UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA: INGENIERÍA ELECTRÓNICA

Tesis previa a la obtención del título de: INGENIERO ELECTRÓNICO

TEMA:

IMPLEMENTACIÓN DEL MODELO DE UNA HABITACIÓN CON UN SISTEMA DE ASISTENCIA PARA DISCAPACITADOS, MEDIANTE EL USO DE UN SENSOR DE INTERFAZ NATURAL DE USUARIO KINECT DE MICROSOFT.

AUTORES:

LUIS FABIÁN QUISHPE TUTILLO

DIEGO ESTEBAN ULLOA JÁCOME

DIRECTOR:

VÍCTOR VINICIO TAPIA CALVOPIÑA

Quito, julio del 2014

DECLARACIÓN DE RESPONSABILIDAD Y AUTORIZACIÓN DEL USO DE TRABAJO DE TITULACIÓN

Nosotros, autorizamos a la Universidad Politécnica Salesiana, la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaramos que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Quito, julio de 2014.

Luis Fabián Quishpe Tutillo	Diego Esteban Ulloa Jácome
CI: 1719298208	CI: 1723687313

DEDICATORIA

Dedico este proyecto a mis padres Enrique e Isabel, por ser un ejemplo de vida y apoyarme incondicionalmente en mi sueño de ser ingeniero, porque ustedes siempre me brindaron confianza, aliento, apoyo, amor, consejos y comprensión para poder culminar el presente trabajo de grado. También dedico esta labor a mi hermano Fernando, por ser un puntal fundamental en la culminación de este trabajo y por último y no menos importante agradezco a mi novia Marisol Chimarro, por ser un pilar muy importante en este proyecto y en el proceso universitario y a la vez brindarme su paciencia, comprensión, bondad y sacrificio lo cual me inspiro a ser cada día mejor. A mi compañero Diego Ulloa, le deseo éxitos en la vida.

Luis Fabián Quishpe Tutillo

Dedico este trabajo a mis padres, Fernando y Amparito, por el apoyo, comprensión y paciencia entregada durante el desarrollo del mismo, a mis hermanos Xavier y Álvaro, por brindarme sus ideas, ayudarme a salir de bloqueos y presionarme para terminar mis estudios lo más pronto posible, a mis amigos, familiares y compañeros por sus palabras de aliento y buenos deseos, y finalmente a Fabián, mi compañero de trabajo de titulación, por su arduo trabajo durante todo este tiempo, sin el cual en este proyecto no hubiese avanzado.

Diego Esteban Ulloa Jácome

AGRADECIMIENTO

Agradecemos a la Universidad Politécnica Salesiana por habernos acogido durante nuestros estudios, así también a todos los profesores que estuvieron a nuestro lado en el transcurso de este tiempo.

A nuestro tutor el Ing. Víctor Vinicio Tapia, por compartir sus conocimientos que fueron de gran ayuda para el desarrollo de este proyecto.

Diego Esteban Ulloa Jácome

&

Luis Fabián Quishpe Tutillo

ÍNDICE

INTRODUCCIÓN **1**

CAPÍTULO 1 ..... **2**

PLANTEAMIENTO DEL PROBLEMA **2**

1.1 Planteamiento del problema **2**

1.2 Hipótesis **2**

1.3 Objetivos **3**

1.3.1 Objetivo general. 3

1.3.2 Objetivos específicos. 3

1.4 Justificación **3**

CAPÍTULO 2 ..... **5**

ESTADO DEL ARTE Y DEFINICIONES **5**

2.1 Estado del arte **5**

2.1.1 Kinect. 5

2.1.1.1 Efecto Kinect. 5

2.1.1.2 Utilización del Kinect en el mundo. 8

2.1.1.3 Herramientas desarrolladoras. 10

2.1.1.4 Aplicaciones relacionadas con el uso del sensor Kinect. 11

2.1.2 Sistemas para discapacitados. 11

2.1.2.1 Funciones y tipos. 11

2.1.2.2 Discapacidad en el Ecuador. 13

2.2 Discapacidad **17**

2.2.1 Definición de discapacidad. 17

2.2.2 Tipos de discapacidad. 17

2.2.2.1 Discapacidad física. 17

2.2.2.2 Discapacidad sensorial. 18

2.2.2.3 Discapacidad psíquica. 18

2.2.2.4 Discapacidad intelectual o mental. 18

2.3 Hardware **18**

2.3.1 Sensor Kinect. 18

2.3.1.1 Historia. ... 18
2.3.1.2 Características. ... 19
2.3.1.3 Funcionamiento. ... 21
2.3.2 Ordenador. ... 23
2.3.2.1 Importancia. ... 23
2.3.2.2 Características de compatibilidad del ordenador con Kinect. ... 24
2.3.3 Arduino. ... 24
2.3.3.1 Historia. ... 24
2.3.3.2 Arduino Mega 2560. ... 24
2.3.3.2.1 Ventajas. ... 25
2.3.3.2.2 Características. ... 25
2.4 Software ... 26
2.4.1 Visual Studio C# 2010 Express. ... 26
2.4.1.1 Introducción. ... 26
2.4.1.2 Aplicación WPF C#. ... 26
2.4.2 Kinect SDK v1.7. ... 27
2.4.3 Speech Recognition. ... 27
2.4.3.1 Requisitos de utilización. ... 27
CAPÍTULO 3 ... 29
CARACTERÍSITICAS DEL PROYECTO ... 29
3.1 Características específicas empleadas en el proyecto ... 29
3.1.1 Reconocimiento de voz. ... 29
3.1.2 Reconocimiento de imágenes. ... 30
3.1.3 Sensor de movimiento. ... 31
3.1.4 Comunicación serial. ... 32
3.2 Diseño del proyecto ... 33
3.2.1 Análisis de los formularios de C#. ... 33
3.2.1.1 Formulario portada. ... 34
3.2.1.2 Formulario window3. ... 34
3.2.1.3 Anexo SpeachGramar. ... 36
3.2.2 Diagramas de flujo. ... 37
3.2.2.1 Programa en C#. ... 37

3.2.2.1.1 Estructura Principal. 37
3.2.2.1.2 Skeleton Tracking. 41
3.2.2.1.3 Reconocimiento de voz. 43
3.2.2.2 Código Arduino. 46
3.3 Diseño de placas PCB **48**
3.4 Caja de reducción **51**
3.5 Guía de instalación del software **52**
CAPÍTULO 4 ..... **55**
ANÁLISIS DE RESULTADOS **55**
4.1 Diagrama de bloques **55**
4.1.1 Fuente de alimentación (AC/DC). 56
4.1.2 Sensor Kinect. 56
4.1.3 Ordenador. 57
4.1.4 Tarjeta de control. 58
4.1.5 Sistema de asistencia. 58
4.1.5.1 Sistema de cama. 59
4.1.5.2 Sistema de intensidad de la luz. 61
4.1.5.3 Sistema de ventana. 62
4.1.5.4 Sistema de persiana. 63
4.1.5.5 Sistema de puerta. 64
4.1.6 Condiciones ideales. 65
4.1.7 Alcance. 66
4.1.8 Guía de usuario. 67
4.1.8.1 Conexión de equipos. 67
4.1.8.2 Menú de inicio. 70
4.1.8.3 Ventana de comandos. 70
CONCLUSIONES **73**
RECOMENDACIONES **75**
LISTA DE REFERENCIAS **76**

ÍNDICE DE FIGURAS

Figura 1. Demostración de Project Natal 5
Figura 2. II encuentro de inspiración y networking. 7
Figura 3. Kinect versión 2 8
Figura 4. Clase utilizando Kinect 9
Figura 5. Silla de ruedas motorizada 12
Figura 6. Head Mouse 13
Figura 7. Datos del INEC 14
Figura 8. Gráfica de porcentaje de discapacidad 16
Figura 9. Discapacidad en el Ecuador 17
Figura 10. Estructura del sensor 19
Figura 11. Partes del sensor Kinect 20
Figura 12. Ubicación del sensor 21
Figura 13. Recepción de escala de grises 22
Figura 14. Esqueleto de orientación 23
Figura 15. Arduino Mega 2560 25
Figura 16. Símbolo C# 26
Figura 17. Reconocimiento de voz 29
Figura 18. El código para comparar el audio del sensor 30
Figura 19. Reconocimiento del sensor Kinect conectado 30
Figura 20. El sensor que se va utilizar 31
Figura 21. Configuración de la cámara RGB 31
Figura 22. Configuración de los frames de la cámara 31
Figura 23. Configuración del Skeleton de Kinect 32
Figura 24. Matriz de las coordenadas 32
Figura 25. Coordenadas de los puntos para su evaluación 32
Figura 26. Comunicación serial 33
Figura 27. Matriz de transmisión por el puerto serial 33
Figura 28. Portada de la aplicación 34
Figura 29. Panel de control del sistema 36
Figura 30. Diagrama de flujo del código de C#. 37
Figura 30. Diagrama de flujo del código de C# (continuación). 38
Figura 30. Diagrama de flujo del código de C# (continuación). 39

Figura 31. Diagrama de flujo de adquisición de datos del Skeleton Tracking. 41
Figura 32. Diagrama de flujo del empleo del reconocimiento de voz en C#. 43
Figura 32. Diagrama de flujo del empleo del reconocimiento de voz en C# (continuación). 44
Figura 33. Diagrama de flujo del código del programa en Arduino. 46
Figura 33. Diagrama de flujo del código del programa en Arduino (continuación). 47
Figura 34. Diseño del sistema de luces 48
Figura 35. PCB sistema de iluminación 49
Figura 36. PCB sistema de tarjeta Arduino 49
Figura 37. Circuito de control de cama 50
Figura 38. PCB sistema de control de cama 50
Figura 39. Caja reductora del sistema de ventana 51
Figura 40. Versión del sistema operativo 52
Figura 41. Referencia de Kinect en C# 53
Figura 42. Referencia de Speech 54
Figura 43. Diagrama de bloques del sistema de asistencia. 55
Figura 44. Esquema de conexión 57
Figura 45. Tarjetas PCB desarrolladas para el proyecto. 59
Figura 46. Tarjeta de control del sistema de la cama 60
Figura 47. Módulo del control de la cama 60
Figura 48. Tarjeta de control del sistema de iluminación. 61
Figura 49. Módulo del control de iluminación 62
Figura 50. Módulo del control de ventana 63
Figura 51. Módulo del control de persiana 64
Figura 52. Módulo del control de puerta 65
Figura 53. Estructura de acoplamiento con la tarjeta Arduino 68
Figura 54. Esquema de conexión del sistema de asistencia para discapacitados 69
Figura 55. Portada de la aplicación 70
Figura 56. Panel de control del sistema 71

ÍNDICE DE TABLAS

Tabla 1. Desarrollo de los SDK. 10
Tabla 2. Estadísticas de discapacidad. 15
Tabla 3. Colores de los indicadores. 35
Tabla 4. Esquema de distribución de puertos en la tarjeta Arduino..... 68
Tabla 5. Lista jerárquica de instrucciones de voz. 72

ÍNDICE DE ANEXOS

Anexo 1. Servomotor HK 15338. 80
Anexo 2 . Servomotor HS-805BB. 81
Anexo 3. Servomotor HS755-HB. 82
Anexo 4. IGBT IRG4PC50WPbF. 83

RESUMEN

El presente proyecto comprende un sistema de asistencia para discapacitados, el cual brinde el control del entorno de una habitación para una persona, cuya discapacidad comprenda la incapacidad, ya sea temporal o de manera permanente, de sus extremidades inferiores. Esto debido a que para la correcta operación del sistema es necesario que el usuario sea capaz de mover la mano derecha y brindar órdenes de manera oral. El sistema comprende diferentes etapas las cuales son: el control de iluminación de la habitación, el cual se hará de manera gradual semejante a la operación de un dimmer; el control de la cama, en este caso es una cama automática semejante a las utilizadas en clínicas y hospitales, de la cual se controlan la elevación de dicha cama y de la inclinación su espaldar; el control de la apertura de la puerta de la habitación, la cual es batiente; el control de la apertura de la ventana, siendo esta corrediza y finalmente el control de la apertura de las persianas de dicha ventana. Para todo este sistema se ha empleado como intérprete del usuario el mismo Kinect que se utiliza en las consolas de video juegos, del cual se capta el movimiento del cuerpo humano, a la vez que la voz y para utilizar como órdenes para el sistema, en este caso se ha utilizado de lado del sensor el lenguaje de programación C# de Microsoft, y de lado de los actuadores el lenguaje de las tarjetas Arduino.

ABSTRACT

This project includes a system of assistance for disabled, which provides control of the environment of a room, the disability could be either temporarily or permanently of his lower extremities. For the correct operation of the system is necessary that the user can move his right arm and speak. The system includes of different stages: the lighting control of a room, bed control, the door opening control, the window's opening control and the blinds. The lighting control works like a dimmer, the automatic control of the bed uses the same kind of beds that are used in clinics and hospitals, this system controls the elevation of the bed and the its back, the door opening control uses a hinged door, the window's opening control uses a sliding windows, and the blind's opening control uses a zebra blind type. The system uses the Kinect sensor, this sensor is the same that used in video game consoles, which captures the movement of the human body and the voice and use them as commands to the system. In the project has been used C # programming language of Microsoft to the program of the sensor and C programming language own Arduino for the actuators.

INTRODUCCIÓN

Las discapacidades físicas generan muchas dificultades para las personas que las padecen, las mismas que impiden un desenvolvimiento normal, limitados a través del tiempo de oportunidades laborales, acceso a espacios públicos, inclusive en la misma sociedad, para evitar esta problemática, el ser humano ha buscado formas de implementar maneras de ayudar y permitir a las personas discapacitadas, equiparar las oportunidades y capacidades de los demás miembros de la sociedad.

Con el avance de la tecnología el ser humano ha llegado, de cierto modo, a facilitar su estilo de vida, inclusive aunque no es común el llegar a automatizar su entorno, es por ello que viendo las bondades de la tecnología y limitantes de individuos discapacitados, se ha promovido el desarrollo de proyectos que permitan explotar las potencialidades, que un individuo discapacitado posee, para que con ellas supere las adversidades de su diario vivir.

Existen diferentes tipos y grados de discapacidad física, valorada en función de su porcentaje motriz y tiempo de acción. En el caso de las deficiencias motoras, se puede hacer uso de la domótica para que el individuo sea capaz de llevar su diario vivir de una manera confortable y apropiada, como medio de estudio para el presente proyecto se ha pensado en un sistema que brinde a las personas el control del entorno de la habitación que ocupen, considerando que los individuos posean discapacidades físicas motoras, ya sea temporal o permanente, ellas sean incapaces de mover sus extremidades, lo que los dejaría postrados o imposibilitados de realizar actividades que requieran el uso de sus piernas.

Observando este inconveniente se ha pensado realizar un sistema domótico, con un sensor de interfaz humana que registre su voz y el movimiento de las extremidades superiores del individuo para controlar su entorno, razón sea la que, entre diferentes sensores, se ha escogido el sensor Kinect por sus bondades en el reconocimiento de variables generadas por el cuerpo humano.

CAPÍTULO 1

PLANTEAMIENTO DEL PROBLEMA

1.1 Planteamiento del problema

A través de la información obtenida de las instituciones gubernamentales encargadas de llevar registros actuales de la información de la población como: el INEC y el CONADIS, se puede evidenciar que en el Ecuador presenta índices considerables de discapacidad, pero de este grupo de personas, aproximadamente la mitad de ella posee discapacidad física.

Esta población enfrenta muchas dificultades en su vida diaria, dependiendo del grado de discapacidad puede llegar a no ser autosuficiente para realizar actividades comunes, por las que pueden llegar incluso a encontrarse limitados a estar postrados en una cama.

Desde la aprobación de la última constitución vigente de la república en el 2008, se ha garantizado la inclusión social y laboral de personas con discapacidad, por tal motivo instituciones públicas y privadas los incluyen en su personal laboral, y algunas inclusive desarrollan proyectos que van en el mejoramiento de la calidad de vida para este grupo de personas, como es el caso de la Universidad Politécnica Salesiana.

1.2 Hipótesis

La implementación de un sistema de asistencia para discapacitados, pretende brindar el control de una habitación mediante un sistema domótico, para personas con deficiencia motora. El cual permite controlar con la voz y gestos del usuario su entorno con las siguientes funciones: el control de intensidad de luces, apertura y cierre de puertas, ventanas, persianas de la habitación y el grado de inclinación de la cama.

1.3 Objetivos

1.3.1 Objetivo general.

Implementar un modelo de una habitación con un sistema de asistencia para discapacitados, mediante el uso de un sensor de interfaz natural de usuario, que registre el movimiento y voz del cuerpo humano a través del sensor Kinect de Microsoft.

1.3.2 Objetivos específicos.

- Analizar el lenguaje de programación del SDK del sensor Kinect basado en C#.
- Desarrollar la aplicación que permita comunicar el movimiento y voz del cuerpo humano, como órdenes de control para un sistema de asistencia para discapacitados.
- Implementar el sistema de control que permita comunicar el computador con la solución electrónica de un sistema de asistencia para discapacitados a través del puerto USB.
- Verificar el correcto funcionamiento, tanto de hardware como de software del sistema de asistencia para discapacitados.

1.4 Justificación

Gracias al gran avance y desarrollo de la tecnología, de todo lo concerniente al reconocimiento de gestos, expresiones, voz, y movimiento del cuerpo humano, existen hoy en día dispositivos que facilitan en cierto grado la realización de tareas cotidianas, la seguridad de los ordenadores, comunicaciones, aplicaciones de teléfonos celulares, consolas de videojuegos, entre otras. Dentro de este campo, los que brindan mayor interactividad hacia el usuario son los sensores de interfaz natural de las consolas de videojuegos como: Kinect de Microsoft, PlayStation Move, Wiimote de Nintendo entre otras soluciones para ordenadores que sirven de interfaz de usuario para videojuegos.

Este proyecto propone el desarrollo de un sistema de asistencia para discapacitados, empleando para ello el sensor de interfaz natural de usuario Kinect del Xbox 360. En

este caso las diversas funcionalidades que nos brinda el sensor Kinect permite el reconocimiento de voz y el movimiento del cuerpo humano.

El desarrollo de esta solución permitirá, proponer soluciones a problemas cotidianos que se presentan en personas discapacitadas con dificultad de movilidad en sus extremidades inferiores, permitiéndole a su operador controlar desde su entorno intensidad de luces, apertura y cierre de puertas, ventanas, persianas de la habitación y el grado de inclinación de la cama.

CAPÍTULO 2
ESTADO DEL ARTE Y DEFINICIONES

2.1 Estado del arte

2.1.1 Kinect.

2.1.1.1 Efecto Kinect.

Kinect es una tecnología relativamente nueva, y hoy en día, el mundo la está empleando gracias todas las bondades que ofrece en el campo de los videojuegos, para brindar al usuario una mayor experiencia en lo concerniente a interacción, al utilizar el cuerpo humano como si fuera un mando de consola.

Kinect fue conocido originalmente bajo el nombre de Project Natal y anunciado por primera vez, el 1 de junio de 2009 en la "Electrnic Entertainment Expo 2009", creado por Alex Kipman, desarrollado por Microsoft para la videoconsola Xbox 360. El dispositivo es un controlador de juego libre y entretenimiento, el cual en Junio del 2011 mediante la plataforma de Windows 7 permite conectarse a un PC.

Figura 1. Demostración de Project Natal

Fuente: (Lanacion.com, 2010)

Kinect permite a los usuarios interactuar y controlar el juego, sin la necesidad del contacto físico, mediante una interfaz natural de usuario que permite reconocer:

gestos, comandos de voz, objetos e imágenes. El dispositivo tenía como objetivo principal aumentar las ventas del Xbox 360, ya que compite en el mercado de los video juegos con los sistemas Wiimote, PlayStation Move, por consiguiente, para cumplir dicho objetivo Kinect fue lanzado en Norteamérica el 4 de noviembre de 2010 y en Europa en Noviembre 10. (Microsoft, 2013)

El sensor de Microsoft ha innovado el mundo de interacción de los videojuegos, haciendo atractivo para una persona formar parte activa del juego, por ello, existen muchos adeptos para este tipo de juegos interactivos, dejando a un lado lo estático de un control de consola, los cuales no permitían tener una actividad física al jugar volviéndose un juego rutinario.

Sin embargo, la gente empezó a utilizar al sensor Kinect para usos completamente ajenos a su propósito original, desde ayudar a niños con autismo, reconstrucción de imágenes en 3D, como herramienta de apoyo para cirujanos en quirófano, entre otras aplicaciones, de este modo Kinect va más allá de los juegos y eso es lo que se denomina efecto Kinect.

El efecto Kinect es un fenómeno mundial al que se ha sumado, Ecuador, desarrollando diferentes aplicaciones de ayuda caseras e innovadoras para el desarrollo de la comodidad dela ciudadanía en general.

Observando el avance de diferentes aplicaciones el 22 de noviembre del 2012, Microsoft organizó en Madrid el "II Encuentro de Inspiración y Networking", en tal evento, propios y extraños se sorprendieron por la evolución de los diferentes proyectos, utilizando el sensor Kinect como una herramienta fundamental en las aplicaciones en diferentes áreas como salud, educación, comercio, arte, inteligencia artificial y turismo. (Pastor, 2012)

> Microsoft se quedó sorprendido de la evolución que ha logrado emitiendo las siguientes palabras: Nos sentimos emocionados de formar parte de este gran proyecto y de fomentar las relaciones profesionales entre instituciones, escuelas, empresas y personas que han encontrado en Kinect una nueva forma para dar rienda suelta a su creatividad y satisfacer necesidades reales de la sociedad actual a través de la tecnología (Pastor, 2012).

