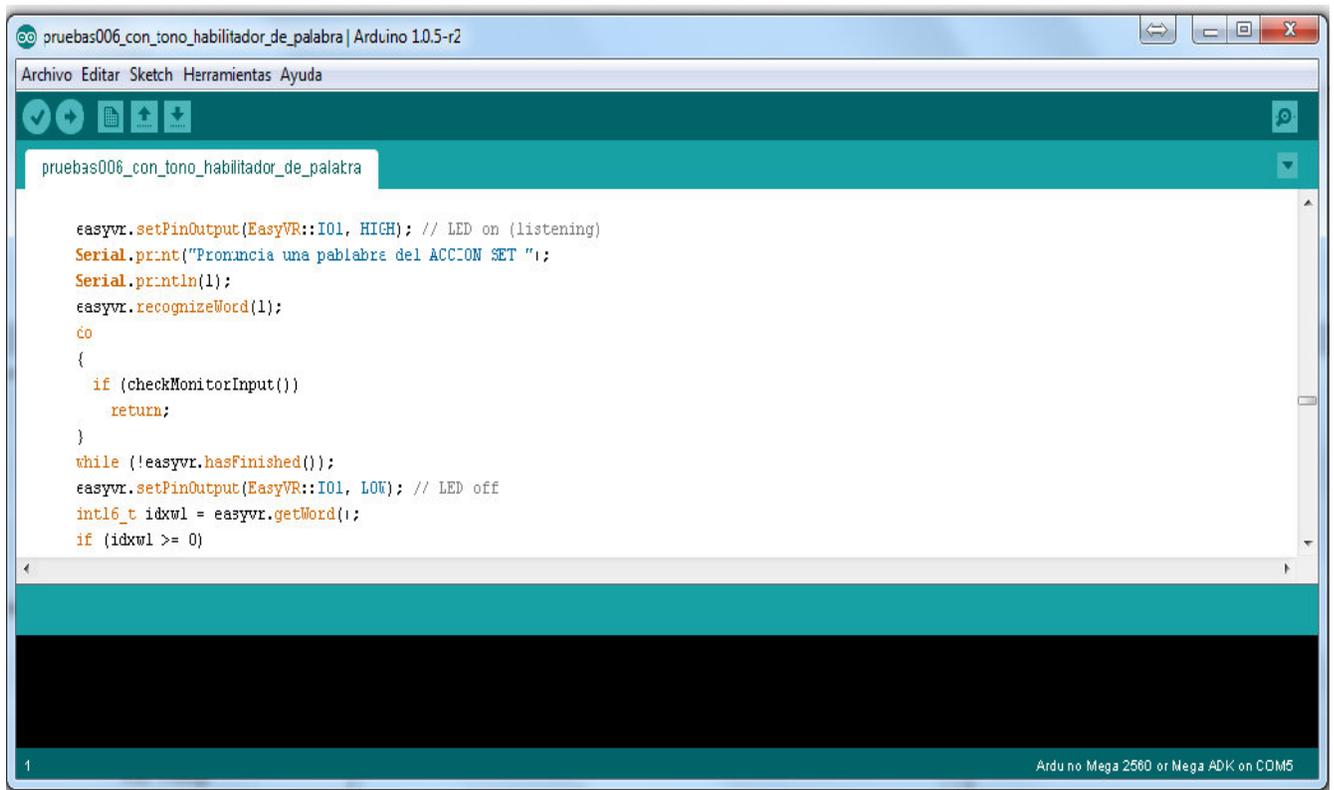
La conexión de altavoces con menor impedancia directamente al módulo, puede dañar permanentemente la salida de audio EasyVR o todo el módulo.

### **3.1.1.3 Construcción de conjuntos de comandos**

Para la programación de los comandos de voz se eligió las voces predefinidas que vienen grabadas de fábrica en el firmware y no pueden ser modificadas. Éstas están disponibles en diferentes idiomas: Italiano, Inglés, Japonés, Alemán, Español y Francés. Están almacenadas en los grupos denominados Wordset 1, Wordset 2, Wordset 3 y Trigger. La Tabla 3.1 muestra, según el idioma elegido, los grupos con las etiquetas o comandos (y las voces) que contienen. Para visualizarlas basta con seleccionar, en el área de grupos, el grupo que se va a utilizar.

En la Tabla 3.1 se muestra una lista de todos los comandos integrados para cada idioma soportado, junto con el índice de grupo (Trigger o wordset), el índice de comando (Comand Index) y un identificador de idioma (número superior de la tabla) el que ayuda a seleccionar el idioma del módulo EasyVR, la numeración en la tabla es usada en la programación y se usa como el *protocolo de comunicación*.

Para elegir una palabra en cualquiera de los idiomas que se muestra en la Tabla 3.1 se debe ubicar el wordset y el Comand Index con estos datos a través de comandos preestablecidos se comunican Arduino con el EasyVR, se configura primero el idioma y después se llama wordset que se lo encuentra en la primera columna y después el Comand Index que se encuentra en la segunda columna después se pone un tiempo de espera para que reconozca el comando, y por último lo valida si es un comando parecido se graba en una variable para que sea comparada, guardada y que posteriormente realice una acción.



```
pruebas006_con_tono_habilitador_de_palabra | Arduino 1.0.5-r2
Archivo Editar Sketch Herramientas Ayuda
pruebas006_con_tono_habilitador_de_palabra

easyvr.setPinOutput(EasyVR::I01, HIGH); // LED on (listening)
Serial.print("Pronuncia una pablbre del ACCION SET ");
Serial.println(1);
easyvr.recognizeWord(1);
do
{
  if (checkMonitorInput())
    return;
}
while (!easyvr.hasFinished());
easyvr.setPinOutput(EasyVR::I01, LOW); // LED off
int16_t idxw1 = easyvr.getWord(1);
if (idxw1 >= 0)
```

1

Ardu no Mega 2560 or Mega ADK on COM5

Figura 3.2 Comandos para llamar un wordset predefinido en EasyVR

Fuente:(Autor, Arduino UI)

En la Figura 3.2 se ve cómo comparar el resultado de un wordset en una variable, para lo cual se comunica via serie del Arduino al EasyVR a través de protocolos de comunicación ya establecidos por la librería montada en Arduino UI.

		Language						
		0	1	2		3	4	5
Trigger Word set	Command Index	English (US)	Italian	Japanese	(Rōmaji)	German	Spanish	French
0	0	robot	robot	ロボット	<i>robotto</i>	roboter	robot	robot
	0	action	azione	アクション	<i>acution</i>	aktion	acción	action
1	1	move	vai	進め	<i>susu-me</i>	gehe	muévete	bouge
	2	turn	gira	曲がれ	<i>magare</i>	wende	gira	tourne
	3	run	corri	走れ	<i>hashire</i>	lauf	corre	cours
	4	look	guarda	見ろ	<i>miro</i>	schau	mira	regarde
	5	attack	attacca	攻撃	<i>kougeki</i>	attaque	ataca	attaque
	6	stop	fermo	止まれ	<i>tomare</i>	halt	para	arrête
	7	hello	ciao	こんにちは	<i>konnichiwa</i>	hallo	hola	salut
2	0	left	a sinistra	左	<i>hidari</i>	nach links	a la izquierda	à gauche
	1	right	a destra	右	<i>migi</i>	nach rechts	a la derecha	à droite
	2	up	in alto	上	<i>ue</i>	hinauf	arriba	vers le haut
	3	down	in basso	下	<i>shita</i>	hinunter	abajo	vers le bas
	4	forward	avanti	前	<i>mae</i>	vorwärts	adelante	en avant
	5	backward	indietro	後ろ	<i>ushiro</i>	rückwärts	atrás	en arrière
3	0	zero	zero	ゼロ	<i>zero</i>	null	cero	zéro
	1	one	uno	一	<i>ichi</i>	eins	uno	un
	2	two	due	二	<i>ni</i>	zwei	dos	deux
	3	three	tre	三	<i>san</i>	drei	tres	trois
	4	four	quattro	四	<i>yon</i>	vier	cuatro	quatre
	5	five	cinque	五	<i>go</i>	fünf	cinco	cinq
	6	six	sei	六	<i>roku</i>	sechs	seis	six
	7	seven	sette	七	<i>nana</i>	sieben	siete	sept
	8	eight	otto	八	<i>hachi</i>	acht	ocho	huit
	9	nine	nove	九	<i>kyu</i>	neun	nueve	neuf
	10	ten	dieci	十	<i>jyuu</i>	zehn	diez	dix

Tabla 3.1 Resumen de comandos y voces predefinidos.  
Fuente: (VEEAR, 2014)

### 3.1.1.3 Tablas de sonido

En el módulo EasyVR se puede reproducir uno de los sonidos o frases guardados en su memoria flash interna. Un sonido "beep" predefinido también está siempre disponible, incluso cuando no hay sonidos que se han descargado en el módulo.

Los sonidos personalizados están organizados en una "tabla de sonido" que los usuarios pueden preparar y construir con la herramienta especial QuickSynthesisTM.

Si se necesita mayor información sobre cómo crear una tabla de sonidos hay que consultar el manual de esta aplicación que se encuentra en la página <http://www.veear.eu/downloads/> para obtener más detalles.