Figura 2. II encuentro de inspiración y networking.

Fuente: (Pastor, 2012)

Las prestaciones del sensor Kinect hizo que sea muy apetecido en el mercado mundial por los video juegos y la tecnología abierta para seguir evolucionando en diferentes ámbitos, por lo tanto se obtuvo como resultado la venta de más de 10 millones de unidades del dispositivo, rompiendo un record de ventas para Microsoft.

La parte fundamental del éxito del Kinect es el software creado por Microsoft para la interpretación de la información obtenida por los diferentes sensores, cámara RGB, cámara de infrarrojos y micrófonos, aquí es donde Microsoft Research ha desempeñado un papel fundamental, e incluso hoy en día, investigando los más diversos usos para Kinect en colaboración con el SDK, ingresando muy fuerte en la evolución de la tecnología y la ayuda en diferentes ámbitos sociales. (Microsoft, 2013)

El sensor ha evolucionado a la par de las peticiones de las necesidades de los usuarios en consecuencia se creó el sensor Kinect versión 2, la diferencia con su predecesor es la nueva cámara principal, la cual captura movimientos e incorpora una cámara time-of-flight (TOF) de alta resolución, lo que permite obtener mayor precisión y resolución, el nuevo método de profundidad proporcionado por la cámara TOF permite tener 3 veces más de fidelidad, además de eso permite un 60% de campo de visión más grande, que con el Kinect versión 1, lo cual permite tener un

espacio mayor de visión y por ende, tener un mayor número de personas registradas (hasta 6 personas) por el sensor, de las cuales son reconocidos los movimientos de cada uno de ellos, obteniendo un avance importante respecto al sensor Kinect versión 1 que solamente registraba las señas de dos personas.

Figura 3. Kinect versión 2

Fuente: (Abrego, 2013)

Una de las ventajas dela segunda versión del dispositivo, es también el nuevo sensor de infrarrojo, el cual permite reconocer a los usuarios y objetos en condiciones de muy poca iluminación en las cuales para el ojo humano no sea visible.

Kinect es una muestra del potencial científico y desarrollo técnico que brinda Microsoft a través de su equipo de investigación denominado Microsoft Research (laboratorio de ideas, es un capital fundamental para el futuro de Microsoft), el cual ha sido parte activa del desarrollo del sensor versión 2.

2.1.1.2 Utilización del Kinect en el mundo.

El uso del sensor de Kinect en el mundo es variado entre personas de diferentes edades, los cuales lo utilizan en diversos sectores como del entretenimiento, medicina, educación, tecnológico entre otros. (Knight, 2012)

Diferentes personas fascinadas con las bondades que ofrece el sensor Kinect, desarrollan aplicaciones de diversos tipos dentro de un entorno de Windows, en lugar de crearlas para su uso en la consola de Xbox 360, permitiendo así, obtener las señales del sensor Kinect y utilizar las mismas en la elaboración de proyectos interactivos con el usuario.

En el mundo de la medicina el sensor Kinect es utilizado para la maniobrabilidad de dispositivos médicos, que permiten al galeno el diagnóstico y monitoreo de diversas enfermedades o dolencias. Dentro de este campo se han desarrollado diversas aplicaciones como por ejemplo: en el Reino Unido se utiliza para el tratamiento de pacientes a que se recuperen después de un accidente cerebro vascular, también el sensor Kinect es utilizado en el mundo de la fisioterapia, mediante un programa completo de ejercicios y el paciente pueda recuperarse relativamente más pronto, se utiliza también en tratamiento de niños con autismo. Las ventajas de Kinect son muy grandes por ello se lo ha implementado en los quirófanos para ayudar al doctor a navegar el historial del paciente intervenido y entre otros usos más en el mundo de la medicina. (Delgado, 2011)

En el entretenimiento, el sensor Kinect es utilizado para su propósito inicial de juego, para realizar la actividad de jugar más interactiva y divertida para el usuario. En la educación el sensor Kinect ha ocupado también, un lugar para realizar las clases más iterativas para los alumnos.

Figura 4. Clase utilizando Kinect

Fuente: (Gutierrez, 2012)

El sensor se emplea también en la elaboración de nuevas tecnologías no invasivas en el caso de medicina, además se utiliza el sensor en los ámbitos de la mecánica, electrónica y muchos ámbitos que necesitan leer los movimientos del usuario.

2.1.1.3 Herramientas desarrolladoras.

Observando las bondades que ofrece el sensor Kinect, diferentes programadores investigaron sobre la forma de utilizar el sensor que empleaban en su consola para el desarrollo de proyectos propios, de este modo un desarrollador español llamado Héctor Martini, logró elaborar un código fuente que hizo de dominio público de cómo utilizar el sensor Kinect sin la consola de Xbox, para controlarlo desde el ordenador. A este tipo de uso que se le dio al sensor, el fabricante, Microsoft, no lo vio de buena manera en un inicio, pero luego fue la empresa quien libero los controladores de Kinect para que los usuarios puedan desarrollar aplicaciones para usos alternativos, en los lenguajes de programación: Visual Basic, C++ y C#, (Microsoft Developer Network, 2013).Además de permitir el desarrollo libre para el dispositivo, Microsoft, creo páginas web y blogs para recopilar experiencias y sugerencias respecto al desarrollo y usos que se pueden dar mediante el sensor. (Manely, 2013)

A través de la recopilación de experiencias de los usuarios, Microsoft ha ido evolucionando los controladores para brindar los mismos una experiencia más sencilla y práctica. Los controladores que ha tenido el sensor son los siguientes:

Tabla 1. Desarrollo de los SDK.

SDK	Empresa
SDK BETA 1	Independiente
SDK BETA 2	Independiente
SDK KINECT 1.0	Microsoft
SDK KINECT 1.5	Microsoft
SDK KINECT 1.7	Microsoft
SDK KINECT 1.8	Microsoft

Elaborado por: Luis Quishpe & Diego Ulloa

2.1.1.4 Aplicaciones relacionadas con el uso del sensor Kinect.

Las aplicaciones del sensor Kinect son diversas a favor de la salud, ciencia, educación y comodidad de los seres humanos, las mismas son de gran ayuda para el desarrollo de la humanidad. Las aplicaciones que más resaltan son las siguientes:

Aplicaciones en la medicina: en diferentes ámbitos como en la rehabilitación los pacientes mediante un programa de ejercicios en los que el realizan rutinas para la recuperación de una fisura ósea mediante el cumplimiento de ciertos parámetros, que son evaluados. También existe la ayuda en quirófanos la cual se realiza mediante la posibilidad de leer el movimiento del cuerpo y voz del cirujano, para pasar las fotografías del paciente de acuerdo al problema que presenta, ya que esto ayuda para la elaboración de la cirugía y optimiza el tiempo.

El sensor Kinect es muy popular que en la mayoría de los países del mundo, porque trabajan en la evolución de novedosas aplicaciones para el usuario como: mover el cursor del ordenador con las manos; cambiar de pista de música y ciertas tareas básicas de un computador. (xataka windows, 2013)

2.1.2 Sistemas para discapacitados.

2.1.2.1 Funciones y tipos.

Los sistemas de asistencia para discapacitados son diversos, los cuales pueden mejorar la calidad de vida de personas que tiene alguna deficiencia: visual, de movilidad, mental, auditiva, etc. Proporcionando un estilo de vida independiente en el entorno. Los cuales pueden ser: sillas de ruedas, prótesis, ayuda para la movilidad, dispositivo de ayuda visual y equipos informáticos especializados con diferentes tratamientos dependiendo del caso aumentando la capacitad de comunicación.

Con la ayuda de estas tecnologías las personas con discapacidad pueden mejorar sus habilidades y pueden participar en la sociedad de forma más activa.

Algunas de las tecnologías implementadas hasta hoy son:

- Orugas: equipos que suben y bajan escaleras para poder independizar a una persona con baja movilidad.

- Scooters eléctricos: facilita que la persona con poca movilidad se pueda desplazar en un entorno de distancias cortas y planas, sin ningún inconveniente.
- Silla de ruedas motorizadas: es una silla de ruedas con control de maniobrabilidad, en consecuencia facilita la movilidad con menos esfuerzo, a la persona que tiene deficiencia de movilidad.

Figura 5. Silla de ruedas motorizada

Fuente: (Mundo Elevación, 2014)

- Siscodis: un software que facilita la comunicación con personas con deficiencia de comunicación.
- HeadMouse: el control de la computadora es por webcam, por lo tanto se realiza el uso del ordenador sin la necesidad de las manos, ya que se controla el mouse por ligeros y movimientos de su cabeza.

Figura 6. Head Mouse

Fuente: (Discapacidad online, 2011)

- Teclado a voz: es una aplicación que pronuncia el nombre de las teclas cuando se pulse una de ellas, emite un sonido correspondiente a cada tecla, ayudando a una persona con discapacidad visual.

2.1.2.2 Discapacidad en el Ecuador.

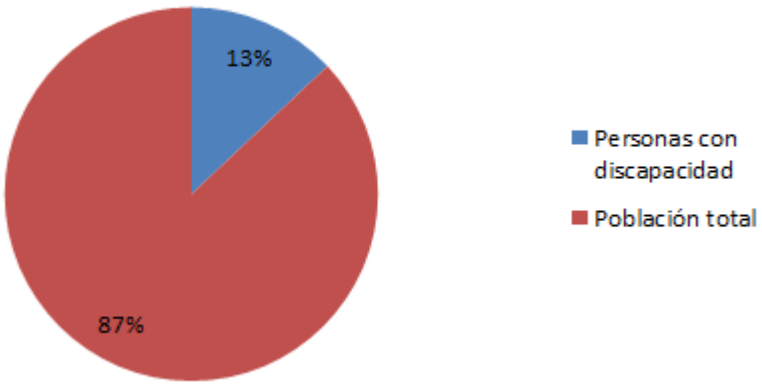
La discapacidad en el Ecuador, es un problema nacional, lo cual se puede evidenciar por los datos presentados en el censo del 2011, el cual indica que el 13% de la población nacional es decir 1.608.334, posee algún grado de discapacidad referenciando a un total 14'483.499 habitantes.

Los grupos considerados discapacitados en el Ecuador, no cuentan con el apoyo necesario, carecen de facilidades en el desempeño de su vida diaria y diversas funciones. El programa más popular y que mayor ayuda brinda desde una entidad gubernamental es el programa Misión Solidaria Manuela Espejo, fundado por el ex Vicepresidente de la República Lenin Moreno.

La Constitución de la República del Ecuador aprobada en el 2008 también realiza reformas en las cuales menciona 21 artículos y una disposición transitoria en defensa de los derechos de los discapacitados y la responsabilidad del estado en su implementación, en la cual dispone la inclusión en el ámbito laboral de las personas con discapacidad, pero desafortunadamente en el 2014 no todas las personas están incluidas en el ámbito social, ni se encuentran registradas en el CONADIS teniendo

un gran sector de esta población olvidada del total de personas con discapacidad, los cuales agrupan cerca del 13% de la población nacional según datos de SENPLADES.

Figura 7. Datos del INEC



Fuente: (INEC, 2013)

Las discapacidad en el Ecuador se clasifican en varios grupos que son: auditiva, física, intelectual, lenguaje, psicológico y visual, de acuerdo al CONADIS la distribución de las personas con discapacidad en el Registro Nacional de Discapacidades (Conadis, 2013).

Tabla 2. Estadísticas de discapacidad.

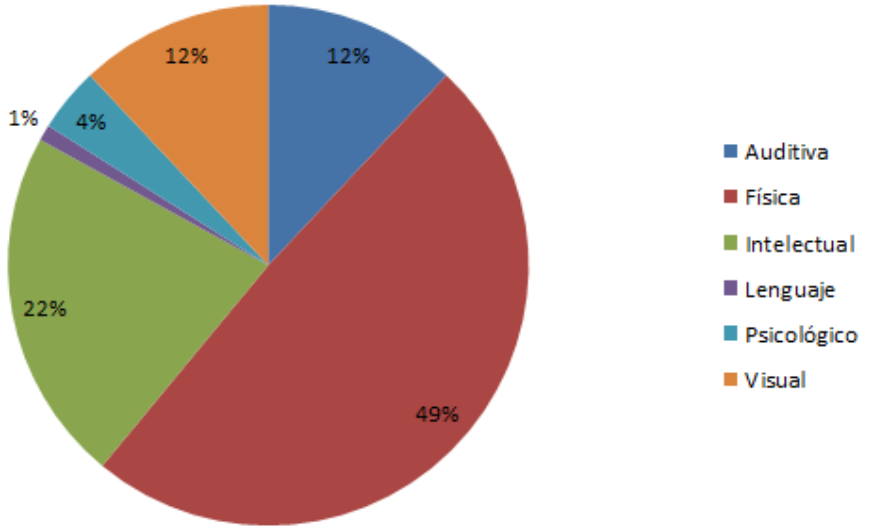
PROVINCIA	AUDITIVA	FÍSICA	INTELECTUAL	LENGUAJE	PSICOLÓGICO	VISUAL	TOTAL
AZUAY	2584	14681	4935	356	642	2864	26062
BOLIVAR	1000	2271	1253	156	161	851	5692
CAÑAR	825	2994	1544	184	280	695	6522
CARCHI	1033	2294	904	75	245	549	5100
CHIMBORAZO	2488	5214	2959	165	146	1192	12164
COTOPAXI	1361	4014	2071	283	193	1166	9088
EL ORO	1602	8145	5413	166	764	1856	17946
ESMERALDAS	994	5944	3557	238	296	1633	12662
GALAPAGOS	32	126	107	2	17	30	314
GUAYAS	8551	38929	20414	902	2866	9036	80698
IMBABURA	2234	4238	1764	143	339	1040	9758
LOJA	1553	5026	4198	154	620	1502	13053
LOS RIOS	1231	9882	3745	250	387	1729	17224
MANABI	3449	23495	5812	278	3239	5676	41949
MORONA SANTIAGO	336	1864	967	115	169	641	4092
NAPO	492	1735	851	135	75	476	3764
ORELLANA	451	2166	761	115	186	925	4604
PASTAZA	334	1078	584	28	79	324	2427
PICHINCHA	7951	23610	11122	712	2289	5981	51665
SANTA ELENA	969	4579	2077	84	162	809	8680
SANTO DOMINGO	973	5069	1971	103	401	1120	9637
SUCUMBIOS	510	2287	1118	78	178	688	4859
TUNGURAHUA	2055	4244	2502	172	321	930	10224
ZAMORA CHINCHIPE	397	1559	821	65	95	366	3303
TOTAL	43405	175444	81450	4959	14150	42079	361487

Fuente: (Conadis, 2013)

El proceso de carnetización del CONADIS es desde 1996 hasta el presente 2014, para lo cual las personas deben calificar, mediante varias pruebas médicas para obtener la inscripción en el Registro Nacional de Discapacidades y posteriormente obtener el carnet del CONADIS, lo que jurídicamente le reconoce como una persona con discapacidad.

Los tipos de discapacidades en el Ecuador son varios, los cuales se producen por diferentes causas o son hereditarios en algunos casos, mediante el registró de personas con discapacidad en nuestro país la mayor cantidad de personas tiene deficiencias físicas, para lo cual mediante la Constitución Nacional del Ecuador del 2008 se realizó políticas de crear rampas en las instituciones públicas y privadas para mejorar la movilidad del día a día de este grupo y así obtener un mejor convivir con la sociedad ecuatoriana en un ambiente de respeto.

Figura 8. Gráfica de porcentaje de discapacidad

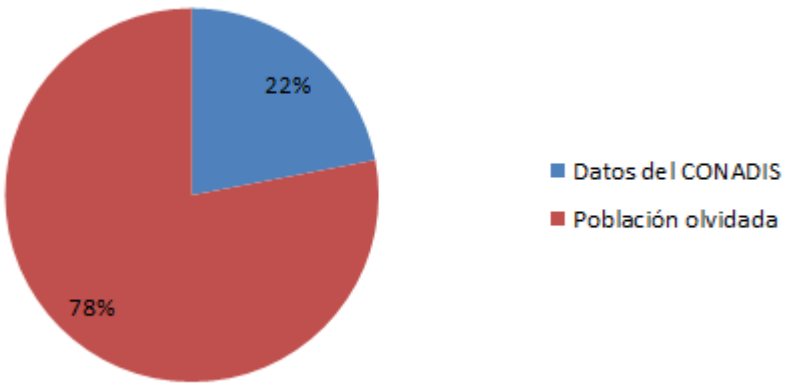


Fuente: (Conadis, 2013)

El índice de personas que tienen el carnet del CONADIS es bajo con respecto a ciudadanos que tienen algún tipo de discapacidad, los cuales por falta de información, no cumplir requisitos burocráticos o no cumplir requisitos médicos no pueden ingresar al Registro Nacional de Discapacidad. La calificación para la obtención del carnet se da mediante tres ámbitos: el social, psicológico y médico, las cuales se dan por secuelas irreversibles, enfermedades genéticas y congénitas, además de personas por edad o actividades diarias y por tal motivo obtienen una discapacidad.

Las personas con discapacidad son muchas dependiendo del grado y porcentaje de deficiencia, obteniéndose una gran cantidad de ciudadanos sin registro.

Figura 9. Discapacidad en el Ecuador



Fuente: (INEC, 2013) (Conadis, 2013).

2.2 Discapacidad

2.2.1 Definición de discapacidad.

Discapacidad es la restricción o ausencia de capacidades, debido a una o varias deficiencias al realizar una actividad de las funciones naturales de un ser humano. Las discapacidades se origina por: accidentes, enfermedades cotidianas y hereditarias. Por lo tanto la deficiencia dependiendo del caso puede ser temporal o permanente (Nuevo León Unido, 2009).

2.2.2 Tipos de discapacidad.

Las discapacidades pueden subdividirse en cuatro tipos los cuales son: física, psíquica, sensorial e intelectual o mental. Dentro del primer tipo de discapacidades se puede manifestarse en diferentes grados, los cuales representan el rango de deficiencia o restricciones que posee una persona (Nuevo León Unido, 2009).

2.2.2.1 Discapacidad física.

La discapacidad física o motriz es aquella que produce una limitación de movimiento, en consecuencia es la parálisis de una, dos, tres o las cuatro extremidades, la cual es producida por diferentes enfermedades o accidentes que tiene el ser humano, sean estos anomalías de en la motricidad de las extremidades o deficiencias del sistema nervioso. (Vivenciasbfm, 2009).

2.2.2.2 Discapacidad sensorial.

La discapacidad sensorial recae sobre las personas con deficiencia visual, auditiva y personas con problemas en la comunicación y el lenguaje. Las cuales pueden presentar diferentes grados de discapacidad (Carvajal, 2012).

2.2.2.3 Discapacidad psíquica.

La discapacidad psíquica es cuando presenta "trastornos en el comportamiento adaptativo, previsiblemente permanentes" (Cruz Roja-España, 2001). Por lo tanto puede ser producida por diversos trastornos mentales, autismo, síndrome orgánico, entre otros.

2.2.2.4 Discapacidad intelectual o mental.

La discapacidad intelectual o mental, más conocida como retraso mental, se refiere a limitaciones sustanciales en funcionamiento intelectual, es decir que coexiste junto a dos o más limitaciones en distintas "áreas de habilidades de adaptación: comunicación, cuidado propio, vida en el hogar, habilidades sociales, uso de la comunidad, autodirección, salud y seguridad, contenidos escolares funcionales, ocio y trabajo." (Vivenciasbfm, 2009).

2.3 Hardware

2.3.1 Sensor Kinect.

2.3.1.1 Historia.

El sensor Kinect del Xbox 360 desarrollado por Microsoft, es una tecnología nueva que fue diseñada e implementada por ingenieros para tener una interfaz natural, en consecuencia la primera vez que fue presentando el dispositivo es en junio del 2009 como "Project Natal" (BotScience, 2012), el cual tuvo un nuevo relanzamiento el 13 de junio del 2010 con el nombre de "KINECT", y en Estados Unidos el mismo año el 4 de noviembre salió a la venta (xataka windows, 2013).

2.3.1.2 Características.

El sensor Kinect es un dispositivo que permite interactuar naturalmente entre una consola de juegos y diferentes aplicaciones que se han desarrollado alrededor del Kinect, gracias a las grandes ventajas que contiene el dispositivo como: una cámara RGB, un servo motor para controlar el ángulo de operación del sensor, cuatro micrófonos multi-array bidireccional y una cámara de profundidad.

Figura 10. Estructura del sensor

Fuente: (Microsoft Co., 2012)

Kinect adquiere imágenes de video con hasta 65536 niveles de sensibilidad. La cámara RGB (rojo, verde, azul) posee una resolución VGA (640 x 480 pixeles) que opera mediante un sensor CMOS y con un filtro de 30 FPS (frames por segundo o cuadros por segundo). La cámara del Kinect funciona con hardware y software propios para el reconocimiento de imágenes.

Posee cuatro micrófonos ubicados en los extremos del sensor, es la única razón para que el sensor sea de un tamaño grande, para el reconocimiento de comandos de voz para lo cual cada canal procesa 2 bytes de audio en un rango de muestreo de 16 KHz (Pensamientos computables, 2010).

Los sensores de profundidad se componen de dos partes los cuales son: un proyector de rayos infrarrojos (IR) y un sensor CMOS monocromático. El sensor de infrarrojos permite obtener datos de video bajo cualquier condición de luz, es decir obtiene la profundidad y posición a la que se encuentra un sujeto por tanto las coordenadas en 3 dimensiones, el mismo que opera con una resolución (480 x 640) pixeles con 11 bit de profundidad y 30 cuadros por segundos.