Para crear la tabla de sonidos se resume a continuación los siguientes pasos:

- Preparar los archivos de audio que desee incluir en la tabla de sonido en formato WAV sin compresión mono 22050 Hz de 16 bits. Para crear los archivos de sonido que puede utilizar un software gratuito como Audacity por ejemplo (<http://audacity.sf.net>)
- Abrir el software Sensorial QuickSynthesisTM 5 y crear un nuevo proyecto, especificando "familia RSC4"
- Añadir los archivos WAV que se deseen y especificar uno de los esquemas de compresión compatible
- Opcionalmente añadir frases, mediante la combinación de sonidos básicos WAV. Esto permite ahorrar memoria cuando se tiene archivos de audio de voz, si comparten algunas piezas (como "Digite" "Uno" + "Digite" + "Dos", y así sucesivamente)

- Generar el proyecto con QuickSynthesis™ y utilizar la configuración predeterminada ("Build módulo enlazable", "Load en el espacio CONST", "Load arriba o en: 0"). El software pedirá volver a comprimir los archivos de sonido nuevos o modificados, sólo hay que confirmar y proceder.
- Guardar el proyecto y construirlo de nuevo, por lo que el software Comant EasyVR verá que la construcción está actualizada.
- A continuación se presentan los formatos de compresión de audio soportados por el módulo EasyVR (de mayor a menor tasa de compresión):

Esquema de compresión de tiempo disponible (8 kHz 15% silencio) tiempo disponible (9.3kHz 15% silencio)

SX-2 8,7 minutos 7,5 minutos

SX-3 7,6 minutos 6,6 minutos

SX-4 6,8 minutos 5,9 minutos

SX-5 6,1 minutos 5,2 minutos

SX-6 5,6 minutos 4,8 minutos

4 bits ADPCM 87 segundos N / A

8-bit PCM 45 segundos 38 segundos

Para el archivo que contiene el discurso de audio, es una buena opción la compresión SX-3. Si se necesitara mayor calidad intentar tasas de compresión más bajas. Hay que tener en cuenta que debido a la frecuencia de muestreo utilizado, los archivos de audio no pueden contener frecuencias muy altas (menos de la mitad de la frecuencia de muestreo).



La Figura 3.5 muestra el esquema de conexión del funcionamiento de una entrada para el módulo de relés:

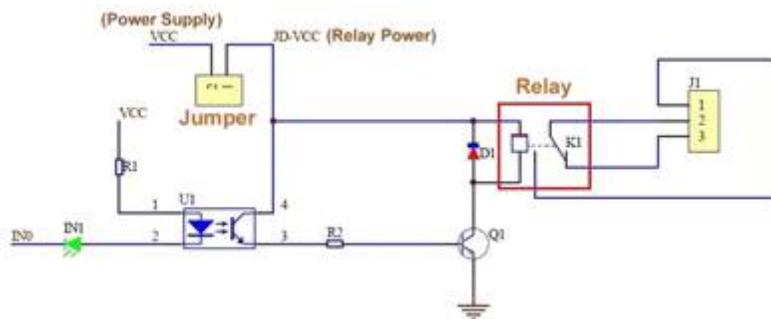


Figura 3.5 Esquema de conexión de un Relé.

Fuente: (Diagrama Relé, 2014)

En el módulo se tiene dos filas de pines, una que está el jumper, que tiene tres pines y se puede seleccionar: alimentar todo el módulo, o bien alimentar de forma autónoma tanto optoacopladores como relés (quitando el jumper). Se decidió usar la forma de alimentar todo de manera unida, ya que es la forma más sencilla de hacerlo, así que se ha dejado el jumper unido jd-vcc y vcc.

La otra fila de pines que presenta el siguiente orden: IN1 IN2 IN3 IN4 IN5 IN6 IN7 IN8 VCC GND, es donde se une la alimentación del Arduino, es decir, GND del módulo con GND del Arduino, VCC del módulo con 5 [V] del Arduino y las entradas IN1a IN8 (se activan las bobinas de los relés cuando se pone a GND 0 [V] a las entradas) se conecta los pines que se necesite en el Arduino; éstos se encargarán de activar las bobinas de los relés correspondientes. Esta es una manera sencilla de poder emplear relés en el proyecto sin necesidad construir una PCB.

### 3.1.1.5 Motores

Se revisó la placa de control a la que estaba conectada anteriormente los conectores de los motores, y se pudo notar que la placa estaba quemada y no existía la tarjeta de comunicación por lo que la cama electrónica no estaba funcionando, después de

analizar las entradas y salidas de los motores se pudo deducir la distribución de los pines, lo que se muestra en la Figura 3.6:



Figura 3.6 Distribución de Pines Conector Motor

Fuente: Autor

Se conectó los motores de manera exitosa y además se pudo deducir que los motores tienen un fin de carrera interno lo que hace que cuando esté en el nivel máximo o mínimo automáticamente los motores se apagan.



Figura 3.7 Dos Motores de Inversión de giro

Fuente: Autor

### 3.2 Diseño de una interface de control automático, utilizando reconocimiento de voz, añadiendo al programa del microcontrolador comandos que posibiliten el cumplimiento de los movimientos de una cama de hospital.

La Figura 3.8 muestra el diagrama de bloques general del proyecto en el cual se muestra la todas las partes del proyecto divididas en integración del EasyVR con Arduino Mega y el control del Hardware.

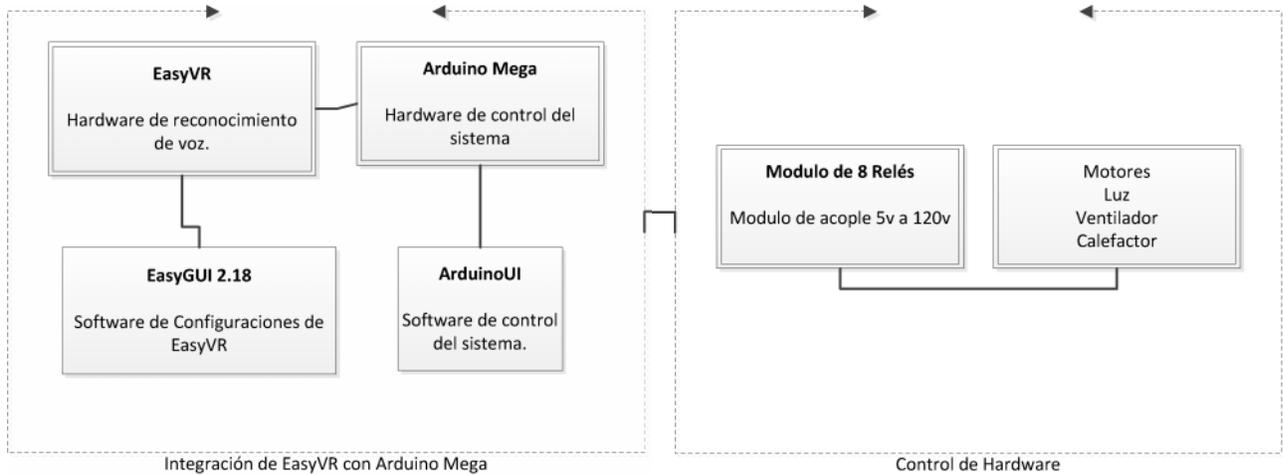


Figura 3.8 Diagrama de Bloques General del proyecto.

Fuente: Autor

El sistema que se propone se lo representa con un diagrama de bloques que se encuentra en la Figura 3.9, donde el sistema integra una interfaz de control utilizando reconocimiento de voz dividido en 4 fases; la primera fase utiliza un micrófono, la segunda fase consta de dos partes, la primera parte usa un entrenamiento de un vocabulario pequeño que construye un modelo y la segunda es una sección es la de reconocimiento de voz que utiliza este modelo, además de la fuente de alimentación que suministra voltaje a todo el sistema principalmente al microcontrolador pues este procesa todas la señales, en la fase 3 y 4 se encuentra un módulo de relés que se encargan de procesar la señales del microcontrolador del Arduino y activa los relés, los mismos que encenderán los motores, luz, ventilador y calefactor de la cama hospitalaria.

Este sistema se implementará con simples comandos de voz, y se diseñará para beneficiar a usuarios que tengan que realizar movimientos de la cama hospitalaria sin ayuda alguna.

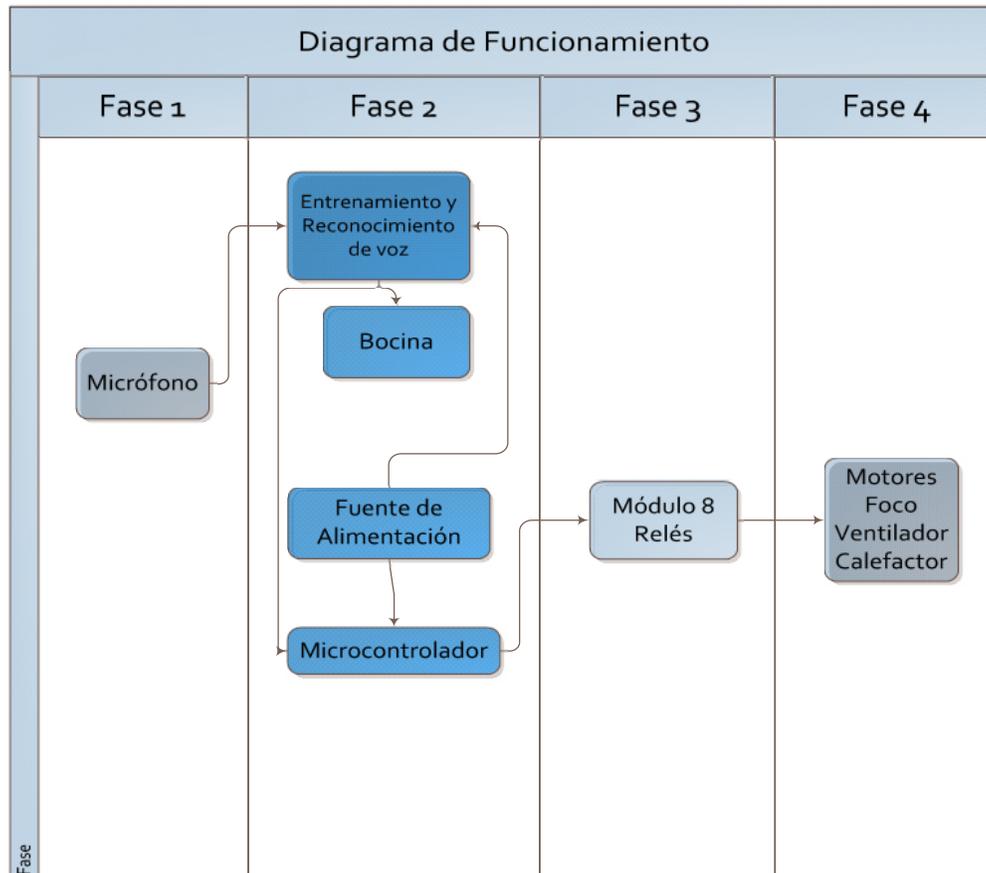


Figura 3.9 Diagrama de funcionamiento por etapas.

Fuente: (Autor)

### 3.2.1 Alimentación

La alimentación del sistema es de 120 [V] conectados directamente al tomacorriente, de este voltaje lo transformamos a través de un adaptador de 9 [V] y de 1 [A] voltaje que está dentro de los rangos para la conexión a Arduino, Arduino tiene reguladores de voltaje que transforma el voltaje de este adaptador.

Se puede conectar por puerto USB (5 [V]) o una fuente de alimentación externa de 6-20 [V]. Sin embargo, si la tensión de entrada es menor de 7 [V], el pin de 5 [V] puede suministrar menos de 5 [V] y la placa puede resultar inestable. Del mismo modo, si el voltaje es mayor que 12[V], el regulador puede sobrecalentarse y dañar la placa.

Los pines de alimentación son los siguientes:

- Vin: La entrada de alimentación del Arduino Mega cuando se utiliza una fuente de alimentación externa.
- 5[V]: La fuente de alimentación regulada utilizada por microcontrolador y demás componentes del Arduino. Se puede ingresar un VIN a través del controlador incorporado o la conexión USB o de otro voltaje regulado de 5 [V].
- GND: Pin tierra.
- 3V3: Una fuente de 3,3 [V] generada por el regulador interno que contiene la placa. La corriente máxima admitida es de 50 [mA].

### **3.2.2 Entrada y Salida**

Los 54 pines digitales de Arduino Mega pueden ser utilizados como funciones de entrada o de salida utilizando `pinMode ()`, `digitalWrite ()`, y `digitalRead ()`. Hay dos pines relacionados con la comunicación en serie, son los pines PIN 0 (RX) recepción y el PIN 1 (TX) transmisión. Estos pines están conectados en serie a los pines del chip de ATmega2560. También cuenta con un LED en el PIN 13, que cuando este pin es un valor ALTO este indicador LED se ilumina y cuando el pin está en BAJO el LED está apagado.

### **3.2.3 Arduino 3.2 UI**

Arduino Mega se programa con una interfaz propietaria llamada Arduino UI. Los ATmega2560 en el Arduino Mega vienen pregrabados con un gestor de arranque que le

permite enviar un nuevo código sin el uso de un programador de hardware externo. Se comunica manejando el protocolo STK500.

### 3.2.4 Configuración

Después de que el Arduino establece una comunicación USB con el PC, el Arduino UI debe estar configurado para el puerto COM correcto (Herramientas -> Serial Port). Después de esta configuración se debe seleccionar el modelo utilizando (Herramientas ->Board).

### 3.2.5 Programación

La programación se realiza en su propia interfaz, y tiene funciones específicas para el manejo de los puertos de entrada y salida y la configuración de hardware en general. Tiene varias bibliotecas para una funcionalidad adicional como SoftwareSerial, EEPROM, Ethernet entre otros. El lenguaje de programación utilizado es C / C + +.

La forma en que recibirá los comandos de voz será la que se puede ver en la Figura 3.10 con un tiempo de 2 segundos para poder decir el comando, caso contrario regresará a la parte inicial, hasta recibir un comando que lleve a una subrutina o que realice una acción.

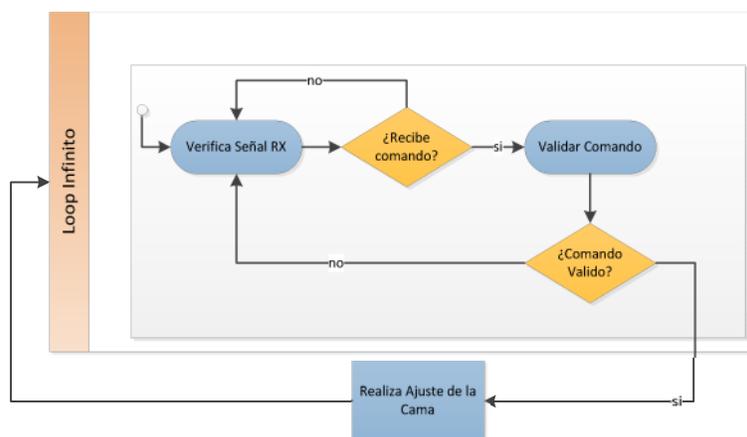


Figura 3.10 Flujograma de Funcionamiento de la programación

Fuente: (Autor)

### 3.2.6 Hardware

El Arduino Mega está conectado a través de USB a la PC. Para tal efecto es necesario instalar un controlador Serial-USB para que la conexión se realice correctamente.

Está conectado con el ETX y ERX EasyVR PIN 12 (RX) y PIN 13 (TX) (definido en el propio programa) Arduino Mega; los pines VCC y GND que alimentan al EasyVR provienen de salidas de alimentación que tiene el Arduino Mega de 5[V] y GND. El micrófono debe ser insertado en el conector EasyVR J3.

### 3.2.7 Integración de Arduino Mega con conexiones Eléctricas y de Control.

La Figura 3.11 presenta el diagrama circuital donde se puede observar que las bobinas de los relés son activadas por la señal de 7 puertos digitales los cuales activan los diferentes actuadores (Motores, Ventilador, Calefactor y Luz), siempre y cuando la señal de las salidas de Arduino estén en BAJO.

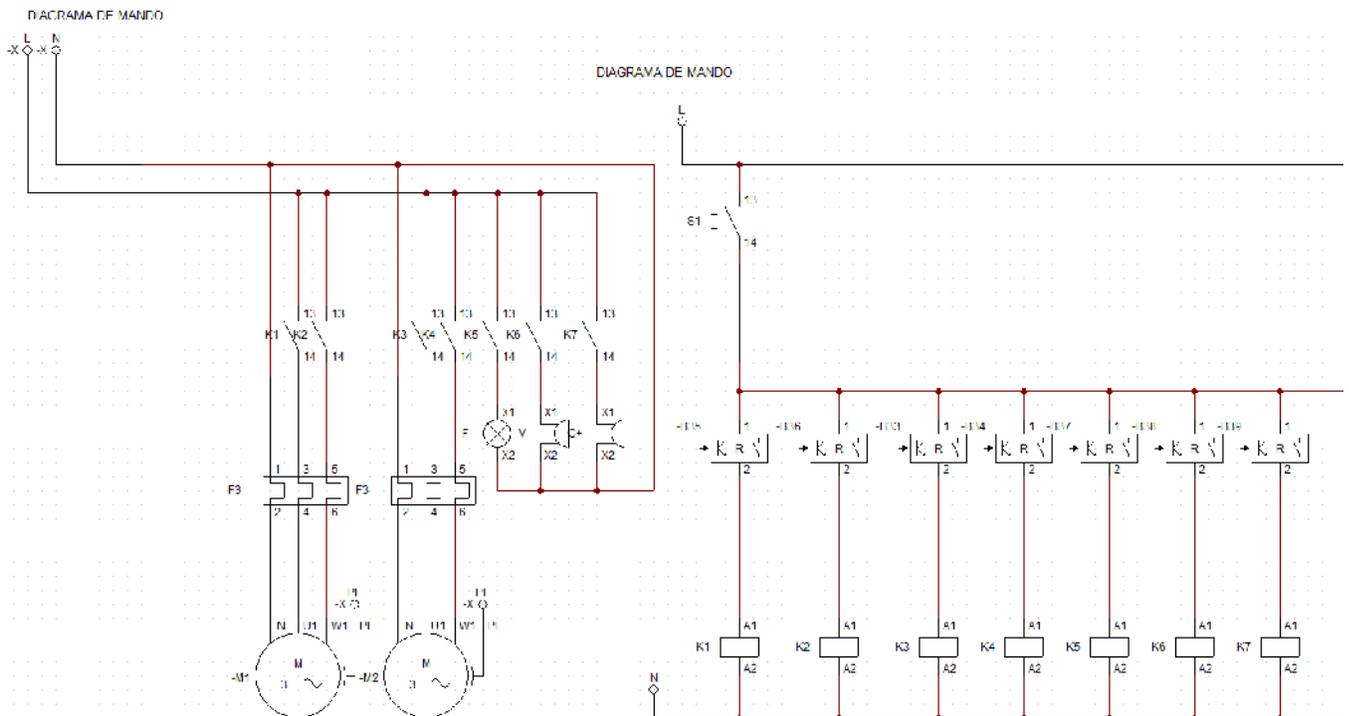


Figura 3.11 Diagrama Eléctrico de Control y Mando.