Figura 11. Partes del sensor Kinect

Fuente: **(Miranda, 2010)**

Características del sensor IR:

- Capacidad para mapear
- Rastrea 20 articulaciones por persona.
- Rastrea hasta 2 personas.

Kinect, en su posicionamiento desde el objeto a reconocer, tiene un rango recomendado de espacio para su correcto funcionamiento los cuales son: verticalmente desde los 60 cm hasta los 180 cm como referencia el suelo, el rango de ancho y de longitud más acertado es entre los 180 cm y los 360 cm. Pero se debe evitar colocar el sensor que tenga luz directa del sol.

Figura 12. Ubicación del sensor

Fuente: (Microsoft Co., 2012)

2.3.1.3 Funcionamiento.

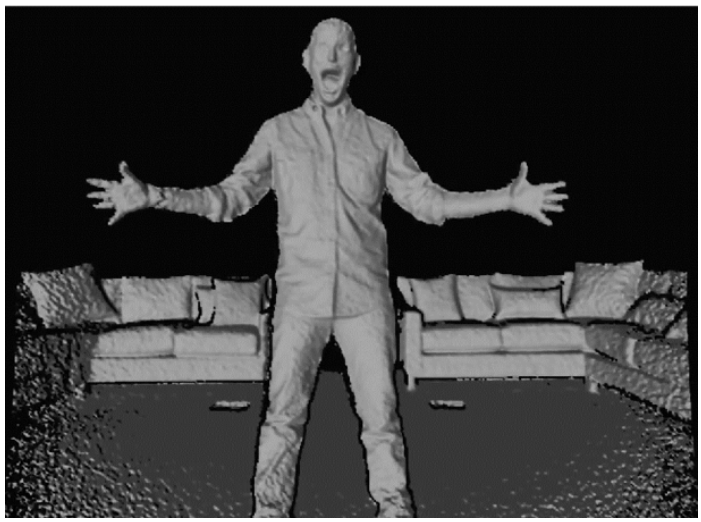
Los elementos del sensor Kinect tienen diferentes funcionalidades las cuales hacen que el sensor sea capaz de capturar el mundo a su alrededor mediante la información procesada por los diferentes elementos.

Para el caso de la captura en 3D, Kinect hace uso de su sensor infrarrojo, empleando una técnica llamada 3D Time of Flight, la cual emplea sensores CMOS de bajo costo. Este sistema trabaja iluminando la escena que se desea registrar mediante una fuente de luz modulada y captando la luz reflejada, el cambio de fase entre la luz incidente y reflejada se mide y se la transforma en distancia, para facilitar la medición. Se emplea como señal de la fuente de luz incidente, una onda cuadrada teniendo un emisor que puede ser un led o diodo laser que emitan la señal en el rango de los infrarrojos.

Ya que Kinect no solo toma una muestra de un punto sino que realiza un barrido pixel por pixel de manera continua, este proceso requiere que la señal incidente sea además de tipo continua, obteniendo múltiples muestras por medición teniendo cada muestra un desfase de 90° con un total de 4 muestras, las cuales permiten conocer, a través de las ventanas de fase, la carga acumuladas en las mismas y por medio de ellas saber el ángulo de fase que existe entre la luz incidente y la reflejada. Éste

último valor permite conocer la distancia que existe entre el sensor infrarrojo y el objeto en cuestión.

Figura 13. Recepción de escala de grises



Fuente: (xataka windows, 2013)

En la fase de calibración, el sensor de luz infrarroja emite sobre un plano de la escena la cual entrega las medidas de distancia en tres dimensiones y captura la imagen del patrón proyectado obteniendo imágenes de referencia. Para el funcionamiento se utiliza las fotografías almacenadas en el proceso de calibración para virtualmente sustituir al emisor IR y tener un margen de error mínimo. Para tener una reconstrucción de un esqueleto que se asemeje a un cuerpo humano.

El caso de la cámara RGB sirve para clasificar lo que es el fondo y lo que es el usuario, ya que la cámara RGB y la infrarroja se encuentran alineadas horizontalmente, es posible sabiendo quien es el usuario, centrarse únicamente en el modelo 3D del mismo y discernir gracias a un software propio de Microsoft, cuales son las partes correspondientes al cuerpo y en qué posición se encuentran.

Figura 14. Esqueleto de orientación

Fuente: (Abrego, 2013)

El micrófono multi-array es un sistema de varios micrófonos en fila y la recepción del sonido es en todas las direcciones. En el sensor Kinect tiene 4 micrófonos distribuidos, tres del lado izquierdo y el último a la derecha, todos en la parte inferior del sensor, los cuales simulan el funcionamiento de los oídos además de facilitar trabajos de filtrado eliminando todo lo que esté fuera del rango de la voz humana y de esta manera se filtra los ruidos del ambiente y se amplifica la voz.

2.3.2 Ordenador.

2.3.2.1 Importancia.

El ordenador es una de las piezas fundamentales para el desarrollo de aplicaciones con Kinect, el computador es el encargado de obtener el máximo provecho del hardware y software a través de los paquetes para Windows: Software Development Kit (SDK) y las licencias para la elaboración del proyecto; los cuales son compatibles con aplicaciones generadas con C++, C# o Visual Basic utilizando Visual Studio 2010 o 2012.

El ordenador que emplee Kinect debe manejar grandes cantidades de información, por lo tanto se necesita un PC que disponga un procesador rápido y una tarjeta gráfica potente para el procesamiento de diferentes aplicaciones que se van a realizar en el proyecto.

2.3.2.2 Características de compatibilidad del ordenador con Kinect.

Los requerimientos mínimos necesarios que el ordenador necesita para poder trabajar apropiadamente con el sensor Kinect son los siguientes.

- Procesador de 32 bits (x86) o 64 bits (x64).
- Procesador de doble núcleo a 2,66 GHz o superior.
- Bus USB 2.0 dedicado.
- 2 GB de RAM.
- Tarjeta gráfica compatible con DirectX® 9.0c

2.3.3 Arduino.

2.3.3.1 Historia.

Arduino fue creado como un proyecto educativo en el año 2005, por la necesidad de que no se cierre el instituto de diseño interactivo IVREA en Italia. Por lo tanto al crear un producto de hardware de libre (Hardware Open Source), no podría ser cerrado el centro (BotScience, 2012).

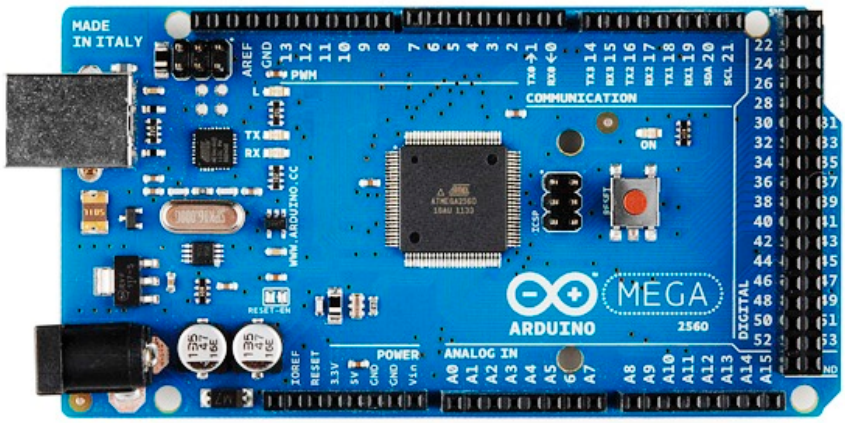
Los principales creadores fueron Massimo Branzi y David Cuartelles, los cuales formaron un equipo de trabajo para el desarrollo del proyecto. La primera producción fue de 300 unidades y fueron vendidas a los propios alumnos de IVREA con ganancias de alrededor de un dólar por prototipo, lo cual incentivo al desarrollo y creación de nuevos modelos manteniendo la plataforma de prototipos de electrónica de código abierto en hardware; software flexible y fácil de usar. Las extensiones de las placas se pueden ensamblar fácilmente y el sistema de programación se puede descargar gratuitamente. (Arduino Co., 2012)

2.3.3.2 Arduino Mega 2560.

La tarjeta Arduino Mega 2560 fue escogida para el proyecto por tener varias características favorables para desarrollar con facilidad el sistema de asistencia. Posee varias salidas analógicas, siendo éstas necesarias para algunos actuadores.

Las tarjetas Arduino se consideran un dispositivo multiplataforma (Windows, Linux y Mac), lo que las hace atractivas para que cualquier tipo de usuario desarrolle aplicaciones en diferentes áreas como por ejemplo: redes de sensores, alarmas, robótica sencilla, sistemas de comunicación, domótica, etc. (Arduino Co., 2012)

Figura 15. Arduino Mega 2560



Fuente: (Arduino Co., 2012)

2.3.3.2.1 Ventajas.

- Hardware extensible (Hardware Open Source).
- Entorno de programación simple y directo.
- Económico.
- Conexión USB con el ordenador.
- Multi-Plataforma.
- Simplifica procesos al trabajar.

2.3.3.2.2 Características.

- Microcontrolador: ATmega2560
- Entrada de voltaje: 7-12V
- Voltaje de operación:5V
- Velocidad del reloj: 16MHz
- I/O Digital: 54
- Entradas Analógicas: 16
- PWM: 14
- UART: 4
- Memoria Flash: 256Kb
- Bootloader: STK500v2
- Interfaz de programa: USB vía ATMega16U2

2.4 Software

2.4.1 Visual Studio C# 2010 Express.

2.4.1.1 Introducción.

C# (C SHARP) es un lenguaje de programación, que ha heredado características añadidas de Java y varios lenguajes orientados a objetos y la utilización de la base de C/C++, brindando un entorno de trabajo amigable al programador.

El programa para el desarrollo de la aplicación del proyecto es Microsoft Visual C# 2010en la versión Express, por la ventaja que su base de desarrollo es gratuito, incluyendo servidores SQL en la versión compacta y los SDK (Software Development Kit), lo cual permite crear programas propios con C#.

Figura 16. Símbolo C#

Fuente: (Microsoft Co., 2012)

2.4.1.2 Aplicación WPF C#.

Windows Presentation Foundation (WPF) es un sistema para crear aplicaciones cliente de Windows, con grandes prestaciones desde el punto de vista visual, ya que la base de WPF es un motor de presentación e independiente de la resolución y extiende su núcleo con características de desarrollo que incluye: el lenguaje XAML (eXtensible Application Markup Language), multimedia, texto, gráficos 2D y 3D, animación, documentos, enlace de datos, estilos, plantillas y elementos de la biblioteca de clases .NET Framework (Microsoft Developer Network, 2013).

El proyecto será realizado en el sistema WPF por las factibilidades que se brindan al manejar librerías multimedia, texto, animaciones, estilos y datos que se manejarán, para al desarrollo del sistema de asistencia y obtener un entorno amigable para el ser humano.

2.4.2 Kinect SDK v1.7.

SDK (Software Development Kit) es un conjunto de herramientas de desarrollo de software con las cuales se puede realizar aplicaciones interesantes para un sistema mediante sensores de Windows, en este caso se utilizara el sensor Kinect.

El desarrollo de los controladores de Kinect para Windows es igual a muchas aplicaciones y dispositivos, en el SDK de Kinect entrega todo el soporte para el control las características del sensor, como las librerías para algunos lenguajes de programación.

El SDK v1.7 incluye nuevos controles con respecto a versiones anteriores oficiales y betas, permitiendo al usuario que la interacción con el sensor sea de forma más natural y sencilla.

2.4.3 Speech Recognition.

El Kinect también puede trabajar como entrada de audio lo cual permite reconocer comandos de voz dados por el usuario. El diccionario de la herramienta speech permite realizar el reconocimiento de palabras y frases en diferentes idiomas, pero al inicio del desarrollo de la herramienta solamente se reconocía el idioma inglés, pero gracias a su desarrollo, hoy en día se tiene diferentes diccionarios de reconocimiento de voz en los idiomas: español, francés, japonés y el original en inglés.

2.4.3.1 Requisitos de utilización.

Para que Kinect realice el reconocimiento de voz es necesario instalar diferentes librerías para obtener un diccionario de reconocimiento de palabras las cuales están en diferentes idiomas, las librerías y paquetes para el reconocimiento de voz son:

- Microsoft Speech Platform
- Kinect to Windows Runtime Language Pack

Siendo estos archivos las librerías para el uso de las herramientas de reconocimiento de voz y el paquete del diccionario temático del idioma o su variante, ya que en diferentes regiones o países puede hablarse un mismo idioma, pero las palabras, modismos y acentos varían.

CAPÍTULO 3
CARACTERÍSITICAS DEL PROYECTO

3.1 Características específicas empleadas en el proyecto

3.1.1 Reconocimiento de voz.

Para emplear Kinect para de reconocer la voz del usuario, se debe configurar diversos parámetros como: el tipo de modulación, el número de muestras por segundo, el canal del sensor empleado y los bytes que se transmiten del lado del sensor. Para el caso del proyecto se ha utilizado una modulación por código de pulsos PCM, se toman 16000 muestras por segundo, se ocupa 16 bits por muestra, se utiliza un solo canal de audio, y envía 32000 bytes al ordenador por segundo, de este modo, el establecimiento de etas características serían del siguiente modo.

Figura 17. Reconocimiento de voz

```
miKinect.AudioSource.Start(),
newSpeechAudioFormatInfo(EncodingFormat.Pcm, 16000, 16, 1,
32000, 2, null));
```

Elaborado por: Fabián Quishpe & Diego Ulloa

Ahora, para extraer y comparar las palabras hay que emplear ciertos métodos dentro de la programación en Visual Studio, los cuales ayudados por el archivo auxiliar XML que va agregado al proyecto, compara los resultados que entrega el sensor con los grupos de palabras encontrados dentro del archivo adjunto, y de estos grupos, de encontrar concordancias, puede ejecutar algún comando que se desee. El código empleado para que el sensor compare el audio del sensor con las palabras deseadas de acuerdo al lenguaje español es el siguiente.

Figura 18. El código para comparar el audio del sensor

```
privatestaticRecognizerInfoGetKinectRecognizer()    {
      foreach (RecognizerInfo recognizer
inSpeechRecognitionEngine.InstalledRecognizers())  {
string value;
recognizer.AdditionalInfo.TryGetValue("Kinect", out value);
if ("True".Equals(value, StringComparison.OrdinalIgnoreCase)
&&"es-ES".Equals(recognizer.Culture.Name,
StringComparison.OrdinalIgnoreCase)) {
returnrecognizer;
                }
            }
returnnull;
        }
```

Elaborado por: Fabián Quishpe & Diego Ulloa

3.1.2 Reconocimiento de imágenes.

En este proyecto existe, dentro de la pantalla principal del proyecto, un cuadro que muestra al usuario el entorno sobre el cual se encuentra tomando muestras, de este modo el usuario es capaz de enterarse si él se encuentra dentro del rango del cobertura del sensor. Para configurar este objeto es necesario especificar en el código si es que existe un sensor Kinect conectado al ordenador del siguiente modo:

Figura 19. Reconocimiento del sensor Kinect conectado

```
if (KinectSensor.KinectSensors.Count> 0)
```

Elaborado por: Fabián Quishpe & Diego Ulloa

De hallarse un sensor conectado, lo siguiente que especifica en el código es cual sensor Kinect se va a emplear, ya que pueden estar conectados más de uno, en el caso del proyecto sólo se utiliza uno de este modo el código necesario para este fin es el siguiente:

Figura 20. El sensor que se va utilizar.

```
miKinect = KinectSensor.KinectSensors[0];
```

Elaborado por: Fabián Quishpe & Diego Ulloa

Luego, para configurar directamente la cámara RGB se emplea el siguiente código donde se especifican la resolución de la cámara a utilizar, y el número de cuadros por segundo que puede ocupar Kinect para este fin.

Figura 21. Configuración de la cámara RGB

```
miKinect.ColorStream.Enable(ColorImageFormat.RgbResolution640x
480Fps30);
```

Elaborado por: Fabián Quishpe & Diego Ulloa

Dentro de la configuración de los frames de la cámara del sensor Kinect es necesario configurar ciertos aspectos como: la resolución y las muestras que adquiere por segundo, las cuales son requeridas para reconstruir y mostrar en el ordenador la imagen del entorno a la que esta direccionada la cámara RGB.

Figura 22. Configuración de los frames de la cámara

```
imageKinect.Source = BitmapSource.Create(colorFrame.Width,
colorFrame.Height, 96, 96, PixelFormats.Bgr32, null, pixels,
stride);
```

Elaborado por: Fabián Quishpe & Diego Ulloa

3.1.3 Sensor de movimiento.

El sensor de movimiento de Kinect se halla formado por un emisor y un sensor infrarrojo, los cuales trabajan en conjunto con la cámara RGB, para ubicar al cuerpo del usuario del sensor y extraer de este la ubicación tridimensional de diversos puntos del cuerpo humano. Kinect es capaz de extraer las coordenadas de 20 diferentes puntos y que para el caso del proyecto se han utilizado las coordenadas horizontal y vertical del codo y mano derecha del usuario.

Para configurarlo, primero es necesario especificar de cual sensor se va a extraer los valores del cuerpo, para ello se coloca el siguiente código, dentro del método de configuración del Skeleton de Kinect.

Figura 23. Configuración del Skeleton de Kinect

```
Skeleton[] skeletons = newSkeleton[0];
```

Elaborado por: Fabián Quishpe & Diego Ulloa

Dentro de este mismo método se crea una matriz de las coordenadas de todos los puntos que Kinect es capaz de extraer, en esta la matriz se almacena los datos del siguiente modo.

Figura 24. Matriz de las coordenadas

```
skeletons = newSkeleton[skelFrame.SkeletonArrayLength];
skelFrame.CopySkeletonDataTo(skeletons);
```

Elaborado por: Fabián Quishpe & Diego Ulloa

Finalmente para extraer las coordenadas de los puntos para su evaluación posterior en el programa se necesita el uso de una función denominada MapSkeletonPointToColorPoint, la cual entrega como resultado las coordenadas en x, y o z del punto evaluado, esta función recibe como parámetros, el nombre del punto a evaluar, y el formato de color del espacio.

Figura 25. Coordenadas de los puntos para su evaluación

```
varcolorPoint =
this.miKinect.CoordinateMapper.MapSkeletonPointToColorPoint(sk
elpoint, ColorImageFormat.RgbResolution640x480Fps30);
```

Elaborado por: Fabián Quishpe & Diego Ulloa

3.1.4 Comunicación serial.

La comunicación entre el ordenador y la tarjeta Arduino, es una comunicación serial a través del puerto USB, en el caso del proyecto, se ha utilizado una comunicación a través del puerto COM3 asignado a la tarjeta Arduino conectada al PC, con una

velocidad de 9600 bps, sin bits de paridad, 8 bits de comunicación y uno de parada, quedando su configuración de la siguiente forma.

Figura 26. Comunicación serial

```
puerto = newSerialPort("COM3", 9600, Parity.None, 8,
StopBits.One);
```

Elaborado por: Fabián Quishpe & Diego Ulloa

En el caso de la transmisión se requiere enviar a la tarjeta Arduino los valores de las 6 variables que se controlan, para esto es necesario declarar una matriz de bytes con las variables de los objetos que son modificados y para su envío por el puerto serial es obligatorio dar parámetros como: el nombre de la matriz a ser transmita, los bits de offset o posición de desplazamiento y el número de bytes que se transmitirán.

Figura 27. Matriz de transmisión por el puerto serial

```
byte[] envio = newbyte[] { (byte)(int_luz), (byte)(ang_cama),
(byte)(ang_puerta), (byte)(aper_ventana), (byte)(ang_persiana)
};
puerto.Write(envio, 0, envio.Length);
```

Elaborado por: Fabián Quishpe & Diego Ulloa

3.2 Diseño del proyecto

3.2.1 Análisis de los formularios de C#.

Este proyecto se ha desarrollado en Microsoft Visual Studio 2010 utilizando para ello, el lenguaje de programación C# y el uso de los SDK de Kinect para el desarrollo de una aplicación en un sistema de asistencia para discapacitados, la cual permite a una persona el control de parte de su entorno dentro de una habitación, en este caso el control comprende: la iluminación, apertura y cierre de puerta, ventana y persianas, la inclinación y altura de la cama donde se halle el usuario.

Ya que el sensor Kinect permite el reconocimiento de voz y partes del cuerpo humano, el sistema diseñado se ha pensado para personas que sean discapacitadas de sus extremidades inferiores, pero sean completamente capaces de mover su brazo derecho y expresar su voz para darle órdenes al programa.

El proyecto en cuestión tiene que activar diferentes actuadores, en su mayoría servomotores, para lo cual requiere que el ordenador transfiera la información de los movimientos del cuerpo del usuario hacia cada uno de ellos empleando una tarjeta de adquisición de datos la cual se encuentra conectada al ordenador mediante la interfaz de comunicación USB.

3.2.1.1 Formulario portada.

El formulario portada tiene por objetivo ser la presentación del proyecto, donde se halla además, del nombre del proyecto, se halla indicado los integrantes que desarrollaron el proyecto, el tutor del mismo, y finalmente un botón que me permite ingresar al llamado window3, el cual contiene toda la parte operativa del proyecto.