Fuente: (Autor)

### 3.2.8 Software

Para la comunicación con el módulo y el software EasyVR GUI se debe compilar el código en Arduino UI que se encuentra en el Anexo I y la comunicación se establecerá como comandos de voz. Para llamar a cualquier comando deberá utilizarse un protocolo ya establecido de líneas de programación en los cuales para cada comando se utilizará el índice correspondiente. El Command Group enviará un valor que el EasyVR volverá a Arduino y así se comprobará que el comando hablado es igual al registrado.

Se usaron los siguientes comandos en la programación:

		4
Trigger Word Set	Command Index	Spanish
0	0	robot
1	0	acción
	4	mira
	5	ataca
	6	para
2	2	arriba
	3	abajo
	4	adelante
	5	atrás

Tabla 3.2 Tabla de comandos del Wordset del EasyVR  
Fuente: (Autor)

Se utilizaron los comandos pre definidos para que cualquier usuario que hable español pueda utilizar, ya que si se usaba los comandos de voz programados, éstos sólo podían ser usados por la persona que realizó la grabación.

Estos comandos se los ordenó para que el paciente pueda usar la cama de una manera fácil y segura como se muestra en la Figura 3.12.

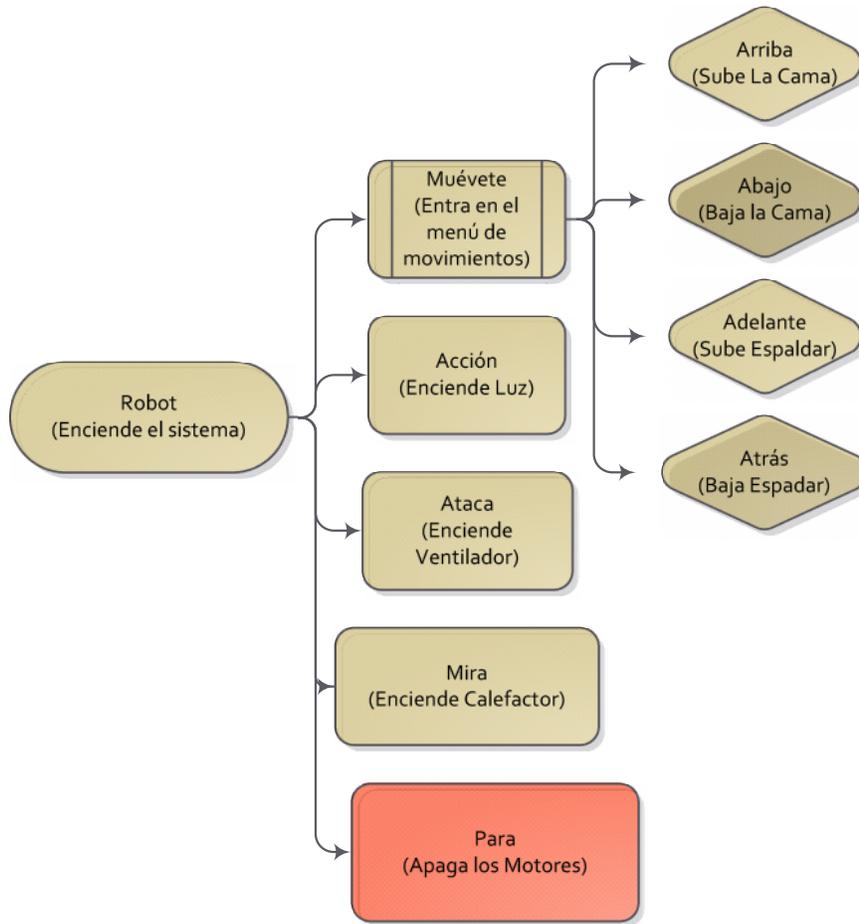


Figura 3.12 Flujograma de comandos de voz.

Fuente: (Autor)

Para comunicarse con el módulo EasyVR y la interfaz de usuario del software de Arduino se suben los archivos: CamaHospitalaria.ino, protocol.h, NewSoftSerial.he, NewSoftSerial.cpp en el software de Arduino.

En el código de la programación se encuentran: los pines que se utilizan y el ajuste de velocidad de transmisión para que exista una adecuada comunicación entre los módulos y el PC.

El programa va a controlar los comandos enviados por el módulo EasyVR y realizará las acciones necesarias en el módulo de control a través de Arduino.

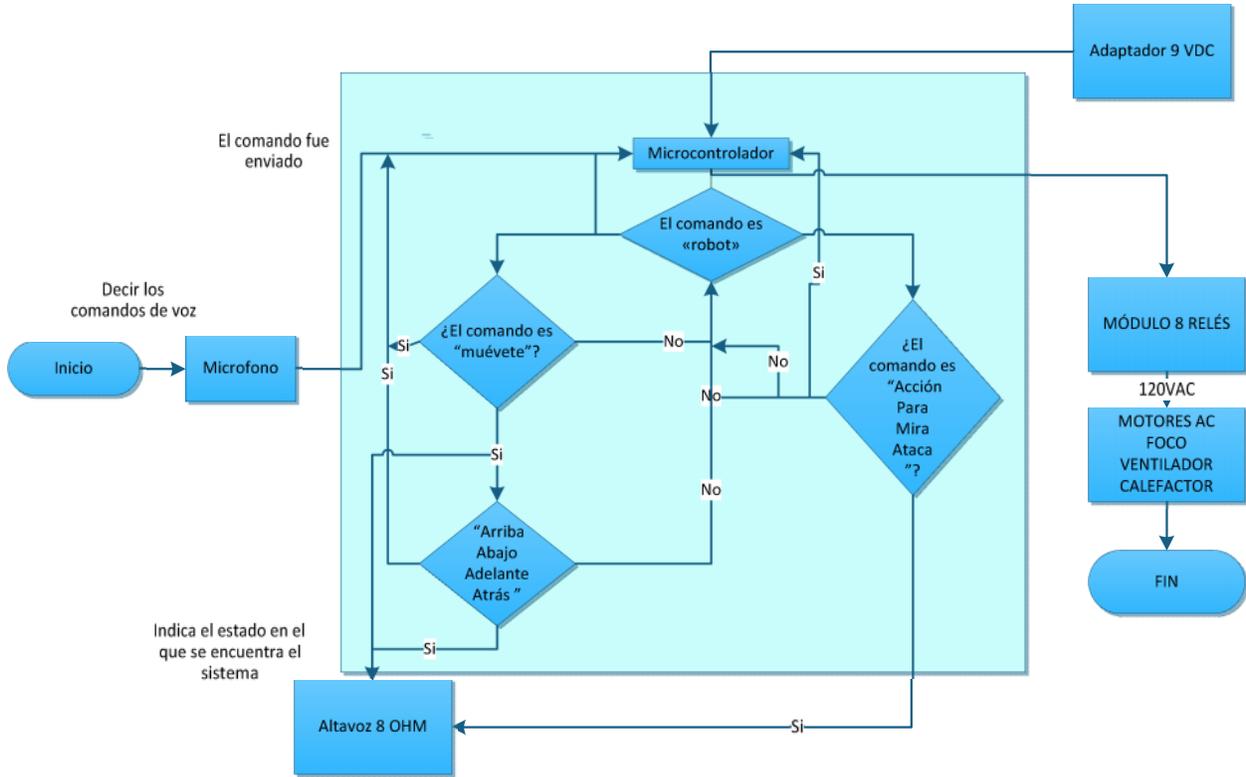


Figura 3.13 Diagrama de flujo de sistema de control por voz

Fuente: (Autor)

### 3.3 Construcción de un sistema automatizado capaz de obedecer a comandos de voz para movilizar una persona en una cama de hospital.

#### 3.3.1 Montaje Hardware

##### 3.3.1.1 Conexión del Micrófono en el módulo EasyVR

El Micrófono se lo conecta en el puerto J3.

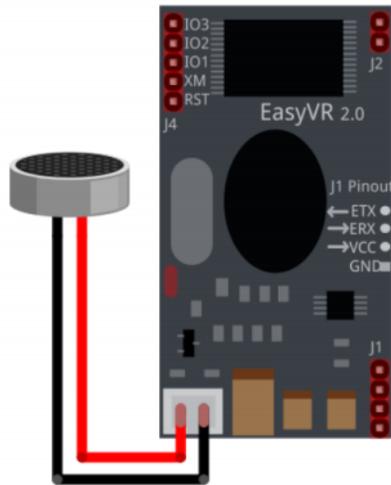


Figura 3.14 Conexión del Micrófono Módulo EasyVR.