Figura 28. Portada de la aplicación

Elaborado por: Fabián Quishpe & Diego Ulloa

3.2.1.2 Formulario window3.

En este formulario se muestra la interfaz gráfica propia de la aplicación del proyecto, en él se muestran diversos indicadores los cuales visualizan los valores o estados de cada una de las variables correspondientes a cada uno de los actuadores, dichas variables se modifican una a la vez manipulándolas a través de la diferencia de coordenadas entregadas por el sensor entre las posiciones del codo y la mano derecha. La adquisición de coordenadas del cuerpo humano se lo realiza gracias a

una de las herramientas de Kinect denominada Skeletal Tracking, la cual extrae las coordenadas de 20 puntos del cuerpo.

Al momento de manipular las variables, si la posición de la mano se encuentra por sobre la posición del codo y refiriéndose a la posición horizontal, la mano se encontrará a la derecha del codo, la variable debería aumentar, y por el contrario si se encuentra del lado izquierdo, la variable disminuirá su valor, por este motivo existen dos indicadores en forma de círculos que en su interior, contienen un signo más y un signo menos respectivamente para que el usuario observe si la variable está incrementando o esta decremento.

Existen también otros indicadores de texto, los cuales despliegan el valor de cada una de las variables que se está controlando y en qué estado se encuentran las demás funciones. Para mostrar al usuario qué función del programa se encuentra manipulando, el recuadro con el nombre del objeto que se esté controlando y el fondo del indicador de la variable cambia de tonalidad de un color jade a otro azul opaco. Cuando la función está activa, mientras que, el resto de las variables no manipuladas se mantienen en el matiz jade. Para complementar la visibilidad del objeto que se está operando también se muestra un gráfico correspondiente a la función que se está utilizando.

Tabla 3. Colores de los indicadores.

Estado de las funciones	**Color del indicador**
Manipulada	
No manipula	

Elaborado por: Fabián Quishpe & Diego Ulloa

Otra de las características del sensor Kinect empleado en este proyecto ha sido el reconocimiento de voz y dentro de este proyecto, es de gran importancia ya que gracias a esta herramienta el usuario es capaz de moverse entre las diferentes funciones del programa: iniciar o parar la manipulación de una variable, enviar a todo el sistema a un modo de reposo entre otras.

Finalmente aunque solo de manera ilustrativa existe un indicador para mostrar al usuario lo que esta, al frente del sensor Kinect, esto es gracias a que el mismo posee entre sus diferentes sensores una cámara de tipo RGB.

Figura 29. Panel de control del sistema

Elaborado por: Fabián Quishpe & Diego Ulloa

3.2.1.3 Anexo SpeachGramar.

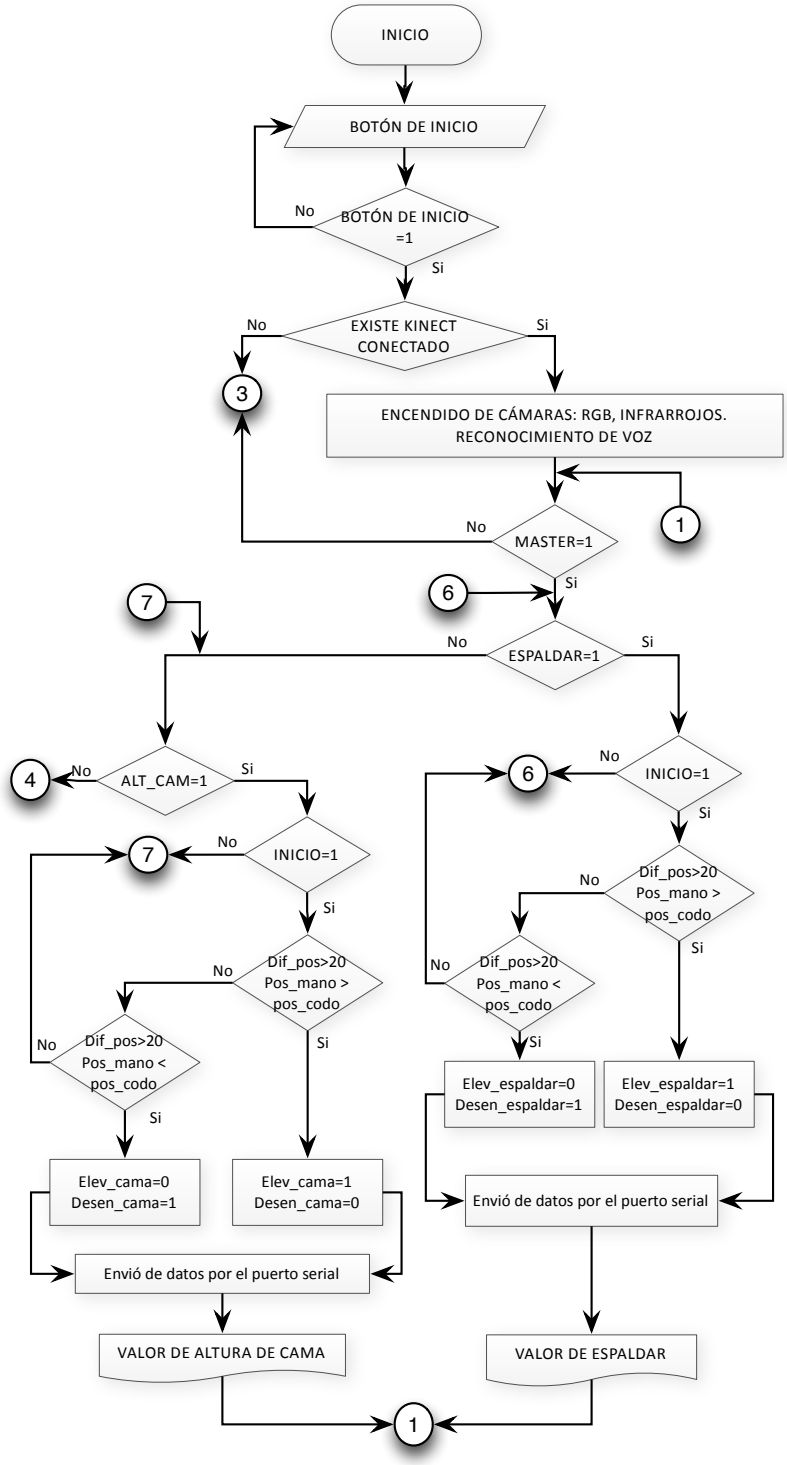
Este archivo dentro del proyecto permite juntar frases o palabras en un solo grupo o conjunto para usarlo posteriormente en la herramienta de reconocimiento de voz, ya que, las palabras pronunciadas por el usuario son convertidas en un objeto tipo String y comparados con las palabras o frases dentro del archivo. Además, es necesario especificar el tipo de idioma personalizado instalado en el sistema operativo, en el caso del proyecto es el español nativo de España, ya que existe también la versión mexicana y por defecto se instala la versión de inglés.

3.2.2 Diagramas de flujo.

3.2.2.1 Programa en C#.

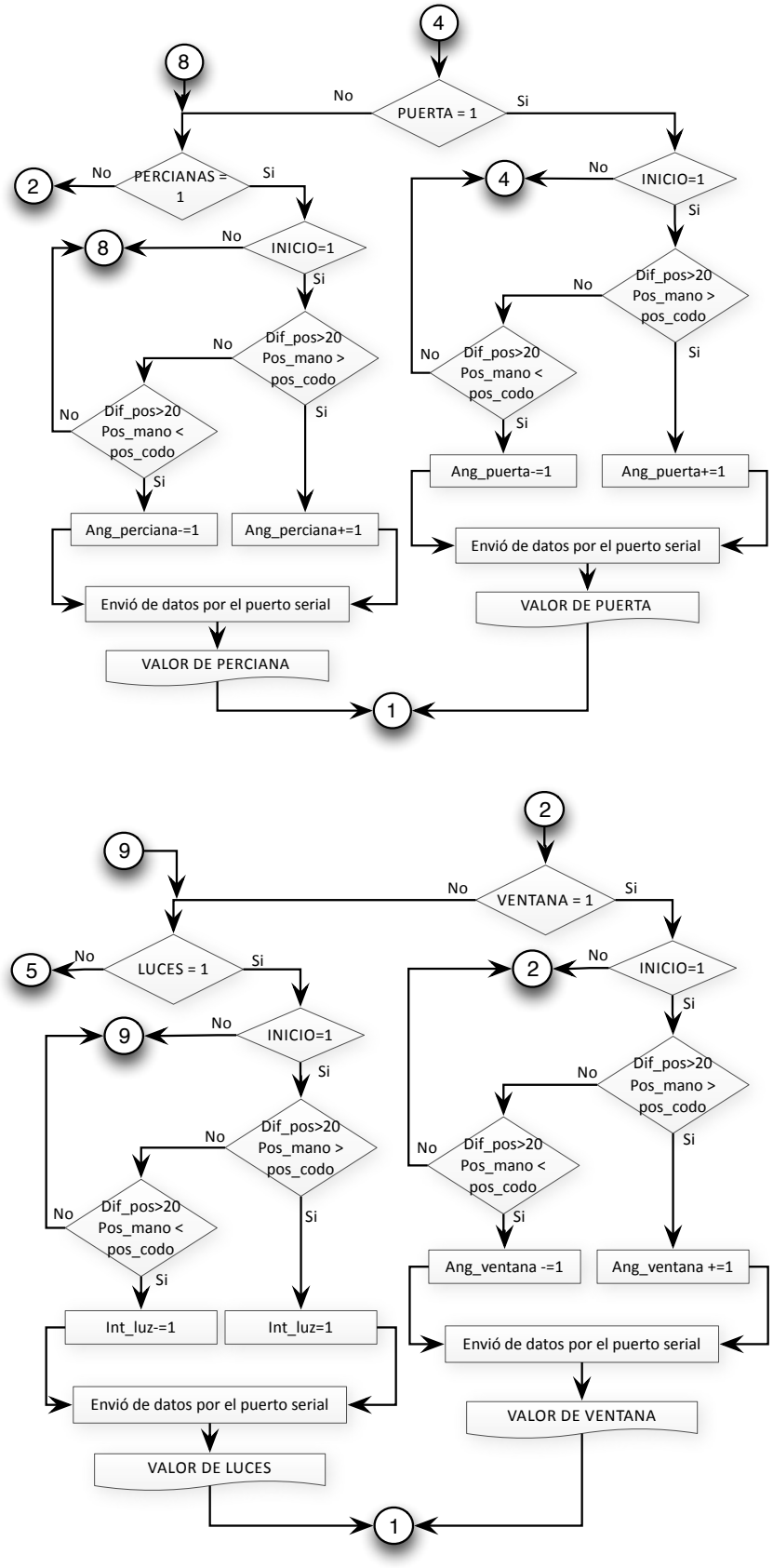
3.2.2.1.1 Estructura Principal.

Figura 30. Diagrama de flujo del código de C#.



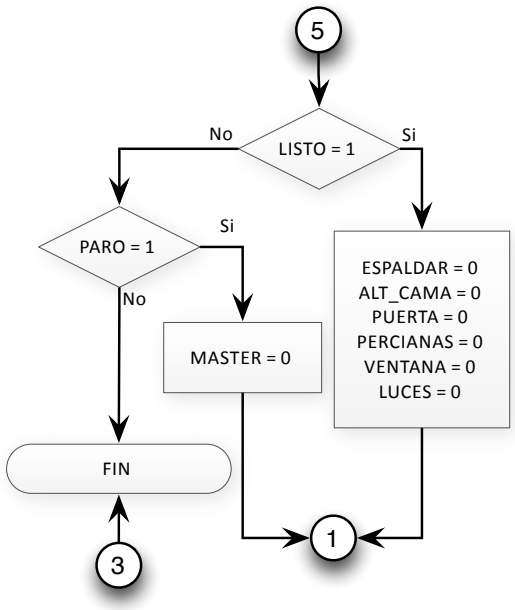
Elaborado por: Fabián Quishpe & Diego Ulloa

Figura 30. Diagrama de flujo del código de C# (continuación).



Elaborado por: Fabián Quishpe & Diego Ulloa

Figura 30. Diagrama de flujo del código de C# (continuación).



Elaborado por: Fabián Quishpe & Diego Ulloa

El diagrama muestra el orden estructural de cómo se ha procedido a desarrollar el programa en C#, empezando por el cambio de la carátula hacia la ventana de control (window3) del programa.

Ya en la ventana window3; lo que se considera en primer lugar es la existencia del sensor conectado al ordenador, comprobado esto, se procede a la configuración y encendido de cada una de las etapas del sensor, como son los sensores infrarrojos, la cámara RGB, y los módulos de reconocimiento de voz.

Ya con el sensor configurado el programa espera la orden del arranque principal, la cual debe ser activada a través de un comando de voz, este permite evitar accionamientos errados por parte del sistema.

Una vez estando el sistema operando, luego de recibir la orden del arranque principal el sistema espera una segunda orden de voz que le permita ingresar a una de las diferentes etapas de las que está comprendido el sistema.

Ya dentro de alguna de las opciones, que representan las etapas del sistema, estas aun no pueden manipularse, ya que es necesario darle una orden más para que las

variables logren ser modificadas, así mismo esto se lo realiza a través de una orden de voz.

Todo lo anterior mencionado es empleando únicamente el módulo de reconocimiento de voz del sensor Kinect, pero en este punto del programa empieza a operar el sensor infrarrojo, ya que una de las funciones propias del sensor Kinect es el reconocimiento de las partes del cuerpo humano, en este caso el reconocimiento de 20 puntos del mismo los cuales a través del sensor infrarrojo, se entregan las coordenadas de los puntos como si fueran vectores. De allí es factible la comparación de los mismos, y por ello se extraen las coordenadas del codo y la mano derecha.

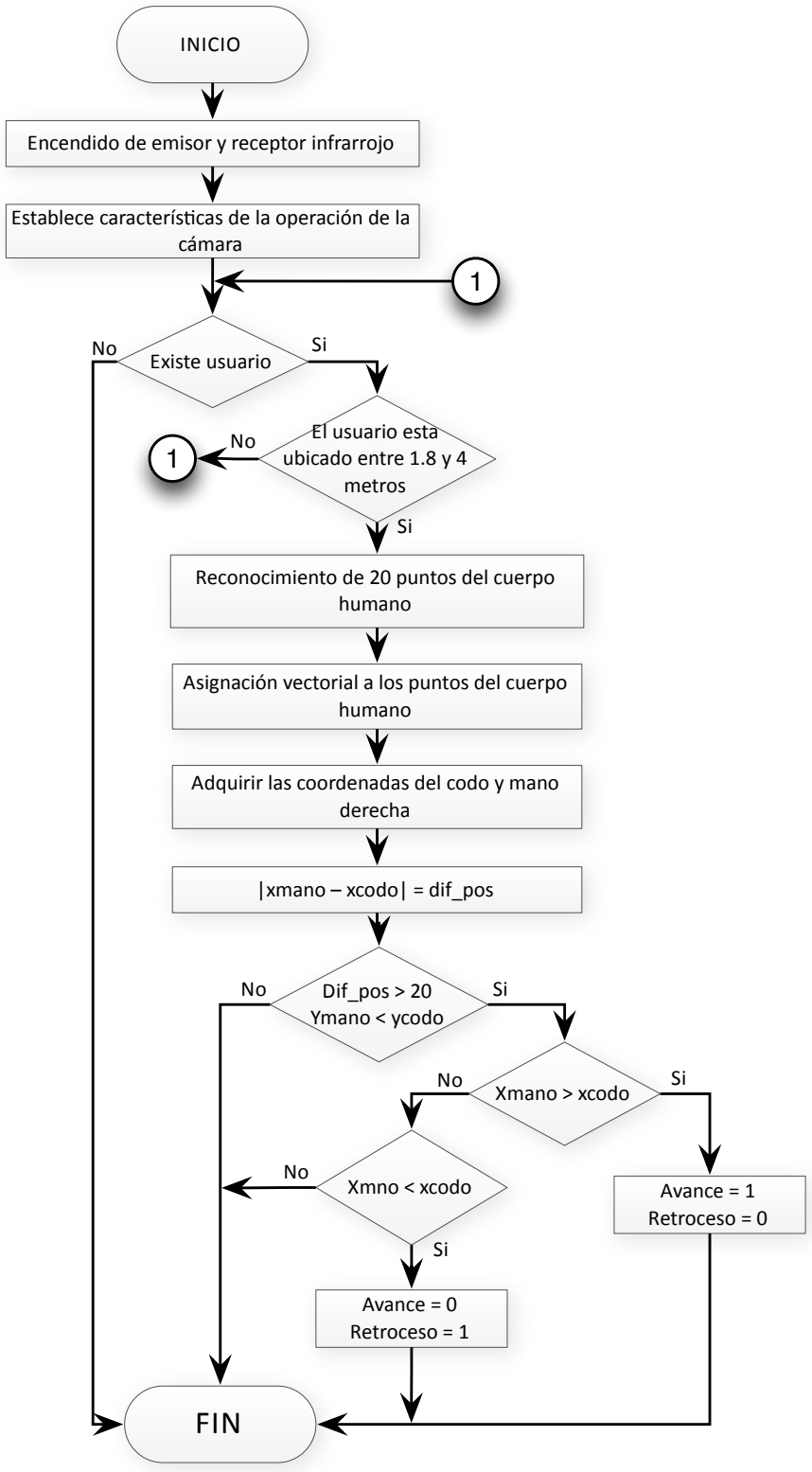
La comparación de la mano y el codo, en el caso del programa establece que: para que la variable se incremente, la mano debe estar por sobre la altura del codo e inclinada hacia el lado derecho del usuario, del mismo modo para que la variable a modificar se reduzca, la mano debe estar por sobre la altura del codo e inclinada hacia el lado izquierdo, en este caso, estar a la misma altura sobre el eje horizontal no ejerce ningún cambio de variable.

Una vez realizados los cambios sobre las variables el programa envía, a través del Puerto USB, únicamente los cambios de estados de las variables. Para dejar de manipular una variable es necesario darle al programa una orden de voz, la cual permite bloquear el cambio de valor del estado de alguna de las etapas del programa.

Finalmente para bloquear el ingreso a la manipulación de las variables es necesario dar una orden de voz más, la cual anulará la orden del arranque principal ubicada al inicio del diagrama, esta orden es también activada luego de transcurrido un periodo de tiempo de inactividad del sistema.

3.2.2.1.2 Skeleton Tracking.

Figura 31. Diagrama de flujo de adquisición de datos del Skeleton Tracking.



Elaborado por: Fabián Quishpe & Diego Ulloa

En el diagrama de flujo del Skeleton Tracking, se explica la operación del módulo del sensor infrarrojo en el código del programa, en este caso, la utilización de la herramienta skeleton tracking permite ubicar 20 puntos del cuerpo humano de manera vectorial.

En primer lugar, para emplear esta herramienta se requiere encender la cámara infrarroja y establecer los parámetros en los cuales va a operar. A través de la herramienta skeleton tracking, lo primero que se confirmará es la existencia de una persona frente al sensor, para ello esta persona debe ubicarse a una distancia de entre 180 cm a 380cm del sensor, los cuales son los rangos de operación apropiados para el mismo, de no estarlo, el sensor arrojará valores erróneos.

A continuación se procede a llamar al proceso del skeleton tracking para que ubique los 20 puntos del cuerpo del usuario. Estos valores se almacenan en una matriz, cuyos objetos pueden ser llamados de acuerdo a su nombre y coordenada.

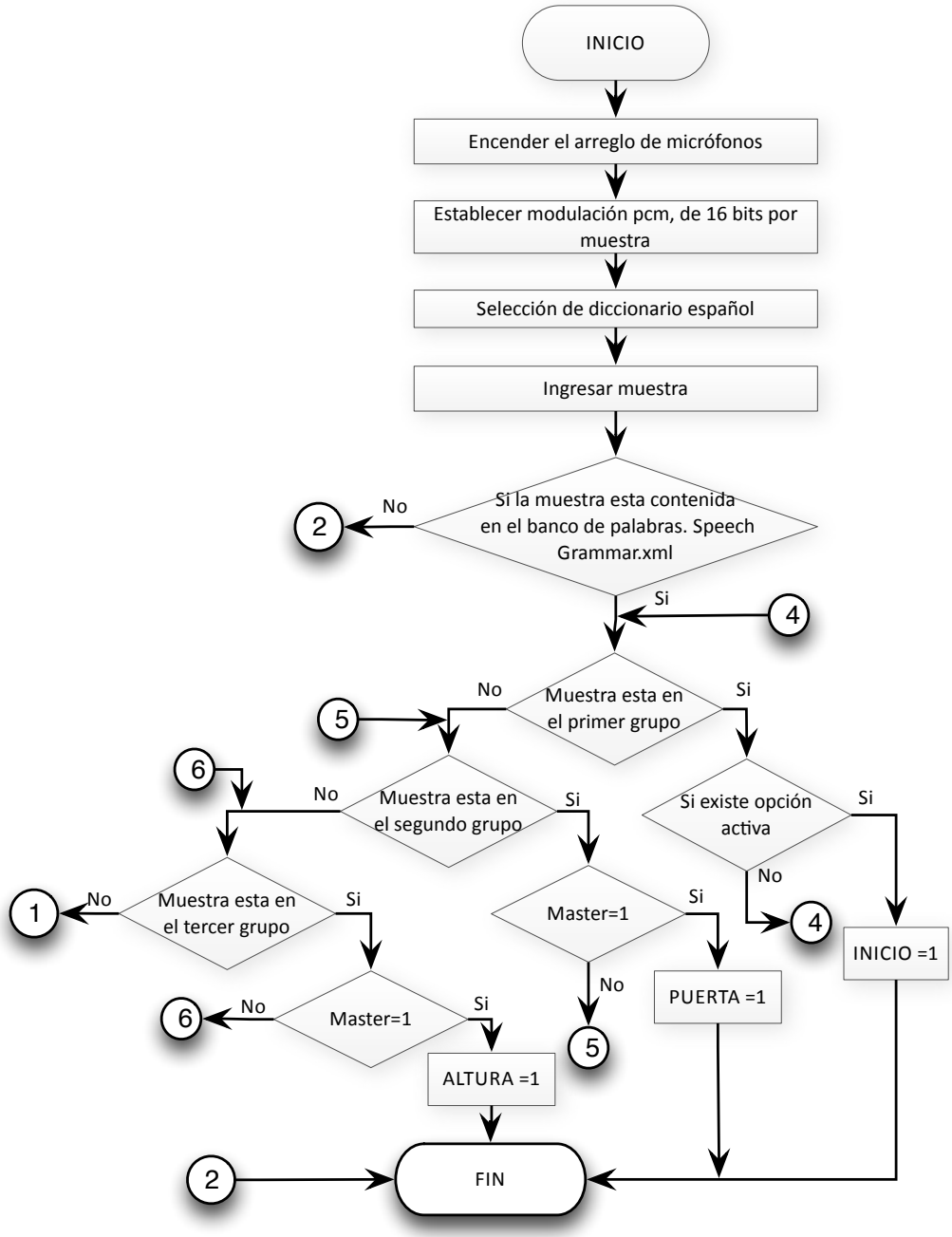
De entre estos objetos sólo son de importancia para el programa las coordenadas horizontal y vertical del codo y la mano derecha.