Fuente: (VEEAR, 2014)

### 3.3.1.2 Conexión de Salida de Audio en el módulo EasyVR

La Salida de audio se la conecta en el pin J2 el parlante tiene que ser de 8  $\Omega$

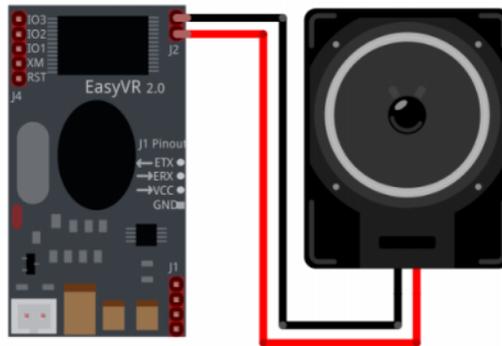


Figura 3.15 Conexión Parlante

Fuente: (VEEAR, 2014)

### 3.3.1.3 Conexión del módulo EasyVR en Arduino Mega

Como el módulo EasyVR es compatible con el Arduino Mega, simplemente hay que alinear A5 del Shield con el pin A5 de Arduino y hacer lo mismo con D0. Tal como se muestra en la Figura 3.16.

El módulo EasyVR tiene que ser programado y la tabla de sonido cargada con una configuración diferente para lo cual se debe cambiar el jumper al estado de: UP para subir la tabla de sonidos, PC para subir los comandos grabados y SW para utilizar el programa y los sonidos ya grabados.

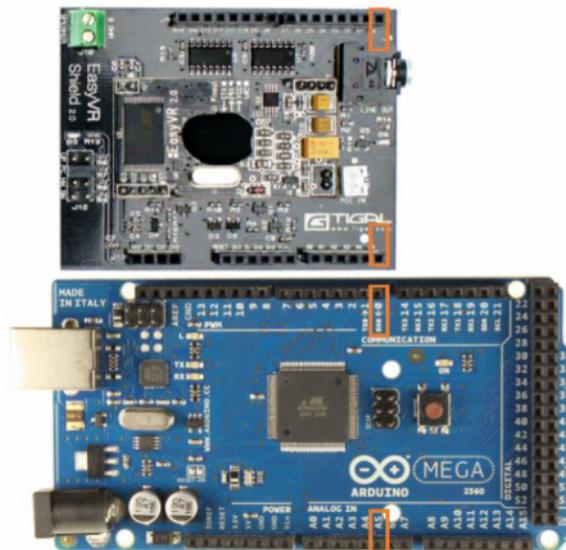


Figura 3.16 Conexión de EasyVR en Arduino Mega

Fuente: (Autor)

### 3.3.1.4 Conexión del módulo de 8 Relés

En la Figura 3.17 se muestra la conexión de los pines de salidas digitales del Arduino Mega que van conectados a las entradas del módulo de Relés.

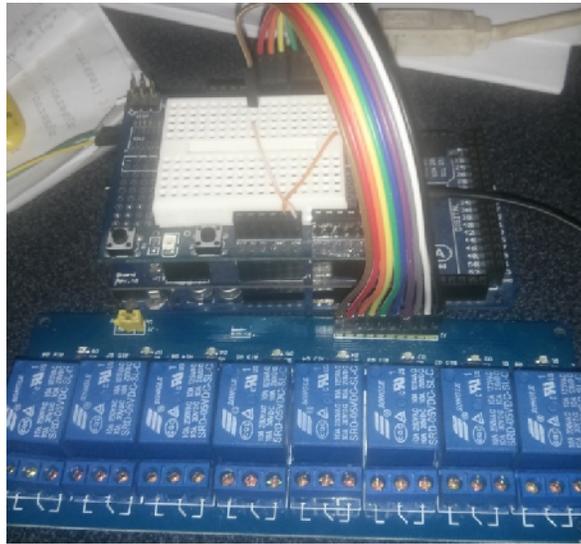


Figura 3.17 Conexión de Módulo de Relés.

Fuente: (Autor)

Se montó los módulos electrónicos de reconocimiento control y fuerza en un case metálico con base de plástico para mayor seguridad del usuario, ver Figura 3.18.

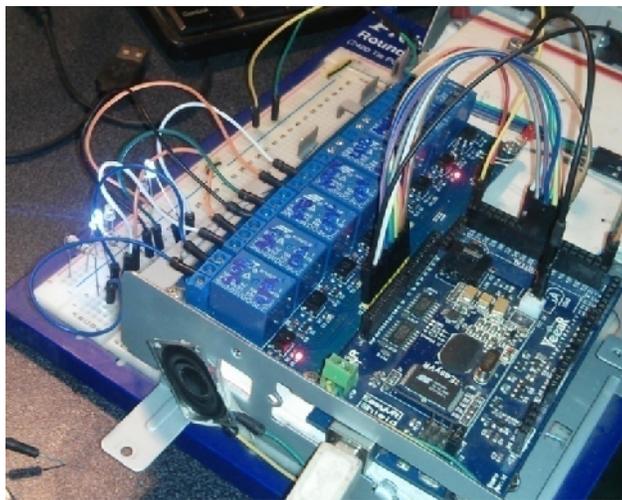


Figura 3.18 Montaje de Módulos

Fuente: (Autor)

### 3.3.1.5 Conexiones Eléctricas y de Control.

Se unió la parte electrónica con la electro-mecánica de acuerdo al diagrama de la Figura 3.11.

La Figura 3.19 muestra las conexiones realizadas en el sistema, uniendo la parte de mando y de control en un sólo sistema, se muestra los diferentes módulos y conexiones realizadas en el proyecto.

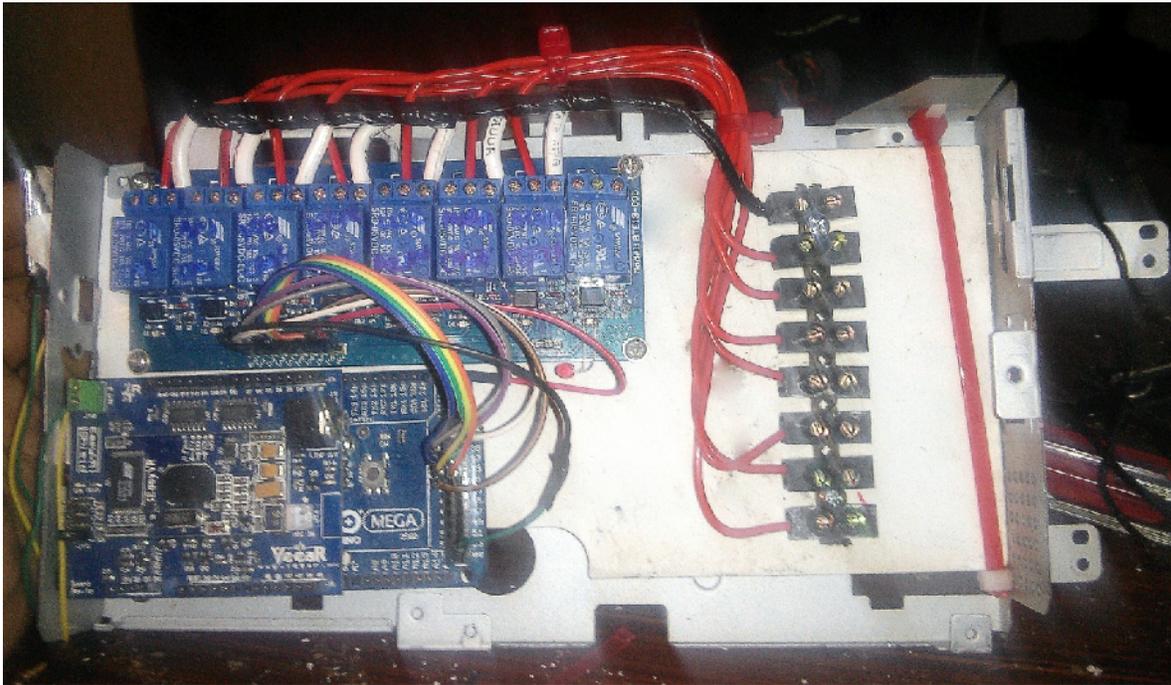


Figura 3.19 Conexiones de las cargas a los relés.

Fuente: (Autor)

### 3.3.2 Montaje de Software

El software está implementado en Arduino el mismo que se comunica con el módulo EasyVR para la transmisión y recepción de datos vía serie.

La programación se la hizo sobre Arduino, se agregó las bibliotecas de EasyVR para la comunicación con el módulo, los comandos utilizados en el programa son comandos ya establecidos en las bibliotecas del EasyVR.

El lenguaje de programación que se empleó es el propio de Arduino, la llamada de los comandos se la realizó a través de protocolos ya establecidos por el Módulo EasyVR.

### 3.3.2.1 Codificación Software

De acuerdo a los flujogramas de las figuras 3.10 y 3.12 se realizó la programación de Arduino, teniendo varias condicionales y variables para el funcionamiento del programa, en la programación también se realizó la comunicación serie con el EasyVR que tiene su propia librería de comandos que anterior a la programación ya fueron subidos, en la Figura 3.20 se puede ver una captura de pantalla de la aplicación y de la programación.