En este caso como en el programa se necesita que la mano se encuentre en cierta posición respecto del codo, en el caso de la coordenada vertical únicamente se necesita que ésta indique que la mano se halla por sobre el codo, pero para el caso de la componente horizontal, se ha considerado que ante existencia de valores iguales no lo considere como una orden para el programa, esto por consideraciones de diseño, por ello se trabaja con el valor absoluto de su diferencia.

Del mismo modo, cumpliendo con cierto margen de diferencia entre las posiciones horizontales del codo y la mano, ésta última, debe estar sobre el codo, la cual se procede a comparar únicamente los valores horizontales de las mismas para discernir si debe ser considerado un cambio incremental o de reducción, por lo mismo se considera que si la mano se encuentra a la derecha del codo la variable debe incrementase, pero de ser el caso contrario, esta se verá reducida, estas consideraciones se las ha hecho desde el punto de vista del usuario.

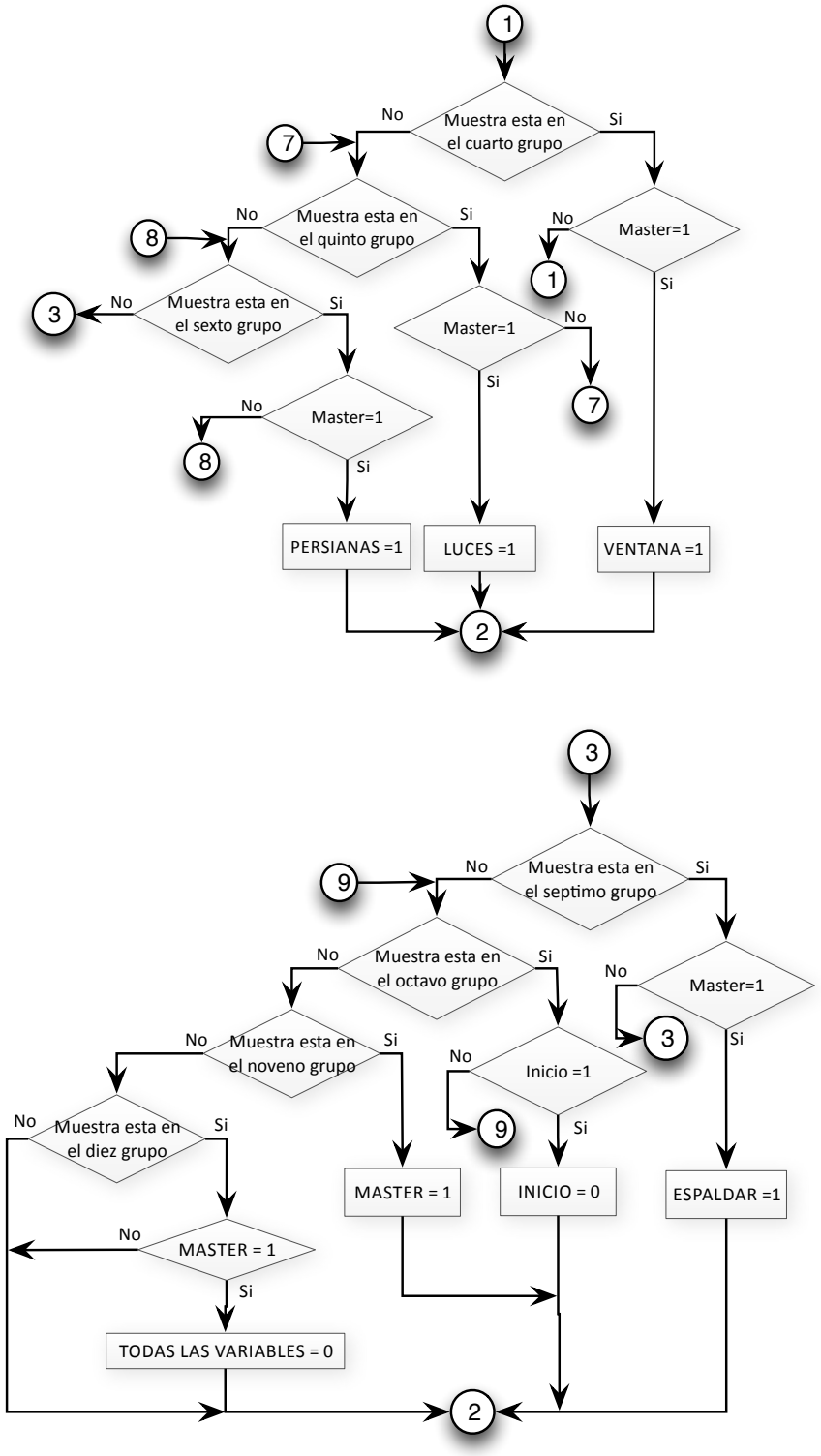
3.2.2.1.3 Reconocimiento de voz.

Figura 32. Diagrama de flujo del empleo del reconocimiento de voz en C#.



Elaborado por: Fabián Quishpe & Diego Ulloa

Figura 32. Diagrama de flujo del empleo del reconocimiento de voz en C# (continuación).



Elaborado por: Fabián Quishpe & Diego Ulloa

Se da inicio al algoritmo al ingresar al panel de control del programa, mediante el cual se enciende el arreglo de micrófonos, se configura la modulación PCM de 16 bits por muestra, para mayor fidelidad en la comparación de la muestra y también se selecciona el idioma del diccionario.

El Kinect recepta todas las muestras de sonido captadas por el arreglo de micrófonos que posee el sensor, para comparar dichas señales con el diccionario de referencia Speech Grammar.xml.

Las muestras se comparan con las respectivas frases y palabras, para ingresar a la respectiva función que contienen el sistema de asistencia.

Si la muestra está en el primer grupo, se activa la función de inicio, la cual solamente es activa, si la función principal y un comando de control esta activo.

Si la muestra está en el segundo grupo, se activa la función de puerta, la cual solamente es activa, si la función principal está activa.

Si la muestra está en el tercer grupo, se activa la función de altura, la cual solamente es activa, si la función principal está activa.

Si la muestra está en el cuarto grupo, se activa la función de ventana, la cual solamente es activa, si la función principal está activa.

Si la muestra está en el quinto grupo, se activa la función de luces, la cual solamente es activa, si la función principal está activa.

Si la muestra está en el sexto grupo, se activa la función de persianas, la cual solamente es activa, si la función principal está activa.

Si la muestra está en el séptimo grupo, se activa la función de espaldar, la cual solamente es activa, si la función principal está activa.

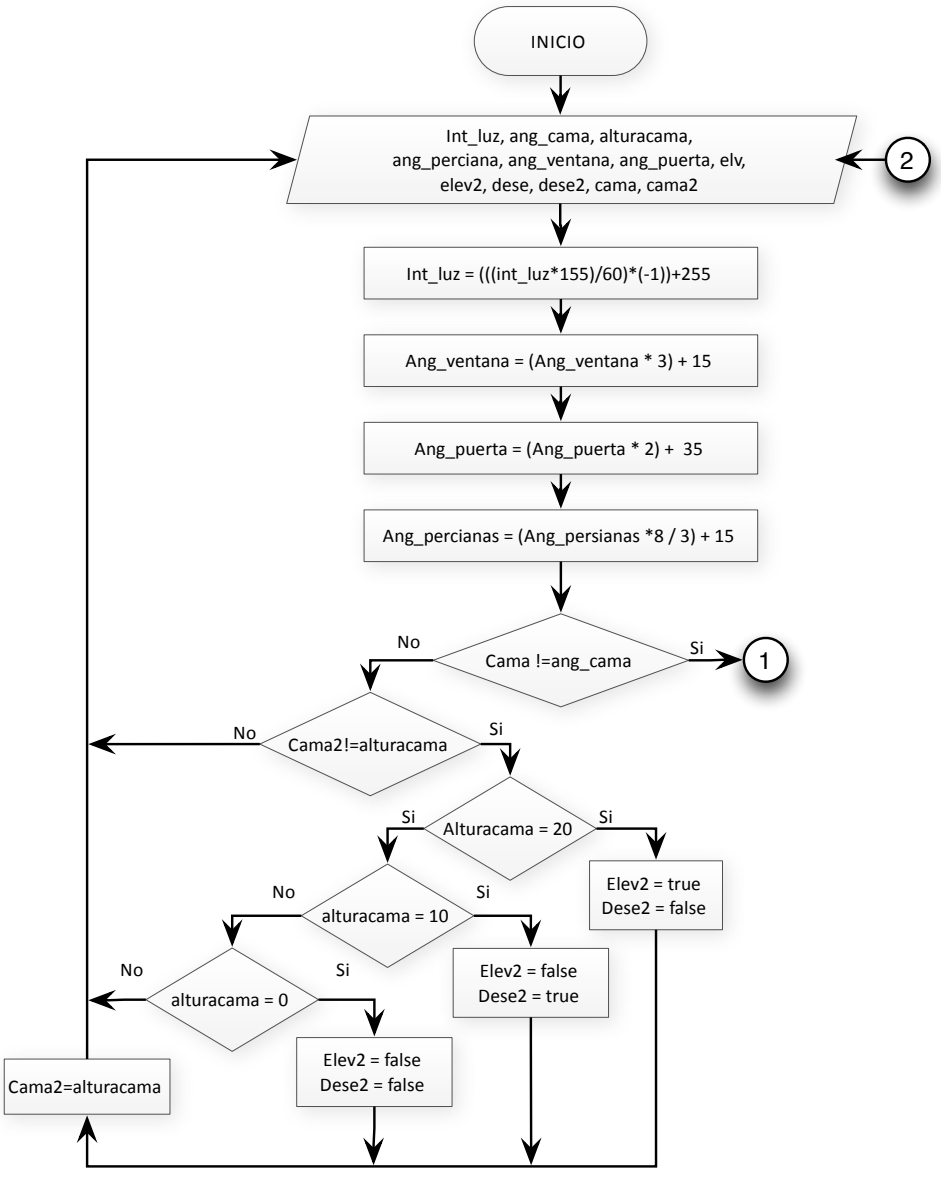
Si la muestra está en el octavo grupo, se activa la función de listo, la cual desactiva a las funciones de control y de inicio.

Si la muestra está en el noveno grupo, se activa la función de máster, la cual puede activar los comandos de control de los dispositivos.

Si la muestra está en el décimo grupo, se activa la función de paro, la cual a todas las variables del programa las desactiva y al programa del sistema se coloca en reposo.

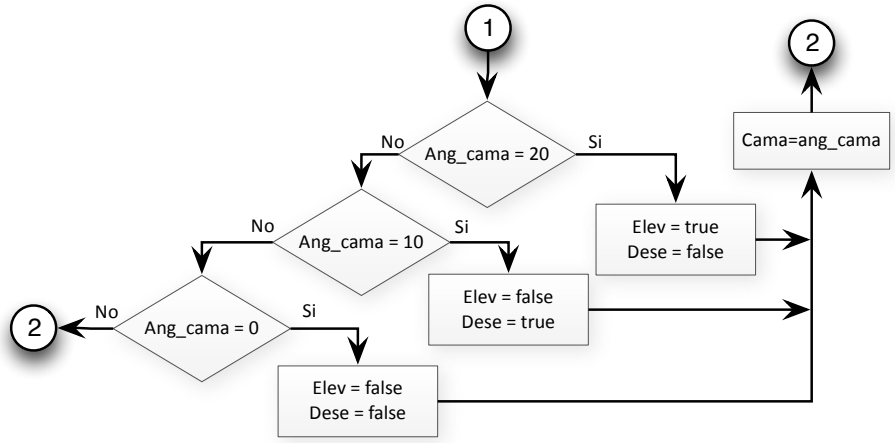
3.2.2.2 Código Arduino.

Figura 33. Diagrama de flujo del código del programa en Arduino.



Elaborado por: Fabián Quishpe & Diego Ulloa

Figura 33. Diagrama de flujo del código del programa en Arduino (continuación).



Elaborado por: Fabián Quishpe & Diego Ulloa

En el caso de la programación realizada en el software de Arduino, ésta funcionará dependiente de lo que el software de C# le entregue, por lo tanto, el estado de las salidas depende de lo que reciba a través de la comunicación serial.

Lo primero que realiza el programa es la configuración de los puertos, en este caso se emplean para la salidas de tipo PWM, en su mayoría, a excepción del control de los movimientos de la cama, ya que para el control de la cama son salidas booleanas.

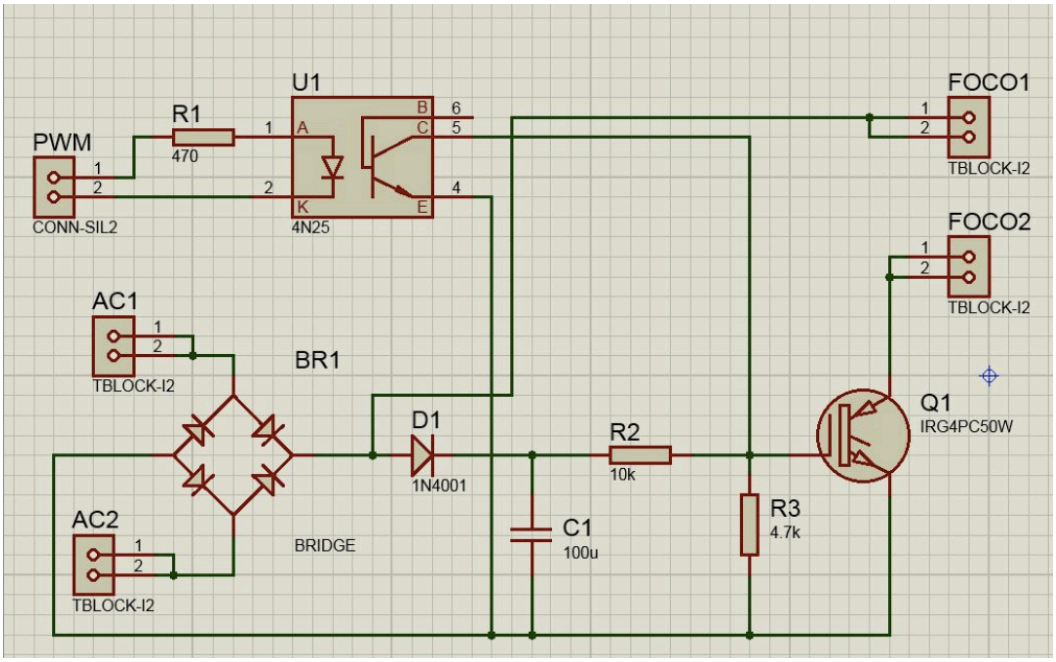
En el caso de las salidas de tipo PWM, se han utilizado de para iluminación y el control de ventana, puerta y persianas. Para la iluminación se utiliza un PWM completo, pero con lógica inversa en su rango de valores, es decir, que, para máxima iluminación el valor de PWM tiende a cero esto es debido al circuito de potencia. Mientras que para el caso de las salidas que se conectan a los servomotores, Arduino posee un método propio para trabajar con este tipo de actuadores, por lo que la variable que los active debe ajustarse entre valores de 0 y 180.

En el caso del control de la cama, como la misma se halla formada por motores DC cuyo control depende de una caja de puentes H y no se deseaba afectar a esta estructura. Se diseñó un circuito de control que remplace al control manual de la cama y ejecute los comandos generados por la placa Arduino. Por lo tanto solo existen 3 estados para cada uno de los actuadores de la cama a controlar, de este modo, el programa debe discernir en qué estado se encuentre la variable, para activar o no los actuadores.

3.3 Diseño de placas PCB

Entre las diferentes placas PCB, que este proyecto requiere, existe una placa la cual permite controlar la intensidad de luz de la habitación a través de una señal de PWM entregada por medio de la tarjeta Arduino, para ello se han utilizado optoacopladores con salida a transistor TBJ, y para controlar la carga se ha utilizado un IGBT, el circuito empleado para la construcción de esta placa es la siguiente:

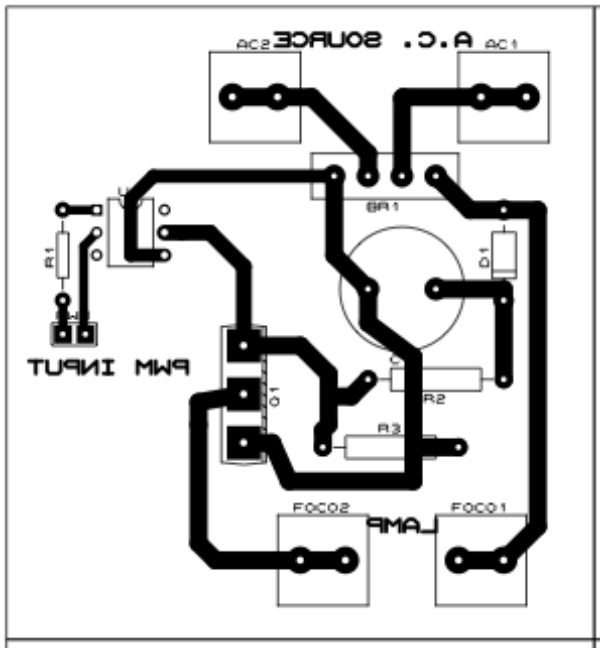
Figura 34. Diseño del sistema de luces



Elaborado por: Fabián Quishpe& Diego Ulloa

El circuito antes expuesto entregó como resultado la siguiente placa PCB:

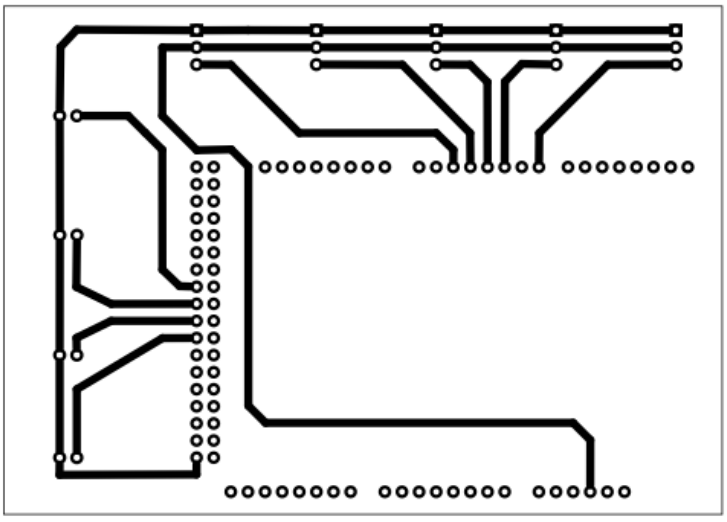
Figura 35. PCB sistema de iluminación



Elaborado por: Fabián Quishpe & Diego Ulloa

La PCB creada para brindar conectividad a la tarjeta Arduino consta de 4 salidas digitales las cuáles serán las órdenes de control para los actuadores de la cama automática, además posee 5 salidas analógicas para controlar lo que serán los motores, y la iluminación, siendo la PCB la siguiente:

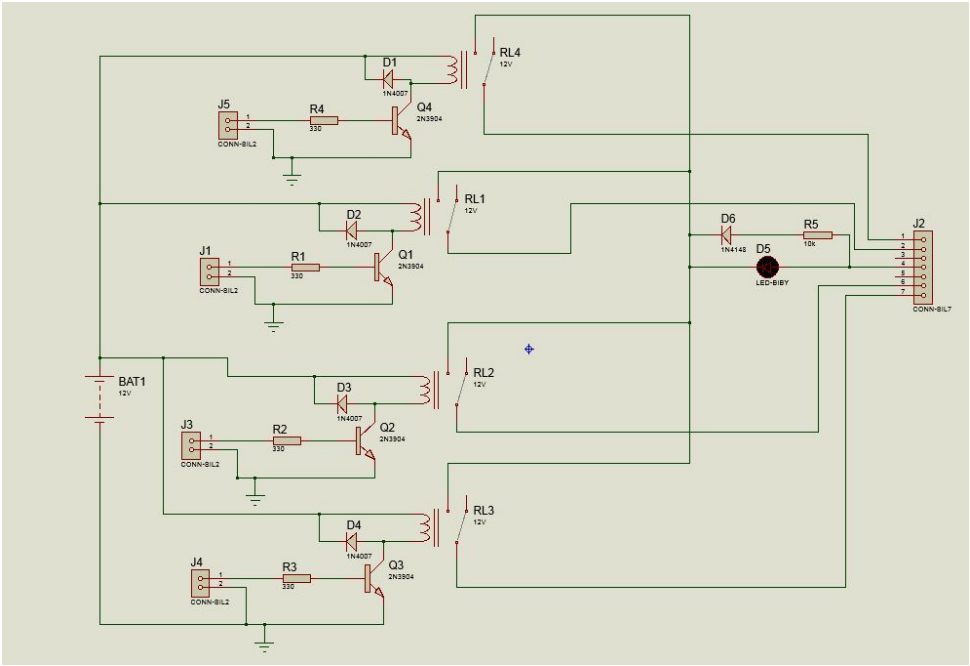
Figura 36. PCB sistema de tarjeta Arduino



Elaborado por: Fabián Quishpe & Diego Ulloa

Para el caso del control de la cama, lo que envía es un dato digital, el cual a través de un circuito de transistores y relés, permite comunicar a la caja de control de la cama que activa los motores con la tarjeta Arduino. En este caso el circuito para el control de la cama es el siguiente:

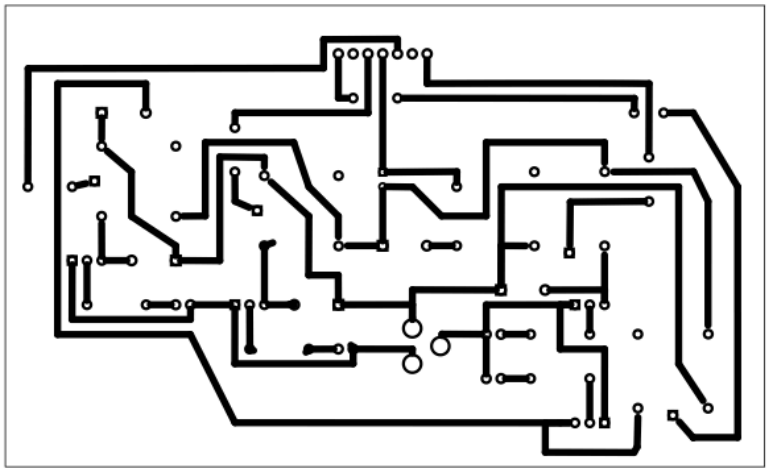
Figura 37. Circuito de control de cama



Elaborado por: Fabián Quishpe & Diego Ulloa

De este último circuito presentado, su PCB es la siguiente:

Figura 38. PCB sistema de control de cama



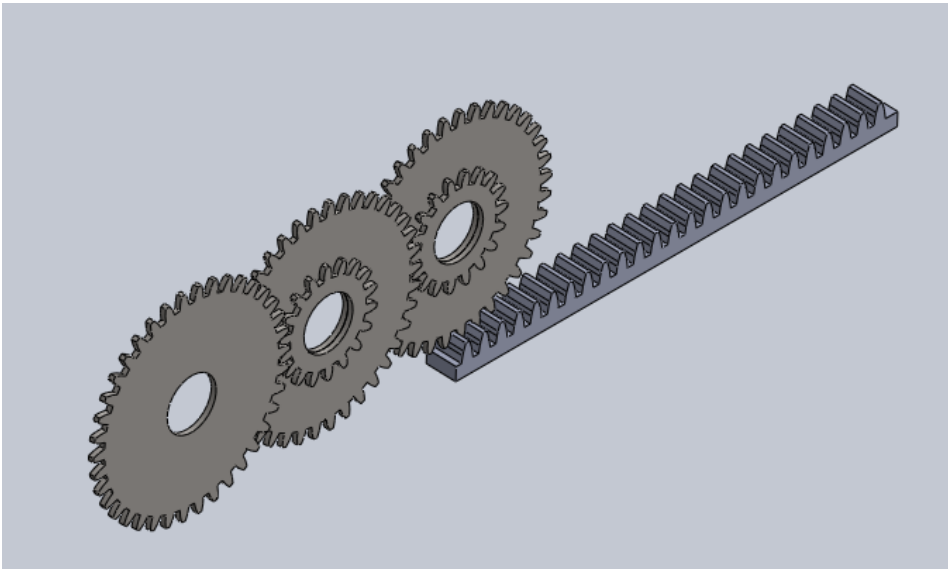
Elaborado por: Fabián Quishpe & Diego Ulloa

3.4 Caja de reducción

En el caso de la ventana, para controlar la apertura de la misma, fue necesario el uso de una caja de reducción para transformar el limitado desplazamiento de un servomotor (180 grados), en un desplazamiento longitudinal correspondiente a 25 centímetros.

Esta caja de reducción se halla formada por un piñón simple, el cual va conectado directamente al eje del servomotor, 2 piñones dobles, iguales, cuyo fin es duplicar en cada uno el recorrido del primer piñón, y finalmente una cremallera, la cual va ha estar ubicada sobre el eje vertical de la ventana.

Figura 39. Caja reductora del sistema de ventana



Elaborado por: Fabián Quishpe & Diego Ulloa

Las características de esta caja de reducción son las siguientes:

El engranaje simple, posee un diámetro de 124 mm, módulo 10, 28 dientes y un ancho de 5 mm. En el caso de los engranajes dobles, los cuales se hallan formados por 2 piñones unidos y concéntricos, en el caso del grande, este posee un diámetro de 124 mm, módulo 10, 28 dientes y un ancho de 5 mm, en el caso del pequeño, este posee un diámetro de 62 mm, módulo 10, 14 dientes y un espesor de 5 mm. Del lado de la cremallera, esta posee una longitud de 270 mm, un ancho máximo de 8 mm, una altura de 12 mm, un módulo 10, y unos 25 dientes.

3.5 Guía de instalación del software

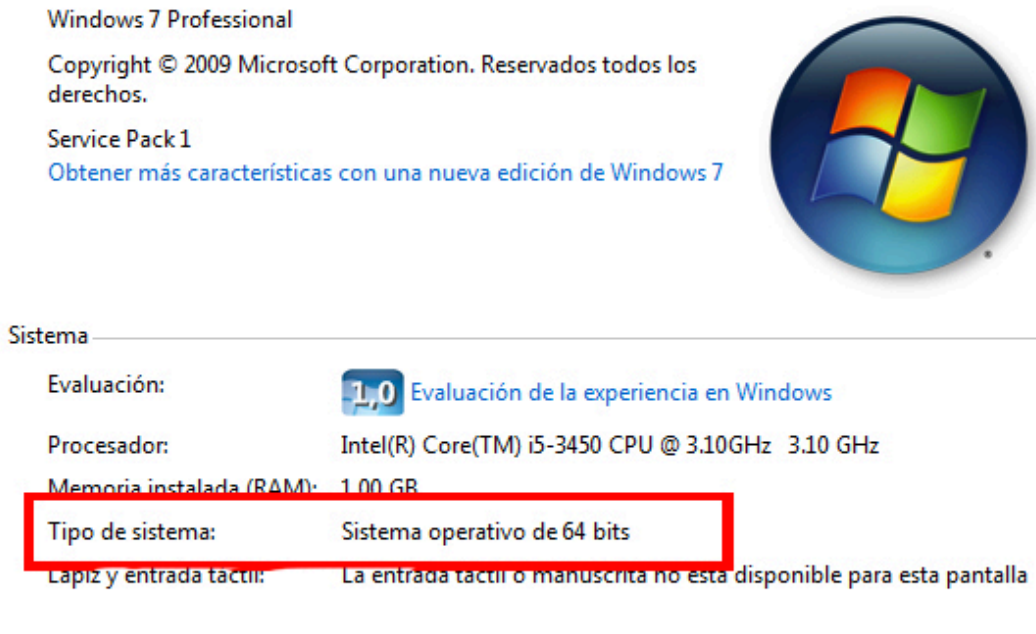
El proyecto ha sido realizado con las versiones de drivers del sensor Kinect propios de Microsoft, en este caso se ha trabajado con la versión 1.7 de los mismos, estos controladores trabajan con la plataforma de Visual Studio, desde la versión 2010, con la que se ha realizado el programa.

Las características del sensor de Kinect que se han utilizado son el reconocimiento de voz, la cámara RGB, y el sensor infrarrojo, en el caso de las cámaras se requiere la instalación de los controladores de Kinect, pero para la herramienta de reconocimiento de voz, es necesario instalar otras dependencias.

En el caso de los controladores de Kinect es necesario instalar los archivos: KinectSDK-v1.7-Setup y KinectDeveloperToolkit-v1.7.0-Setup en el orden correspondiente.

En el caso de la herramienta del reconocimiento de voz, esta requiere la instalación de 2 archivos, el primero es MicrosoftSpeechPlataformSDK, teniendo en cuenta que este depende de la versión del sistema operativo, ya que para este archivo existen versiones para sistemas de 32 y 64 bits, para verificar esto, hay que revisar en las propiedades de equipo.

Figura 40. Versión del sistema operativo

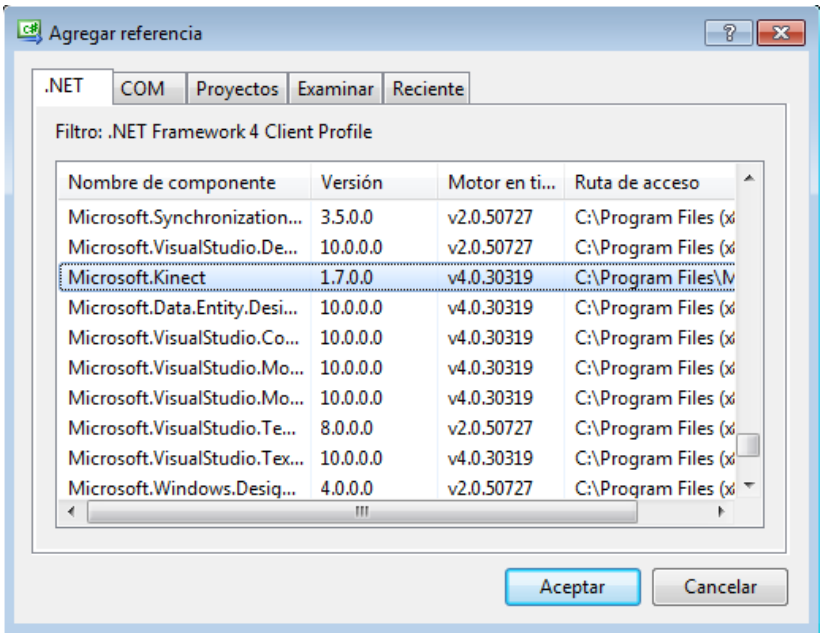


Elaborado por: Fabián Quishpe & Diego Ulloa

Una vez instalado, el software de voz, es necesario instalar los controladores para el reconocimiento de voz, este viene por defecto el idioma en inglés, en el caso del proyecto en cuestión se necesita que este utilice español, para ello se instala el paquete de lenguaje español con su variación nativa española, ya que también existe la mexicana, ejecutando el archivo KinectSpeechLanguagePack_es-ES.

Ahora, para ejecutar en el proyecto en un PC diferente al de desarrollo es necesario agregar dos referencias, una para trabajar con las herramientas comunes de Kinect, a excepción del reconocimiento de voz, para ello hay que dirigirse al menú de proyecto, y allí escoger la opción, agregar referencia, en la pestaña .Net y allí buscar la opción que dice Microsoft Kinect.

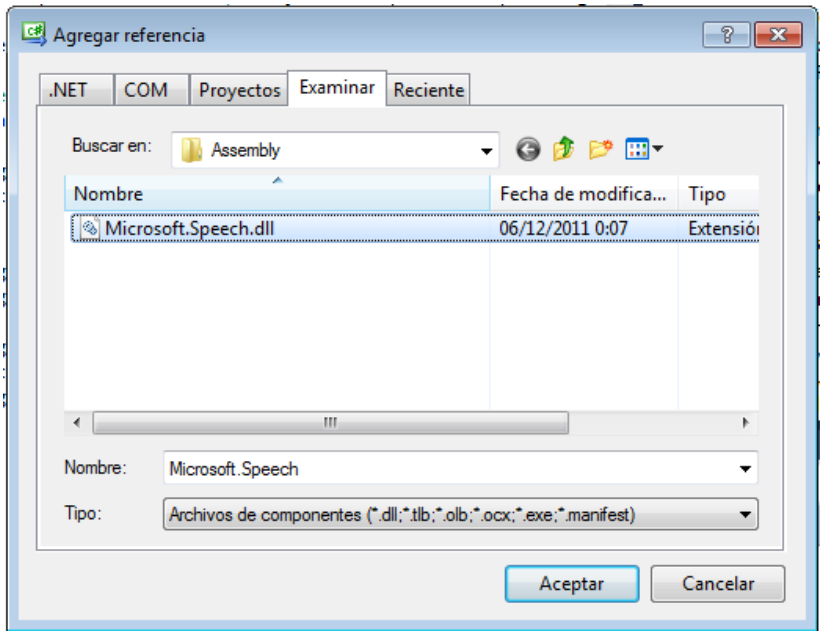
Figura 41. Referencia de Kinect en C#



Elaborado por: Fabián Quishpe & Diego Ulloa

Para agregar la relación de la herramienta del reconocimiento de voz, hay que dirigirse a la ventana agregar referencia del mismo modo que se hizo para el caso anterior, pero ahora hay que ingresar a la pestaña de examinar, allí hay que buscar la siguiente ruta: C:\Archivos de Programa\Microsoft SDKs\Speech\v11.0\Assembly, y escoger de allí el archivo Microsoft.Speech.dll.

Figura 42. Referencia de Speech



Elaborado por: Fabián Quishpe & Diego Ulloa

CAPÍTULO 4
ANÁLISIS DE RESULTADOS

A continuación se presentan la descripción del funcionamiento del sistema de asistencia para discapacitados, condiciones ideales, el alcance de los dispositivos y la guía de usuario.

4.1 Diagrama de bloques

Las etapas que forman el proyecto son diversas y acoplan el mejor funcionamiento del sistema asistencia, estas son: fuentes de alimentación (AC/DC), sensor Kinect, ordenador para el procesamiento de la información del sensor, tarjeta de control (Arduino) para controlar los dispositivos y actuadores.

Figura 43. Diagrama de bloques del sistema de asistencia.

Elaborado por: Fabián Quishpe & Diego Ulloa

4.1.1 Fuente de alimentación (AC/DC).

La fuente de alimentación energiza los dispositivos como: el sensor Kinect, el ordenador y los actuadores para los diferentes sistemas implementados. Por consiguiente el tipo de corriente (corriente directa o alterna), a utilizar, dependerá de las características que componen los elementos del sistema de asistencia para discapacitados.

4.1.2 Sensor Kinect.

En el proyecto se utiliza el sensor Kinect por sus características, las cuales permiten la recolección de diferentes datos como: voz, imágenes, coordenadas en el plano espacial, las que son obtenidas por el sensor mediante los micrófonos, cámara RGB, sensores IR receptor y emisor, respectivamente, todos los elementos mencionados tiene una conexión con el chip de adquisición de datos para la transmisión de información hacia el ordenador.

Para obtener las coordenadas del ser humano mediante el Kinect, se realiza un barrido pixel a pixel de manera continua, este proceso requiere que el emisor del proyector de luz infrarroja sea continuo. Para que el receptor obtenga múltiples muestras desfasadas en 90°, mediante la cual se define la profundidad a la que se encuentra el sujeto del sensor.

Para el reconocimiento de voz el sensor tiene un arreglo de micrófonos, los cuales están ubicados a lo largo del sensor, para que todos los sonidos producidos alrededor del Kinect sean receptados.

En el caso de Kinect se tiene 4 micrófonos en línea, tres en el lado izquierdo y otro pegado en la derecha, mirando de frente al sensor, todos situados en la parte inferior del sensor, lo cual permite saber de dónde se emite el sonido. Además realiza un filtrado, eliminando todo lo que este fuera de las frecuencias de la voz humana (80Hz y 1100Hz) y amplifica la voz teniendo en cuenta claramente los comandos de voz que tiene el diccionario. Por lo tanto permite eliminar los ecos que produce la voz al rebotar en muebles y paredes al calibrarse según la reflejo del entorno,

También contiene una cámara a color a 640X480 pixeles con sensor CMOS y autofocus.

4.1.3 Ordenador.

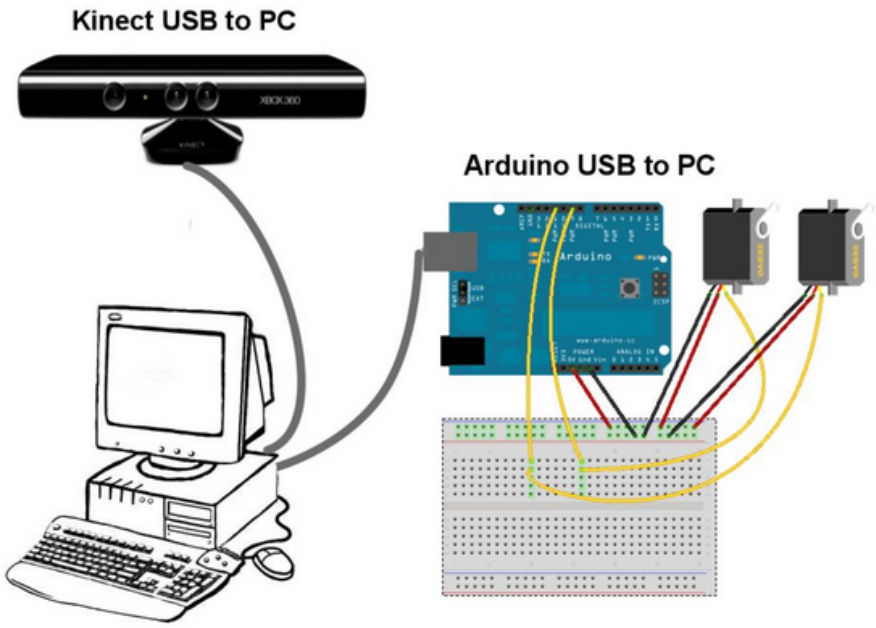
El ordenador es indispensable para el proyecto, por sus diferentes funcionalidades como: la adquisición de datos desde el sensor Kinect, el procesamiento de las señales, y la transmisión de señales de mando hacia la tarjeta de control.

El ordenador se encarga de obtener la información del sensor Kinect para procesar la información de: coordenadas, voz e imagen. Lo cual se realiza mediante las diferentes librerías que ofrece de forma gratuita Microsoft Corporation, las cuales se fusionan con el programa C#.

El programa de ejecución del proyecto se realiza en C# para procesar la información obtenida del Kinect, para elaborar los algoritmos necesarios según las funciones requeridas por el usuario para posteriormente, enviar las señales de control a la tarjeta Arduino mediante una transmisión serial.

La transmisión serial permite obtener el estado de cada uno de los sistemas asistencia y de acuerdo a la misma podemos realizar los diferentes comandos y enviar las órdenes desde la computadora hacia la tarjeta y realizar las tareas que necesita el usuario.

Figura 44. Esquema de conexión



Fuente: (Instructables, 2013)

4.1.4 Tarjeta de control.

La tarjeta de control permite la comunicación directa entre el ordenador y los sistemas de asistencia dependiendo cuál de las funciones va a realizar, para resolver el algoritmo de control y obtener la señal de la función (apertura y cierre de puertas, ventanas y persianas, el control de la intensidad de luz, la inclinación de la cama).

Para la interface entre la computadora y la tarjeta de control se realizó mediante la comunicación serial, que permite obtener datos del estado de cada uno de los puertos y si existe alguna modificación de estado en los actuadores enviar la señal a al respectivo elemento.

Para el cumplimiento de cada una de las funciones del sistema de asistencia, las señales que debe tener la tarjeta Arduino son diversas, como para el control de luces se obtendrá la señal PWM para realizar un Dimmer, las funciones siguientes de la ventana, puerta y persiana se realiza el control de un servo motor dependiendo de la opción tendrá mecánica adicional para cumplir la funcionalidad que requiere el usuario.

4.1.5 Sistema de asistencia.

El sistema de asistencia para discapacitados se desarrolló como un conjunto de procesos que ayudaran al usuario con deficiencia de movilidad a tener un mejor desempeño en su entorno y poder realizar funciones cotidianas directamente. El sistema se compone de varias etapas o salidas que son controladas de diferentes maneras, entre estas salidas existen: el control de iluminación, control del espaldar y altura de la cama, la apertura y cierre de ventanas, puertas y persianas.

Figura 45. Tarjetas PCB desarrolladas para el proyecto.