En el Anexo I se puede ver la codificación completa.

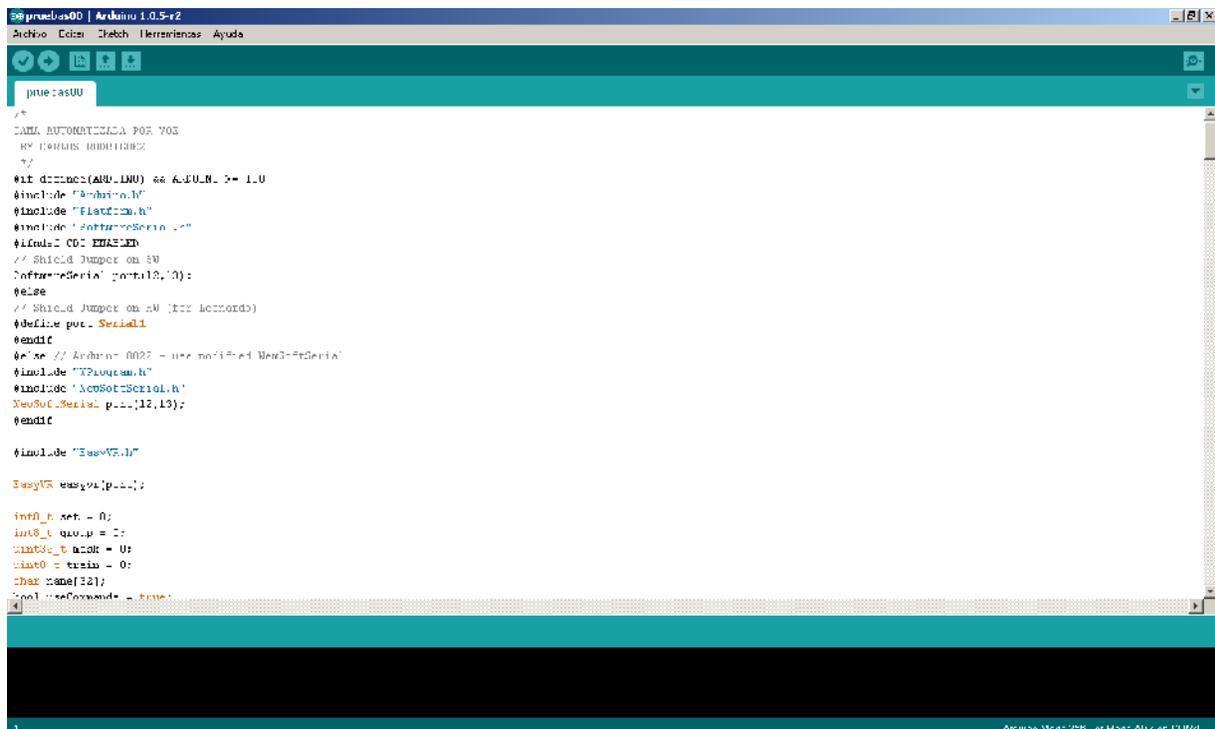


Figura 3.20 Arduino UI con el Archivo prueba00.ino

Fuente: (Autor)

### 3.3.2.2 Tabla de Sonidos

La tabla de sonidos fue implementada con los archivos generados por Balabolka que es un Sintetizador de voz gratuito que permite generar archivos de audio .wav a través del texto. Ver la Tabla 3.3 donde se muestran todos los sonidos generados y subidos al módulo EasyVR.

<b>Número de Sonido</b>	<b>Acción que realiza en la programación</b>	<b>Texto a Sonido</b>
001	Encendido	Bienvenido !! el sistema se activa con la palabra robot !! ...
002	Respuesta Comando Robot	Por favor ... diga la acción que desea realizar !!
003	Respuesta a Comando Erróneo	Comando invalido !! ... intente de nuevo por favor !!
004	Respuesta a Comando Acción	Luz
005	Respuesta a Comando Mira	Calefacción
006	Respuesta a Comando Ataca	Ventilador
007	Respuesta a comando Adelante	Adelante cabecera
008	Respuesta a Comando Atrás	Atrás cabecera
009	Respuesta a Comando Arriba	Subiendo Cama
010	Respuesta a Comando Abajo	Bajando Cama
011	Respuesta a comando sin respuesta	Por favor repita
012	Respuesta a Comando Para	Motores Apagados

Tabla 3.3 Sonidos Generados por Balabolka (TTS)  
Fuente: (Autor)

Después estos archivos fueron compilados por QuickSyntesis y posteriormente subidos al EasyVR Comander.

### 3.3.3 Implementación del sistema

La implementación de la cama automatizada, se dividió en diferentes partes; la primera fue las conexiones de los motores, de acuerdo a la Figura 3.11 se conectó las salidas de los relés, las respectivas tierra y neutro para funcionamiento y protección de los motores. El primer motor realiza un movimiento gradual de 0° a 70° en el espaldar o cabecera de la cama, el segundo motor realiza un movimiento subiendo o bajando toda la cama de 40 cm a 80 cm de altura del piso.



Figura 3.21 Conexiones de los motores

Fuente: (Autor)

Hay que notar en la Figura 3.11 cómo la señal de la fase va conectada a la entrada del contactor normalmente abierto del relé y éste sólo cerrará el circuito y pondrá a funcionar el motor o actuadores sólo si se activa el relé,

Se implementó un interruptor y dos tomacorrientes para que el sistema se haga más sencillo de conectar y acoplar cualquier ventilador o calefactor que se encuentre en el mercado. Se podría conectar cualquier otro tipo de artefactos electrónicos que no sobrepasen los 3 amperios de corriente.



Figura 3.22 Sistema implementado en la Cama

Fuente: Autor

Se realizó un análisis de la corriente consumida por el sistema para poder realizar las conexiones con un cable adecuado sin que exista ninguna sobrecarga en el conductor del circuito.

#### CÁLCULO DEL CABLE

MOTORES=3 A

Foco= 20 Vatios  $I=P/V= 0.16$  A

Ventilador=100 Vatios;  $I=P/V = 0.8 \text{ A}$

Calefactor= 300 Vatios;  $I=P/V = 2,8 \text{ A}$

La suma de las corrientes es: 6.76 A

El cable recomendado según la tabla de AWG es Cable #16 AWG y para mayor maniobrabilidad se utilizó cable flexible.



Figura 3.23 Visión General de la Cama

Fuente: (Autor)

Todas las conexiones fueron realizadas con las debidas precauciones, aislándolas para que no exista contactos a masa o cortocircuitos, a razón de que se trabajará con voltajes altos.

Se colocó canaletas y los cables que se usaron fueron aislados para que no existan cortocircuitos en el sistema asegurando el buen funcionamiento del proyecto.



Figura 3.24 Parte de elevación del espaldar.

Fuente: (Autor)

En la Figura 3.25 se observa el producto terminado, se añadió un case de plástico que cubra toda la circuitería para que no exista manipulación.



Figura 3.25 Proyecto terminado

Fuente: (Autor)

## CAPÍTULO 4

### RESULTADOS Y COSTOS

#### 4.1 Evaluación Técnica

Evaluación	Funciona		Observación
	Si	No	
Conexión de la cama hospitalaria	x		Se enciende la luz piloto del Swich General
Alimentación Arduino	x		Encendido Led Indicador
Alimentación de EasyVR	x		Encendido Led Indicador
Alimentación de Módulo 8 Relés	x		Encendido Led Indicador
Encendido de Luz de lectura	x		Encendido Foco
Encendido de Ventilador	x		
Encendido de Calefactor	x		
Encendido de Motores	x		Se mueven los Motores
Sensores de fin de carrera	x		Máximo y mínimo de los 2 motores
Reconocimiento de voz	x		Trigger del sistema
Comandos de voz	x		Prueba de todos los comandos
Arduino UI	x		Verificación Software, compilación del programa
Comunicación Serial	x		A través de HyperTerminal
Transmisión de datos entre módulos	x		Funcionamiento de los Actuadores
Funcionamiento del Sistema	x		

Tabla 4.1 Evaluación técnica  
Fuente: (Autor)

#### 4.2 Pruebas de funcionamiento

Está dirigido a usuarios que conocen del funcionamiento interno del sistema, en este proyecto se desarrollo los siguientes exámenes de cada módulo:

#### 4.2.1 Pruebas de funcionamiento del EasyVR

<b>Pruebas</b>	<b>Descripción</b>	<b>Resultados Esperados</b>	<b>Resultados Obtenidos</b>	<b>Medidas adoptadas en caso de falla de la prueba</b>
<b>Detección del sonido de la voz.</b>	Detecte el sonido de la voz del usuario.	Comprobar el comando de voz es o no es correcto.	Se ha tenido éxito en la prueba.	Repita el comando de voz.
<b>Transmisión de mando</b>	Tiene la finalidad de informar al usuario si el comando se realizó correctamente.	Reproducir un sonido de Beep cada que los comandos se realizan correctamente.	Se ha tenido éxito en la prueba.	Revisar el código.