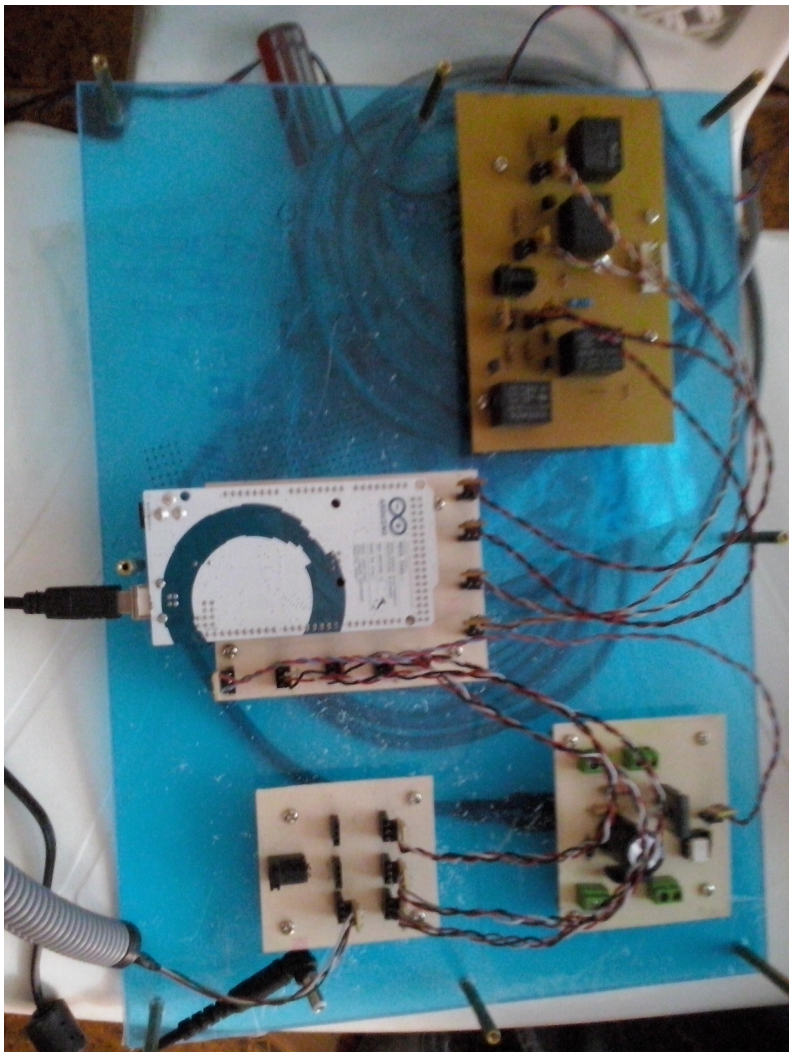


Imagen: Fabián Quishpe & Diego Ulloa

4.1.5.1 Sistema de cama.

El sistema de la cama permite el control de 2 de los 3 movimientos de una cama automática de las empleadas en clínicas y hospitales, en este caso se ha empleado una de la marca Century Medical, la cual emplea motores de corriente continua, para realizar los movimientos de la misma.

El control de giro de los motores se lo realiza a través de puentes H, para permitir al usuario un control sencillo y que los actuadores lineales de la cama no tengan movimientos bruscos para el paciente.

Internamente la caja de control de la cama únicamente tiene como control de sus movimientos finales de carrera conectados a relés que limitan el despliegue del actuador.

Por ello para controlar el movimiento de los mismos, se envían datos booleanos, los cuales, a través de un circuito que permite el reemplazo del control remoto, al emplear relés en lugar de los botones que utilizaría el usuario.

Figura 46. Tarjeta de control del sistema de la cama

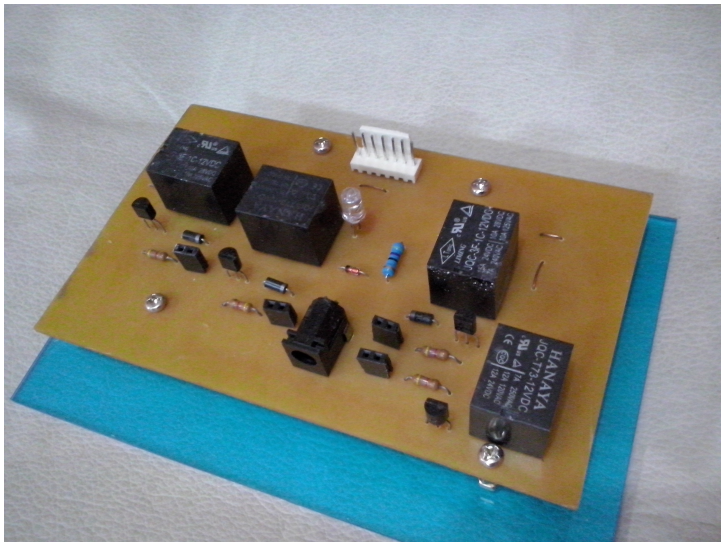


Imagen: Fabián Quishpe & Diego Ulloa

Figura 47. Módulo del control de la cama

Imagen: Fabián Quishpe & Diego Ulloa

4.1.5.2 Sistema de intensidad de la luz.

En el sistema de iluminación, se buscó controlar la intensidad de luz mediante un circuito similar a un dimmer, el cual es controlado mediante PWM proveniente de la tarjeta Arduino, el cual posee una frecuencia de operación de alrededor de 500 Hz, por consiguiente para obtener el resultado deseado se requirió el empleo de dispositivos que tengan una respuesta a frecuencia acorde a la señal de control que se recibe, en este caso se descartó por pruebas el uso de un circuito de optoacoplador a TRIAC, ya que el mismo no permitió la regulación de la intensidad de luz al no tener una buena respuesta en su conmutación, por ello se buscó como alternativa, un circuito de optoacoplador a IGBT, ya que éste permite trabajar a frecuencias en el orden de los kilohertzios, lo que permitió tener el resultado deseado.

Figura 48. Tarjeta de control del sistema de iluminación.

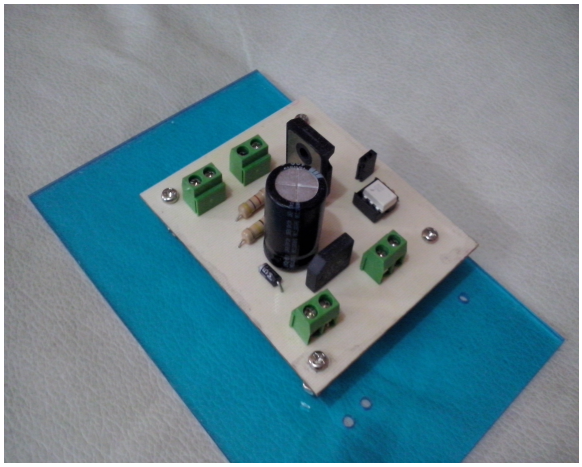


Imagen: Fabián Quishpe & Diego Ulloa

Figura 49. Módulo del control de iluminación

Imagen: Fabián Quishpe & Diego Ulloa

4.1.5.3 Sistema de ventana.

Para el control de apertura y cierre de la ventana, se buscó una solución que permitiera controlar de manera efectiva la posición en la que se encuentre la misma, siendo esta de tipo corrediza de media hoja, pero con la dificultad que el control de la misma no implique un sistema burdo que ocupe demasiado espacio y que sea de cierto modo antiestético, por este motivo se descartó el uso de sensores externos, poleas y demás dispositivos que puedan ocupar un espacio y que el usuario u otras personas puedan tener a la mano para su manipulación, ya que podría afectar su funcionamiento. De este modo se eligió por idea el uso de un servomotor.

El inconveniente del servomotor es el limitado giro que este posee, ya que comúnmente estos sólo giran alrededor de 180 grados, por lo que para resolver esta problemática, se decidió por el uso de una caja de piñones que permita multiplicar el giro del motor, para que a través del uso de una cremallera alcanzar la apertura y cierre de la ventana. Óvidamente el uso de piñones para incrementar el movimiento lineal afecta a los requerimientos mecánicos de torque de los motores, y

considerando que en nuestro país no posee industria en este campo, la selección del motor se ve limitado a pocas opciones, por lo tanto la caja de piñones que se mandó a fabricar resultó tener un tamaño relativamente grande respecto de lo planificado. Al no tener un fabricante con un torno y fresadora específica para piñones pequeños ya que este tipo de fabricantes sólo realizan trabajos al por mayor. Por estas razones y el peso del mecanismo se optó por un servomotor de 20 Kg.cm de torque.

Finalmente, para evitar que el usuario tenga el mecanismo a la vista o pueda afectarlo al manipularlo se optó por colocar todo el mecanismo y los motores internamente dentro de la pared.

Figura 50. Módulo del control de ventana

Imagen: Fabián Quishpe & Diego Ulloa

4.1.5.4 Sistema de persiana.

En el caso de la automatización de la apertura y cierre de las persianas de la habitación, se buscó del mismo modo que con la ventana el uso de un servomotor para no emplear sensores externos para regular su posición. Originalmente se trató de controlar una persiana de tipo vertical, a la cual se trató de darle el funcionamiento por medio de piñones, poleas, pero desafortunadamente el movimiento del motor hacia el eje de las persianas fue deficiente.

Debido a este inconveniente se optó por cambiar el modelo de persiana por una de tipo cebra, a la cual se le conectó directamente el motor al eje de giro de la misma, lo que permitió la apertura y cierre deseado

Figura 51. Módulo del control de persiana

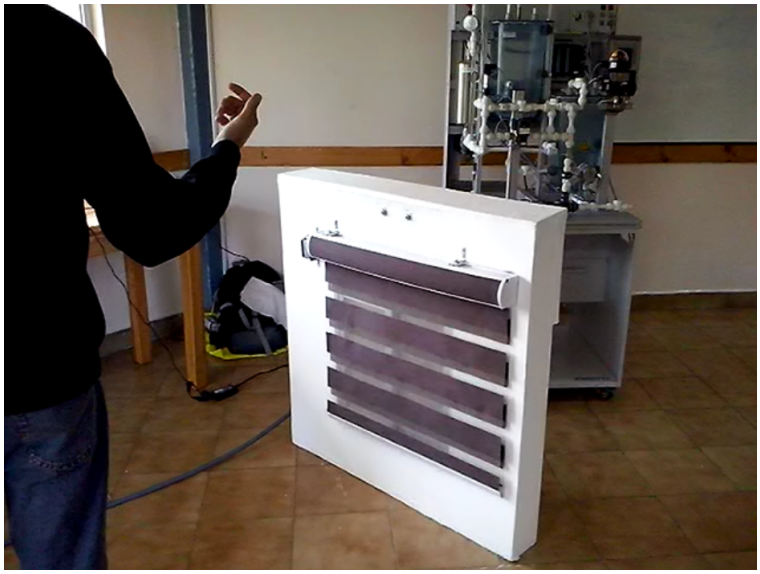


Imagen: Fabián Quishpe & Diego Ulloa

4.1.5.5 Sistema de puerta.

Para el caso del desarrollo del sistema de apertura de la puerta se procuró en un inicio el desarrollarlo con una puerta corrediza, pero debido al excesivo costo del mecanismo en el mercado o de mandarlo a fabricarlo, se investigó sobre el tipo de puertas aptas para discapacitados, con lo que una puerta abatible que se observa en cualquier lugar, cumple con los requerimientos para este fin según las normas INEN, por este motivo se optó por desarrollar un sistema similar al que se observó en algunos portones automáticos, el cual consta de un motor con un brazo que se comunica sobre un riel, para lo cual se utilizó un servomotor, en este caso por el desgaste mecánico que pueda producir en los piñones internos del motor, se optó por uno cuyos componentes mecánicos sean metálicos, y por el largo del brazo, se buscó que tenga un torque apropiado para el trabajo, de allí se tomó uno que posee 25Kg.cm.

Figura 52. Módulo del control de puerta

Imagen: Fabián Quishpe & Diego Ulloa

4.1.6 Condiciones ideales.

Para el correcto desempeño del sistema de asistencia, el sensor Kinect debe estar ubicado a una altura entre 1.10m y 1.80m, desde el nivel del piso. La distancia entre el Kinect y el usuario debe ser 1.80m y 3.80m para obtener las coordenadas de la persona y no se distorsionen con las demás extremidades, para lo cual puede realizar el respectivo algoritmo del programa y realizar la respectiva función que necesita el usuario.

El ruido en el entorno cercano del usuario dentro o cerca de la habitación debe permanecer bajo, por la utilización de la herramienta de reconocimiento de voz en el software del proyecto, ya que se requiere que se pronuncie de manera clara cada uno de los comandos de voz, además se debe tener en cuenta con la correcta posición del brazo derecho para que este sea el instrumento que permite controlar las variables que actúan sobre cada uno de los actuadores del sistema.

Como el sensor Kinect es un dispositivo de interfaz desarrollado para las consolas de videojuego, éste es capaz de reconocer al usuario que esté en frente, ya sea que se

encuentre de pie o sentado, obviamente, en este caso el usuario de ser el mismo individuo que se encuentre medianamente recostado o sentado en la cama.

4.1.7 Alcance.

El sistema de asistencia está basado en brindar facilidades al paciente para realizar actividades cotidianas, para lo cual se realiza una implementación domótica de un modelo de habitación. Teniendo como principales partes automatizadas: puerta, ventana, persiana, cama e iluminación.

En cada una de las etapas del proyecto existen consideraciones a tomar en cuenta: en el caso del control de la cama, cabe señalar que el diseño de esta etapa está expuesto con el modelo de la cama a utilizarse, en este caso el circuito de conexión entre la cama y el sistema, fue desarrollado específicamente para la cama automática de la marca Century Medical, otros fabricantes proponen circuitos diferentes.

En lo que se refiere al control de la iluminación se debe tener precaución con la potencia de las cargas (focos) a aplicar, puesto que por las características de potencia de los componentes utilizados, este se encuentra pensado para cargas de máximo 60 watts.

En el caso de la ventana, en el proyecto, ésta ha sido pensada para una ventana de medio metro cuadrado, el diseño de su mecanismo ha sido pensado para un desplazamiento de alrededor de 25 cm en línea recta, por ello en caso de requerir una ventana con un recorrido más extenso es necesario rediseñar el mecanismo para acoplarse al motor. Del mismo modo en el caso de la persiana, esta se encuentra pensada únicamente para una persiana de tipo cebra, en este caso las dimensiones de la persiana no afectan el desempeño del sistema.

Para el caso de la puerta, esta ha sido pensada para una puerta abatible de aluminio, por su ligero peso, y poco rozamiento, se debe tener en cuenta, que la puerta debe ser construida enteramente de aluminio o llevar acrílicos en lugar de vidrios, por la fuerza del motor al momento de instalar el sistema o reiniciarlo luego de un corte de luz, el motor lanzaría la puerta, lo que podría romper los cristales de la puerta.

Para el correcto funcionamiento de los servomotores, estos debe ser alimentados por fuentes externas y no depender de la alimentación de la tarjeta Arduino, ya que para

el trabajo de la puerta y ventana, ésta no aporta la energía suficiente, del mismo modo en el caso de la cama se requiere el uso de una fuente externa de 12 voltios que ara permitir a los relés conmutar.

En el caso del software, este proyecto utiliza la versión exprés de visual C# 2010 ya que la versión de Visual Studio 2012 consume demasiados recursos del ordenador lo que evita un desempeño apropiado del sistema, en el caso de que el sensor ubique a 2 personas, el reconocimiento de las partes del cuerpo se ha desarrollado para que reconozca a la persona ubicada a la izquierda.

El ordenador siempre debe encontrarse en funcionamiento y con el sistema operando, ya que este es el núcleo del sistema, y sin él los actuadores no realizarían ningún cambio.

4.1.8 Guía de usuario.

4.1.8.1 Conexión de equipos.

Previo a poner en marcha el sistema se debe realizar las siguientes conexiones, en primer lugar, conectar la tarjeta Arduino a un puerto USB del ordenador, del mismo modo se procede a acoplar el sensor Kinect a la computadora y a la fuente de energía AC.

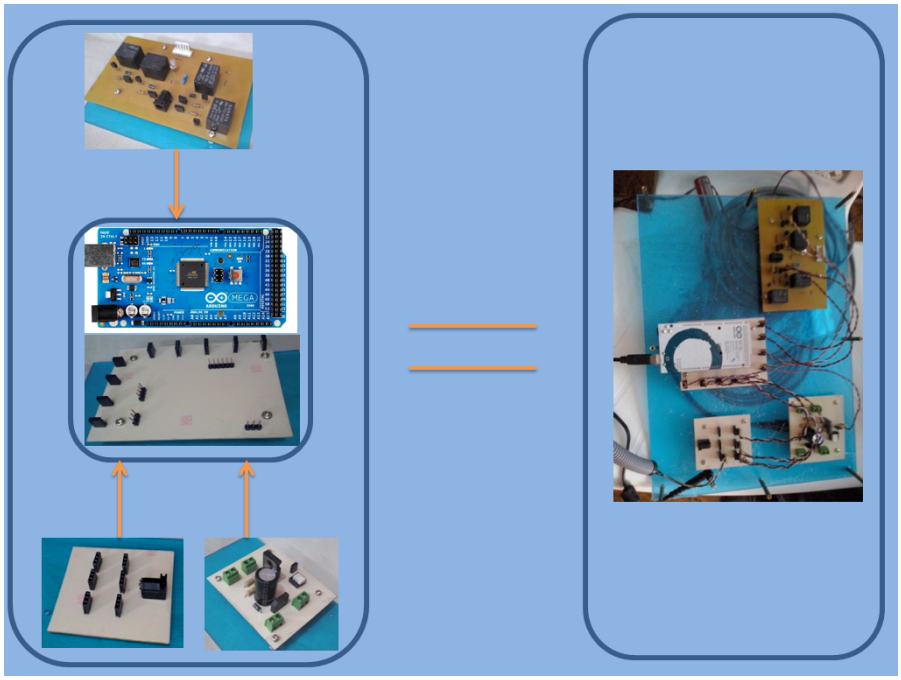
La tarjeta Arduino, se procede a unir el puerto 7 de la tarjeta con la tarjeta de control de iluminación, de la misma manera conectamos los puertos del 5 al 3 para acoplar la fuente externa para los servomotores, se debe conectar los puertos 35, 37, 39 y 41 los cuales son salidas digitales, hacia la tarjeta de control de la cama automática.

Tabla 4. Esquema de distribución de puertos en la tarjeta Arduino.

Variable	**N° de puerto en Arduino**		**Tipo de señal**
Elevación de cama	41	39	Booleano
Espaldar de cama	37	35	Booleano
Luces	7		PWM
Puerta	3		PWM
Persiana	4		PWM
Ventana	5		PWM

Elaborado por: Fabián Quishpe & Diego Ulloa

*Figura 53.*Estructura de acoplamiento con la tarjeta Arduino



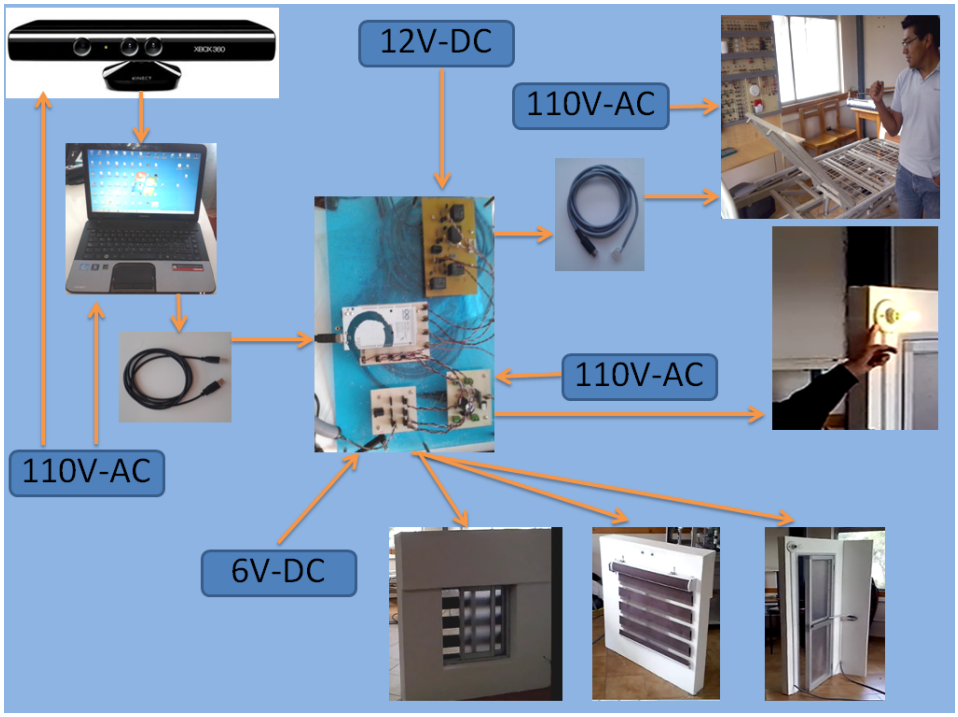
Elaborado por: Fabián Quishpe & Diego Ulloa

Se debe reemplazar de la cama automática, el cable que va ligado con el control remoto hacia la caja de control, porque la conexión entre el dispositivo anteriormente mencionado y la tarjeta elaborada para este proyecto es un cable de conexión directa entre estos dos dispositivos.

También se conecta los servomotores a la tarjeta que acopla una fuente externa para los mismos, verificando que correspondan los mismos con los pines apropiados de la tarjeta Arduino. Se procede a conectar en la tarjeta de control de luces, los cables que van dirigidos hacia la lámpara, y los que se conectan hacia la fuente de alimentación de corriente alterna.

Finalmente se procede a conectar todas las fuentes externas y la alimentación de la iluminación, en el caso de las fuentes externas, se debe colocar la fuente de 12 voltios en la tarjeta de control de la cama, mientras que la fuente de 6 voltios se aplica sobre la tarjeta de acople para los servomotores. Cabe indicar que el sensor Kinect se debe colocar a una distancia apropiada al usuario dentro de los rangos de operación del dispositivo infrarrojo los cuales están entre 1.80 y 3.80 metros, y debe enfocarlo de tal manera que a través del sensor se pueda observar con claridad las extremidades superiores del cuerpo humano.