Tabla 4.2 Tabla Pruebas EasyVR  
Fuente: (Autor)

<b>Comando</b>	<b>Tiempo</b>	<b>Porcentaje de asertividad para el reconocimiento de voz</b>
Robot	10 ms	100%
Acción	28 ms	93%
Mira	20ms	94%
Muévete	30ms	88%
Ataca	24ms	90%
Adelante	29ms	93%
Atrás	30ms	95%
Arriba	25ms	96%
Abajo	20ms	97%
Para	32ms	89%

Tabla 4.3 Tiempos de respuesta y asertividad  
Fuente: (Autor)

#### 4.2.2 Pruebas de funcionamiento de Arduino Mega

<b>Pruebas</b>	<b>Descripción</b>	<b>Resultados Esperados</b>	<b>Resultados Obtenidos</b>	<b>Medidas adoptadas en caso de falla de la prueba</b>
<b>Operación TX.</b>	Compruebe si el EasyVR está recibiendo comandos de voz.	Comprobar si se escuchan dichos comandos.	Se ha tenido éxito en la prueba.	Repita el comando de voz.
<b>Comunicación en serie.</b>	Compruebe si se está realizando la comunicación.	El TX y RX LEDs parpadearán.	Se ha tenido éxito en la prueba.	Compruebe que los cables están conectados correctamente.

Tabla 4.4 Tabla de prueba de Arduino Mega  
Fuente: (Autor)

#### 4.2.3 Pruebas de funcionamiento de EasyVR GUI 2.1.8

<b>Pruebas</b>	<b>Descripción</b>	<b>Resultados Esperados</b>	<b>Resultados Obtenidos</b>	<b>Medidas adoptadas en caso de falla de la prueba</b>
<b>Reconocimiento del sonido de la voz pre grabada.</b>	Reconocer el sonido grabado previamente.	Comprobar si se reconoce el sonido de la voz.	Se ha tenido éxito en la prueba.	Volver a intentar el comando de voz.

Tabla 4.5 Tabla de prueba de Arduino UI  
Fuente: (Autor)

#### 4.2.4 Pruebas de funcionamiento de Arduino UI

Se refiere a la correcta compilación del programa y el levantamiento y validación de comandos vía el monitor serial en Arduino UI.

<b>Pruebas</b>	<b>Descripción</b>	<b>Resultados Esperados</b>	<b>Resultados Obtenidos</b>	<b>Medidas adoptadas en caso de falla de la prueba</b>
<b>Función programada correctamente.</b>	Verifique que el comando de voz está de acuerdo con la función programada.	Verificar que los comandos de voz se escuchan de acuerdo con su función.	Se ha tenido éxito en la prueba.	Depurar el código.
<b>Transmisión RX / TX.</b>	Compruebe si la configuración de la comunicación serie es correcto.	Comprobar si la comunicación se está haciendo correctamente.	Se ha tenido éxito en la prueba.	Compruebe el código de programa.
<b>La rotación de los motores de derecha / izquierda.</b>	Compruebe si el programa es correcto.	Los motores deben girar hacia el lado correcto cuando se le solicite.	Se ha tenido éxito en la prueba.	Compruebe el código de programa.

Tabla 4.6 Tabla de prueba de Arduino UI  
Fuente: (Autor)

#### 4.2.5 Pruebas de funcionamiento de los Motores

El funcionamiento del motor a través de los comandos definidos. Si no hay órdenes definidas efectúen la acción no se puede realizar la pruebas.

<b>Pruebas</b>	<b>Descripción</b>	<b>Resultados Esperados</b>	<b>Resultados Obtenidos</b>	<b>Medidas adoptadas en caso de falla de la prueba</b>
<b>El funcionamiento del motor.</b>	Compruebe el funcionamiento del motor.	Hacer funcionar el motor.	Se ha tenido éxito en la prueba.	Revise las conexiones y programación.
<b>Girar a la izquierda / derecha del motor.</b>	Compruebe la funcionalidad del motor.	El motor debe girar a la izquierda y la derecha.	Se ha tenido éxito en la prueba.	Revise las conexiones y programación.
<b>Velocidad</b>	Compruebe que la velocidad del motor sea la establecida.	Comprobar que los motores terminen el proceso en un tiempo determinado	Se ha tenido éxito en la prueba.	Cambiar un nuevo motor.

Tabla 4.7 Pruebas Tabla de Motores  
Fuente: (Autor)

#### 4.2.6 Pruebas de funcionamiento de la Cama Automatizada por Voz

Existe el funcionamiento del modelo a través de los comandos definidos.

Si no hay órdenes definidas que efectúen la acción deseada es la prueba defectuosa.

<b>Pruebas</b>	<b>Descripción</b>	<b>Resultados Esperados</b>	<b>Resultados Obtenidos</b>	<b>Medidas adoptadas en caso de falla de la prueba</b>
<b>Funcionamiento engranajes.</b>	Compruebe la funcionalidad de los engranajes.	Realiza los movimientos sin ningún contratiempo	Se ha tenido éxito en la prueba.	Cambiar el tamaño de los engranajes.
<b>Ángulo de elevación del modelo.</b>	Compruebe que el ángulo de ascenso y descenso sea el requerido.	Se mueva los motores según las órdenes dadas.	Se ha tenido éxito en la prueba.	Compruebe estructuras del modelo.

<b>Velocidad de ascenso y descenso.</b>	Compruebe si el tiempo de ascenso y descenso es razonable para ejemplificar.	Estar con un buen tiempo de subida y caída para ilustrar el problema propuesto.	Se ha tenido éxito en la prueba.	Compruebe los engranajes del modelo.
<b>Compruebe el funcionamiento de los sensores.</b>	Compruebe si el fin de carrera de los motores están trabajando a su mínimo y máximo.	Deje de mover la cama cuando llegue el límite de tiempo.	Se ha tenido éxito en la prueba.	Cambiar el sensor y comprobar su funcionamiento .
<b>Compruebe el funcionamiento del LED.</b>	Compruebe si el LED está encendido cuando el sistema esté encendido.	LED permanece encendido mientras la cama está habilitada.	Se ha tenido éxito en la prueba.	Compruebe el funcionamiento del LED y el código.

Tabla 4.8 Tabla de pruebas simultaneas  
Fuente: (Autor)

### 4.3 Análisis de resultados

#### 4.3.1 Análisis de la Evaluación técnica.

La Tabla 4.1 muestra una lista de partes y procesos que se revisaron, para evaluar el sistema. No hubo ninguna novedad en cuanto al funcionamiento del sistema con lo cual se procedió a realizar la siguientes pruebas de funcionamiento.

#### 4.3.2 Análisis del funcionamiento del EasyVR

Para el análisis de resultados del funcionamiento del EasyVR se realizaron algunas pruebas listadas en las Tablas 4.2 y 4.3.

En la tabla 4.2 se analiza dos aspectos, el primero es la detección, en el que se comprobó los comandos si son o no son correctos, teniendo como medida en caso de falla repetir el comando de voz.

El segundo aspecto analizado es la transmisión de un comando, indicando con la reproducción de un sonido que el comando fue enviado correctamente, en caso de que no exista este sonido habría que revisar el código.

Dentro de las pruebas realizadas y listadas en la Tabla 4.3 donde se repitieron todos los comandos 100 veces para averiguar la asertividad del sistema de reconocimiento de voz, también se tomaron los tiempos de ejecución con ayuda de la comunicación serial para saber cuánto tiempo tardó el sistema en reconocer cada comando, el porcentaje de asertividad para evaluar los comandos fue alto.

#### **4.3.3 Análisis del funcionamiento de GUI EasyVR**

En la Tabla 4.5 se presentan los resultados comprobando si la funcionalidad de la interfaz EasyVR GUI realiza las tareas necesarias.

La interface también ayuda a comprobar los comandos de voz pregrabados en el firmware esta habilitados y funcionando.

#### **4.3.4 Análisis del funcionamiento Arduino Mega**

Es una conexión USB que se realiza a través de Arduino que está conectado a la PC, y esta comunicación se realiza a través de la interfaz de usuario de Arduino por su monitor de serie. Comprobando si se ha establecido una conexión.

#### **4.3.5 Análisis del funcionamiento Análisis Arduino UI**

Comprueba si la funcionalidad de la interfaz Arduino UI está realizando tareas debidas y si se compiló correctamente el programa. (Tabla 4.6)

#### **4.3.6 Análisis del funcionamiento de la Integración Arduino Mega con EasyVR**

Para comprobar si su integración es correcta se ha incrustado directamente en el código de Arduino en el modo normal. Las respuestas que se muestran en el Arduino Monitor Serial donde mostrará la correcta conexión del módulo.

Se realizaron rutinas de validación para verificar que los comandos enviados por EasyVR hacia Arduino están siendo tratados correctamente y realizar la acción previamente estipulado.

#### **4.3.7 Análisis del funcionamiento de los Motores**

Para un buen funcionamiento de los motores hay que tomar en cuenta que estos se muevan de la manera correcta y en los tiempos promedio estimados, de no ser de esta manera será necesario el cambio por un nuevo motor y en caso de que exista algún error al movimiento revisar la programación.

#### **4.3.8 Análisis del funcionamiento de la Cama Automatizada**

En la cama automatizada por voz, son varios los factores que influyen para el funcionamiento, entre estos están los engranajes de los motores, los movimientos que realiza, la velocidad de los motores y los diferentes actuadores y módulos que hacen posible que funcione el proyecto, por lo tanto se reviso mediante pruebas que todos los elementos del circuito funcionen, como se puede observar en la Tabla 4.8

### **4.4 Análisis del Proyecto Implementado**

#### **4.4.1 Características positivas**

- Mano de obra ecuatoriana calificada.
- Tecnología de Hardware y Software se maneja en Open Source.
- Elementos existentes en el mercado ecuatoriano
- Funciona con cualquier tipo de voz, y con un porcentaje de asertividad bastante bueno.

#### 4.4.1 Características negativas

- El límite de Memoria que tiene el microcontrolador y el EasyVR, restringieron y limitaron los comandos a usar las palabras simples predefinidas.
- Los comandos programados por la interface gráfica de EasyVr sólo funciona para la persona que grabaría el comando.
- El sistema no funciona cuando existe ruido excesivo.
- Distancia de uso 60[cm]

#### 4.5 Costos del proyecto

##### Parte Electrónica

	Unidades	Precio Unitario	Total
Arduino Mega	1	\$ 54,00	\$ 54,00
Módulo EasyVR	1	\$ 60,00	\$ 60,00
Shield Protoboart	1	\$ 12,00	\$ 12,00
Módulo 8 Relés	1	\$ 23,00	\$ 23,00
Octoacopladores	1	\$ 0,50	\$ 0,50
Microtransistores	2	\$ 0,10	\$ 0,10
Leds	10	\$ 0,10	\$ 1,00
Pines Machos	1	\$ 0,40	\$ 0,40
Jumper	1	\$ 0,10	\$ 0,10
Cable de Arduino	100	\$ 0,10	\$ 10,00

##### Parte Eléctrica

Cable #16 AWG	15	\$ 0,60	\$ 9,00
Taype	1	\$ 0,60	\$ 0,60
Tomacorriente e Interruptor	1	\$ 4,81	\$ 4,81
Cajetín	1	\$ 0,50	\$ 0,50
Focos	2	\$ 3,50	\$ 7,00
Canaleta	1	\$ 2,00	\$ 2,00
Amarras plásticas	100	\$ 0,02	\$ 2,00
Boquillas	2	\$ 1,77	\$ 3,54
Borneras	2	\$ 1,71	\$ 3,42
Tornillos	1	\$ 0,28	\$ 0,28

**Otros Gastos**

Mano de Obra	8 horasx40 días x 5 dólares	\$ 320,00	\$ 1.600,00
Trasporte	2	\$ 20,00	\$ 40,00
<b>Total</b>			USD \$1834,25

Tabla 4.10 Tabla de Costos del proyecto  
Fuente: (Autor)

## CAPÍTULO 5

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 CONCLUSIONES

- El proyecto realizado se centra en un entorno importante para la sociedad, como es el tema de las personas con discapacidad ya sea esta total o parcial, personas con dificultades motrices o personas de la tercera edad, que se les ayuda con el desarrollo de la Electrónica se hace para que su estancia en el hospital sea más placentera, pudiendo realizar acciones con simples comandos de voz sin necesidad de la ayuda de terceros.
- Se aprendió diferentes formas de realizar el reconocimiento de voz como lo es el uso del computador, módulos Electrónicos (EasyVR) y a través de la plataforma Android.
- Se logró reconocer comandos programados y también realizar las tareas programadas para cada comando de voz a través de módulos de hardware libre como son la placa de Arduino y el modulo EasyVR
- El shield de reconocimiento EasyVR con la placa Arduino Mega, resultaron ser una muy buena opción para el desarrollo de prototipos de aplicaciones que necesiten de reconocimiento del voz.
- Al reconocer un comando también se activaron los sonidos que se guardaron en la memoria del módulo de reconocimiento de voz para verificar que el comando se ejecutó correctamente
- Se desarrolló un sistema de control por voz que tuvo un porcentaje de asertividad muy alto, cabe destacar que el sistema funciona con cualquier tipo de voz.
- Los costos de realización de este proyecto son relativamente bajos y es aplicable a otros sistemas electrónicos.
- Los sistemas inteligentes crean nuevas posibilidades en la atención del paciente.

- Bajo el funcionamiento del proyecto final posteriormente se implemento un dispositivo Bluetooth, y se realizo una Aplicación a través de App Inventor la cual se conecta a la cama hospitalaria, realizando las mismas tareas que se puede realizar mediante los comando de voz a través de un teléfono inteligente; el cual tiene una interface muy amigable con botones y un lugar para enviar comandos ya sean escritos o se utilice el sistema de reconocimiento de voz de androide.

## 5.2 RECOMENDACIONES

- Buscar textos en inglés u otros idiomas a parte del español para que de esta manera la investigación sea mucho más provechosa.
- Si se necesita mayor rapidez para el uso del sistema se puede omitir la reproducción de la tabla de sonidos, con esto se ganaría mayor tiempo eliminando la reproducción del audio.
- Si el sistema va a ser utilizado por una sólo persona, éste puede ser personalizado entrenando el módulo para reconocer sólo su tono de voz.
- Si se necesita mayor capacidad o mayor rapidez se recomienda el uso de una memoria externa o el uso de un ordenador.
- Pronunciar los comandos a una distancia no mayor a los 60[cm] y si se lo hace, se debería hablar más fuerte.

### 5.3 BIBLIOGRAFÍA

*Diagrama Relé.* (2014). <http://www.opirata.com/arduino-modulo-rele-canales-p-7990.html>.  
*Eco Bed Parts.* (2014). <http://www.medicalpartssource.net/Beds/Echo/EchoBedParts.htm>.  
*Yo Reparo .* (2014). <http://hogar.yoreparo.com/electricidad/necesito-conectar-un-motor-monofasico-con-inversor-de-marcha-t1178773.html>.  
Arduino. (2014). *Arduino*. Recuperado el MAYO de 2014, de <http://www.arduino.cc/>  
Arduino. (2014). *Arduino*. Recuperado el 30 de Mayo de 2014, de <http://arduino.cc/en/Tutorial/LiquidCrystal>  
Jurafsky, Daniel, Martin y James H. (2009). *Speech and language Processing*. Pearson Hall.  
S.L, I. d. (2012). *Guía EasyVR*. <http://es.scribd.com/doc/213589342/EasyVR-Guia-Rapida> .  
VEEAR. (2014). *EasyVR*. <http://www.veear.eu/downloads>.

Borrello, Charles, Rabnovich, Chris y Jabrucki, Andrew. 2011. *RIT "Tigerbot" Humanoid Platform for Future Expansion*. Kate Gleason College of Engineering, Rochester Institute of Technology. Rochester, New York : s.n., 2011.  
<http://edge.rit.edu/edge/P12201/public/Home>.

Burgess, Phil. 2011. chipKIT Uno32: first impressions and benchmarks. *Hack a Day*. [En línea]  
27 de Mayo de 2011. [Citado el: 28 de Junio de 2014.]  
<http://hackaday.com/2011/05/27/chipkit-uno32-first-impressions-and-benchmarks/>.

Englund, Christine. 11th March 2004. *Speech recognition in the JAS 39 Gripen aircraft - adaptation to speech at different G-loads*. Department of Speech, Music and Hearing, Royal Institute of Technology (Suecia). 11th March 2004. Master Thesis in Speech Technology.

Jiang, Leo. 2014. *Project Proposal for Multifunction Intelligent Headphone System*. Sound Tech Inc. Burnaby : s.n., 2014.

Jurafsky, Daniel y Martin, James H. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. s.l. : Pearson Prentice Hall, 2009. 0131873210 .

Kim, Ryan. 2011. Nuance buys Vlingo. *GigaOm*. [En línea] 20 de Diciembre de 2011. [Citado el:  
27 de Julio de 2014.] <http://gigaom.com/2011/12/20/nuance-buys-vingo-builds-a-voice-technology-giant/>.

Poza Luján, José Luis y Posadas Yagüe, Juan Luis. 2009. *Revisión de las arquitecturas de control distribuido*. 2009.

Salvador, Marc Franco. 2011. *Navegación web usando la voz*. Escuela de Informática, Universidad politécnica de Valencia. 2011. Proyecto Final de Carrera.

*Samsung unveils new Android phone with "S Voice" personal assistant*. William Meisel. 2014. Junio de 2014, Speech Strategy News. ISSN 1932-8214.

Stan Augarten. 1998 - 2009. The Most Widely Used Computer on a Chip. *The National Museum of American History*. [En línea] Smithsonian Institution, 1998 - 2009. [Citado el: 14 de Julio de 2014.] <http://smithsonianchips.si.edu/augarten/p38.htm>. ISBN 0-89919-195-9.

Stevens, Richard. 2005 . *Advanced Programming in the UNIX Environment*. 2005 . ISBN 0201563177.

Toshiba. 64-Bit MIPS-Based Microcontroller With PCI Interface. *Toshiba*. [En línea] [Citado el: 14 de Julio de 2014.] <http://uk.computers.toshiba-europe.com/innovation/jsp/news.do?service=UK&year=NONE&ID=00000005a4>.

XMOS. Development Kits. *XMOS*. [En línea] [Citado el: 15 de Julio de 2014.] <http://www.xmos.com/products/development-kits/xc-3-led-tile-kit>.

# ANEXOS



# **ANEXO 1**

## **Programación**

# **ANEXO 2**

Manual del Módulo

EasyVR

# **ANEXO 3**

## **Manual de Usuario**

# **ANEXO 4**

## **Análisis Viper**