Figura 54. Esquema de conexión del sistema de asistencia para discapacitados



Elaborado por: Fabián Quishpe & Diego Ulloa

4.1.8.2 Menú de inicio.

Una vez ejecutado el sistema de asistencia, se tendrá una portada en la interfaz gráfica que se le presenta al usuario, en la cual mediante un clic en el botón inicio ingresamos a la siguiente ventana de comandos.

Figura 55. Portada de la aplicación

Elaborado por: Fabián Quishpe & Diego Ulloa

4.1.8.3 Ventana de comandos.

La ventana de comandos es amigable con el usuario, porque cuenta con indicadores por cada función que realiza. Para cambiar el estado de una función es necesario ingresar con una orden de voz al menú haciéndole interactiva la aplicación.

Para evitar ingresos errados u órdenes que no fueron intencionadas se le ha proporcionado al sistema con una orden de arranque maestro a través de la orden de voz, **estableciendo parámetros de control,** la cual habilita el ingreso hacia las demás funciones

Una vez ingresado al menú de funciones se puede escoger la actividad que se desee realizar: el cierre y la apertura de la puerta, la ventana, persianas, la inclinación del espaldar de la cama, el control de su altura y la intensidad de iluminación. Por lo tanto a tales opciones también se debe ingresar mediante comandos de voz como

(**ventana, puerta, altura de cama, espaldar, persianas, luces**), dependiendo de la actividad que desea modificar de los parámetros de control.

Escogida la función que se desea realizar, para iniciar con la misma se debe dar otro comando de voz como **empezar o inicio**, cualquiera actividad sea esta.

Para la calibración de la función de los elementos finales del sistema se debe utilizar la mano derecha, por lo tanto si se inclina a la derecha dependiendo de la actividad se incrementara en grados o en porcentaje, y si se inclina hacia el lado izquierdo disminuirá el valor de la función que esté activa, teniendo en cuenta que el codo derecho se debe mantener fijo en un posición. También si la mano y el codo se encuentran con el mismo valor en la coordenada horizontal para el sensor, esto mantendrá la variable que se desea modificar sin cambios.

Si se han realizado los cambios deseados, para mantener el estado de éstos, se debe emplear en nuevo comando de voz para salir hacia el menú de comandos del sistema mediante la orden, **listo o terminado.**

Por cuestiones de seguridad el sistema de asistencia también tiene un paro general con el comando de voz **suspendiendo sistemas, apagando sistemas, suspensión de sistemas**.

Si la instrucción de voz es reconocida por la aplicación, el usuario obtendrá una respuesta de voz confirmando que está en el menú.

Figura 56. Panel de control del sistema

Elaborado por: Fabián Quishpe & Diego Ulloa

Tabla 5. Lista jerárquica de instrucciones de voz.

Comandos	Ingreso principal al sistema	Menú de funciones	Comandos para inicio y fin de calibración		Paro general
Explicción	Es la orden de arranque maestro, con el siguiente comando de voz.	Son las funciones de las actividades que se desea realizar, con los siguientes comandos de voz.	El comando inicio da la apertura para la modificación de la variable mientras que listo la termina.		Para seguridad del sistema de asistencia, se colocó un comando para la salida de las funciones.
Comandos de voz	**MASTER:** "Estableciendo parámetros de control"	**PUERTA:** "Puerta"		**EMPEZAR:** "Empezar" "Inicio"	**PARO TOTAL:** “Suspendiendo sistemas” “Apagando sistemas” “Suspensión de sistemas” “Desactivación” “Paro”
		VENTANA: "Ventana"			
		LUCES: "Luces"			
		PERSIANAS: "Persianas"		**LISTO:** "Listo" "Terminado" "Alto"	
		ALTURA DE CAMA: "Altura de cama" "Elevación de cama"			
		ESPALDAR DE CAMA: "Espalda" "Espaldar" "Inclinación de cama"			

Elaborado por: Fabián Quishpe& Diego Ulloa

CONCLUSIONES

- Se ha estudiado las diferentes sentencias y comandos para obtener los tres diferentes tipos de señales que el sensor Kinect es capaz de determinar del cuerpo humano, como lo son las imágenes que se hallan frente al sensor a través de una cámara RGB, la ubicación de las personas en el espacio otorgándole las coordenadas de su posición respecto al Kinect utilizando una cámara infrarroja y finalmente la captura de audio para el reconocimiento de voz empleando el arreglo de micrófonos con los que el sensor viene incorporado, todo esto dentro de la plataforma de Microsoft Visual C#.

- Se ha desarrollado un software que permite registrar el movimiento y las órdenes de voz del usuario para el cual se ha realizado el proyecto, en este caso, del usuario se registran sus palabras como órdenes o comandos que da el acceso a manipular cada una de las variables de las salidas del sistema, y mediante la captura de las coordenadas del brazo derecho del usuario, se puede modificar el valor del estado de las salidas, y así la persona que emplee el proyecto es capaz de interactuar de una manera confortable y sencilla con la plataforma.

- Se implementó un sistema que captura para los valores correspondientes a cada una de las salidas que se obtiene del software desarrollado en Microsoft Visual C#, y enviarlos a cada uno de sus correspondientes actuadores, en este parte del proyecto se realizaron consideraciones pensando primero en los actuadores que se controlaron y ajustar las variables de salida a los requerimientos de los mismos, realizando esta operación dentro de la tarjeta de control Arduino.

- Una vez finalizado la implementación del proyecto se realizaron pruebas para verificar el correcto funcionamiento y calibración de los diferentes dispositivos que componen el mismo, para buscar posibles deficiencias o limitaciones para evitar funcionamiento erróneo que pueda causarle contratiempos al usuario; en este punto se ha logrado evitar problemas por mal uso o interpretación errónea de órdenes de voz que el sensor pueda llegar

interpretar, además se consideró problemas por corte de energía en lo que el único inconveniente que pueda llegar a molestar de cierto modo al usuario seria al momento del restablecimiento de la energía, ya que la puerta podría llegar a abrirse de manera abrupta y golpear a alguien que esté ubicado cerca de la misma, finalmente, como inconveniente importante estaría la pérdida de conexión entre el sistema del ordenador y la tarjeta de control, ya sea por desconexión del cable, o malfuncionamiento del ordenador que produzca que este se apague, el sistema dejará los actuadores en el último estado en el que se quedasen.

RECOMENDACIONES

- Para un funcionamiento apropiado y sin accidentes es recomendable mantener conectado el ordenador, y las fuentes externas del que alimentan los servomotores a una fuente de alimentación ininterrumpida (UPS), para evitar accidentes o tener los actuadores en un estado incómodo para el usuario.

- Se debe tener cuidado del uso de los comandos de voz, para activarlos solo cuando se requiera y desactivarlos una vez se haya culminado su uso, para evitar que el sistema interprete el órdenes de voz, o el movimiento del cuerpo del usuario cuando éste no lo desee, puesto que esto puede llegar a propinar inconvenientes y molestias. Para solucionar este problema se podría utilizar el sensor Kinect versión dos que posee un micrófono con tecnología de eliminación de ruido y capaz de reconocer la voz natural incluso en habitaciones llenas de personas.

- Es recomendable que en el ordenador donde se instale el sistema se halle instalado la versión exprés de Microsoft Visual C#, ya que la suite completa de Microsoft Visual Studio resulta demasiado pesada la ejecución de sistema, además es recomendable que como hardware, éste ordenador posea unos 4GB de memoria en RAM, y que su procesador haya sido fabricado como mínimo a medidos de 2011 lo que concuerda con los procesadores Core de segunda generación de Intel, ya que ordenadores con procesadores más antiguos no han respondido de manera apropiada al ejecutar el sistema.

- Procurar que el usuario no manipule directamente o trate de forzar los actuadores mecánicos a excepción de los de la cama automática, cuando estos se hallen encendidos, ya que estos no son lo suficientemente fuertes para vencer a una persona, y en algunos casos puede llegar a dañarlos.

- Se debe tener precaución con la ubicación del sensor Kinect, el cual debe enfocar de frente al usuario, de preferencia enfocar el cuerpo completamente, y además, estar posicionado a una distancia que se encuentre en el rango de operación del sensor, es decir entre 1.80 y 3.80 metros.

LISTA DE REFERENCIAS

Abrego, M. (13 de julio de 2013). *Kinect 2.0*. Recuperado el 4 de enero de 2014, de http://malenyabrego.wordpress.com/2013/07/13/kinect-2-0/

Arduino Co. (19 de julio de 2012). *Arduino Mega*. Recuperado el 25 de mayo de 2013, de Arduino: http://arduino.cc/es/Main/ArduinoBoardMega

Arduino Co. (s.f.). *Arduino Reference*. Recuperado el 12 de mayo de 2013, de Arduino: http://arduino.cc/es/Reference/HomePage

BotScience. (5 de junio de 2012). *Historia de Arduino y su nacimiento*. Recuperado el 23 de noviembre de 2013, de http://botscience.wordpress.com/2012/06/05/historia-de-arduino-y-su-nacimiento/

Carvajal, M. (6 de agosto de 2012). *Vicepresidencia entrega vivienda equipada a niño con discapacidad*. Recuperado el 20 de agosto de 2013, de http://www.radiopublica.ec/ecmov3.php?c=17913

Conadis. (mayo de 2013). *Registro nacional de discapacidades*. Recuperado el 20 de octubre de 2013, de http://www.consejodiscapacidades.gob.ec/wp-content/uploads/downloads/2014/03/conadis_registro_nacional_discapacidades.pdf

Cruz Roja-España. (22 de mayo de 2001). *TIPOS Y GRADOS DE DISCAPACIDAD*. Recuperado el 28 de mayo de 2013, de Cruz Roja: http://www.cruzroja.es/portal/page?_pageid=418,12398047&_dad=portal30&_schema=PORTAL30

Delgado, A. (18 de enero de 2011). *Los otros usos del Kinect*. Recuperado el 20 de octubre de 2013, de consumer.es: http://www.consumer.es/web/es/tecnologia/hardware/2011/01/18/198227.php

Discapacidad online. (5 de mayo de 2011). *Programas accesibles: Software libre facilita uso del ratón a personas con discapacidad*. Recuperado el 20 de agosto de 2013, de http://www.discapacidadonline.com/programas-accesibles-software-libre-discapacidad.html

Gutierrez, G. (29 de diciembre de 2012). *Kinect muestra su potencial en ámbitos como sanidad, turismo y educación*. Recuperado el 9 de septiembre de 2013, de http://www.3djuegos.com/noticia/129731/0/kinect/ii-encuentro-de-inspiracion-y-networking/

Ideas geek. (10 de noviembre de 2010). *Funcionamiento del sensor de movimiento en Kinect*. Recuperado el 26 de octubre de 2013, de http://www.ideasgeek.net/2010/11/10/funcionamiento-del-sensor-de-movimiento-en-kinect/

INEC. (4 de junio de 2013). *Indicadores Sociodemográficos*. Recuperado el 15 de julio de 2013, de http://www.inec.gob.ec/sitio_carto/

INEN. (2001). Retrieved 2013 йил 16-octubre from law.resource.org: https://law.resource.org/pub/ec/ibr/ec.nte.2293.2001.pdf

INEN. (20 de enero de 2013). *Accesibilidad a las personas con discapacidad y movilidad reducida al medio físico. Área higiénico sanitaria.* Recuperado el 18 de octubre de 2013, de https://law.resource.org/pub/ec/ibr/ec.nte.2293.2001.pdf

Instructables. (24 de octubre de 2013). *Kinect controla Arduino Servos cableadas utilizando Visual Basic 2010*. Recuperado el 7 de febrero de 2014, de http://www.instructables.com/id/Kinect-controls-Arduino-wired-Servos-using-Visual-/

Knight, M. (6 de septiembre de 2012). *Cómo el Kinect inspiró una era de innovaciones científicas*. Recuperado el 20 de noviembre de 2013, de CNN México: http://mexico.cnn.com/tecnologia/2012/09/06/como-el-kinect-inspiro-una-era-de-innovaciones-en-todos-los-campos

Lanacion.com. (10 de marzo de 2010). *Las expresiones corporales desafían a los viejos joysticks*. Recuperado el 17 de noviembre de 2013, de http://www.lanacion.com.ar/1241874-las-expresiones-corporales-desafian-a-los-viejos-joysticks

Logros 360. (7 de agosto de 2010). *Características de KINECT: así es por dentro*. Recuperado el 17 de mayo de 2013, de logros 360: http://www.logros360.com/foro/general/caracteristicas-kinect-asi-por-dentro-t75690.html

Manely. (8 de mayo de 2013). *Kincect*. Recuperado el 20 de agosto de 2013, de http://malenyabrego.wordpress.com/category/kinect/

Microsoft. (11 de julio de 2013). *El Efecto Kinect Cómo el mundo está utilizando Kinect*. Recuperado el 22 de agosto de 2013, de xbox.com: http://www.xbox.com/es-ES/Kinect/Kinect-Effect

Microsoft. (31 de agosto de 2013). *Qué es Kinect?* Recuperado el 22 de septiembre de 2013, de xbox.com: http://www.xbox.com/es-ES/Kinect/GetStarted

Microsoft Co. (2012). *Kinect Sensor*. Recuperado el 29 de abril de 2013, de http://msdn.microsoft.com/en-us/library/hh438998.aspx

Microsoft Developer Network. (22 de agosto de 2013). *Introducción a WPF*. Recuperado el 16 de septiembre de 2013, de http://msdn.microsoft.com/es-es/library/aa970268.aspx

Ministerio de Inclusión Económica y Social. (5 de julio de 2013). *Propuesta de atención integral para personas con discapacidad.* Recuperado el 22 de octubre de 2013, de http://www.inclusion.gob.ec/wp-content/uploads/downloads/2013/11/Modelo-de-Atención-de-Discapacidades.pdf

Miranda, D. (7 de noviembre de 2010). *Los secretos que guarda kinect en sus entrañas*. Recuperado el 18 de octubre de 2013, de http://hamemir.blogspot.com/2010_11_07_archive.html

Mundo Elevación. (2014). *Sillas de Ruedas Motorizadas Equipos para la Movilidad*. Recuperado el 08 de marzo de 2014, de http://www.mundoelevacion.com/catalogo/equipos-para-la-movilidad/sillas-de-ruedas-motorizadas/goya/fotografias/

Nuevo León Unido. (27 de marzo de 2009). *Qué es la discapacidad*. Recuperado el 12 de agosto de 2013, de http://www.nl.gob.mx/?P=info_discapacidad

Nuño, J. (2012). *Reconocimiento de objetos mediante sensor 3D Kinect.* Madrid, España: Universidad Carlos III de Madrid.

Pastor, J. (29 de noviembre de 2012). *Inicio » Actualidad »Innovación »Noticias Éxito del II Encuentro de Inspiración y Networking con Kinect*. Recuperado el 4 de diciembre de 2013, de muycomputer: http://www.muycomputer.com/2012/11/29/exito-del-ii-encuentro-de-inspiracion-y-networking-con-kinect

Pensamientos computables. (26 de noviembre de 2010). *Kinect: Cómo funciona su micrófono multiarray*. Recuperado el 20 de noviembre de 2013, de http://www.pensamientoscomputables.com/entrada/Kinect/microfono/multiarray/como-funciona/xbox-360

Rashid, M. (2004). *Electrónica de Potencia* (Tercera ed.). México: Pearson Educación.

Rea, S. (2012). *Control del movimiento del manipulador de cinco grados de libertad CRS-A255 por esqueletización.* Sangolquí, Ecuador: Escuela Politécnica del Ejército.

Sharp, J. (2010). *Microsoft Visual C# 2010 Step by Step.* Redmond, Washington, Estados Unidos: Microsoft Press.

Vivenciasbfm. (10 de mayo de 2009). Recuperado el 13 de agosto de 2013, de http://vivenciasbfm.wordpress.com/2009/05/10/tipos-de-discapacidad/

Vivenciasbfm. (10 de mayo de 2009). *Tipos de discapacidad*. Recuperado el 13 de agosto de 2013, de http://vivenciasbfm.wordpress.com/2009/05/10/tipos-de-discapacidad/

xataka windows. (7 de octubre de 2013). *La evolución de Kinect y la importancia real de Microsoft Research*. Recuperado el 26 de noviembre de 2013, de http://www.xatakawindows.com/xbox/la-evolucion-de-kinect-y-la-importancia-de-microsoft-research

xataka windows. (27 de diciembre de 2013). *La NASA utiliza Kinect 2 para controlar un brazo robótico*. Recuperado el 12 de enero de 2014, de http://www.xatakawindows.com/xbox/la-nasa-utiliza-kinect-2-para-controlar-un-brazo-robotico

Anexo 1. Servomotor HK 15338.

HK15338 Giant MG Digital Servo

Spec.

Torque: **19.80kg @ 4.8v, 25kg @ 6v**

Weight: **175g**

Speed: **0.23 / 60deg @ 4.8v, 0.20 / 60 deg @ 6v**

Voltage: **4.8v~6v**

Plug: **JR style**

Features:

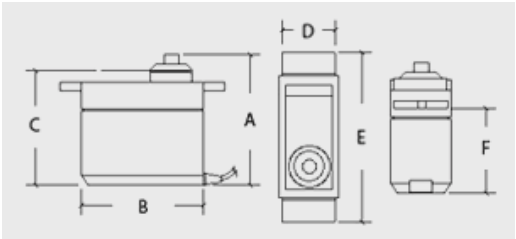
Metal Gears

Dual Ball-race

Water Resistant O-Rings

Ideal For 1/5 Scale Car

Weight (g)	175
Torque (kg)	25
Speed (Sec/60deg)	0.2
A(mm)	66
B(mm)	66
C(mm)	60
D(mm)	30
E(mm)	83
F(mm)	42

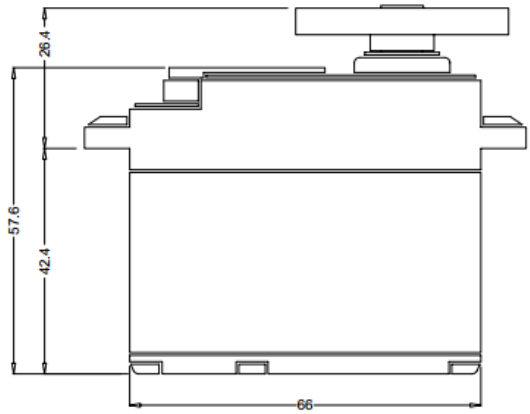


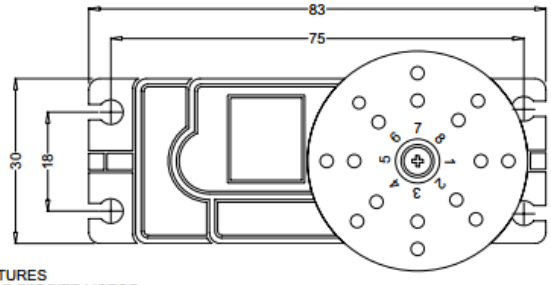
Anexo 2 . Servomotor HS-805BB.

ANNOUNCED SPECIFICATION OF
HS-805BB+ MEGA 1/4 SCALE SERVO

1.TECHNICAL VALUES

CONTROL SYSTEM	: +PULSE WIDTH CONTROL 1500usec NEUTRAL	
OPERATING VOLTAGE RANGE	: 4.8V TO 6.0V	
OPERATING TEMPERATURE RANGE	: -20 TO +60°C	
TEAT VOLTAGE	: AT 4.8V	AT 6.0V
OPERATING SPEED	: 0.19sec/60° AT NO LOAD	0.14sec/60° AT NO LOAD
STALL TORQUE	: 19.8kg.cm(274.96oz.in)	24.7(343.01oz.in)
OPERATING ANGLE	: 45°/ONE SIDE PULSE TRAVELING 400usec	
DIRECTION	: CLOCK WISE/PULSE TRAVELING 1500 TO 1900usec	
CURRENT DRAIN	: 8mA/IDLE AND 700mA/NO LOAD RUNNING	
DEAD BAND WIDTH	: 8usec	
CONNECTOR WIRE LENGTH	: 300mm(11.81in)	
DIMENSIONS	: 66x30x57.6mm(2.59x1.18x1.26in)	
WEIGHT	: 152g(5.36oz)	





2.FEATURES
3-POLE FERRITE MOTOR
LONG LIFE POTENTIOMETER
DUAL BALL BEARING
INDIRECT POTENTIOMETER DRIVE
WATER & DUST TIGHT

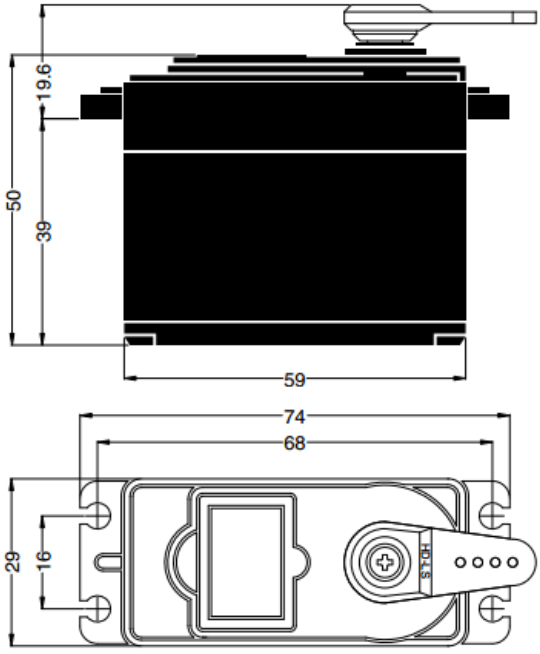
3.APPLICATIONS
GIANT SCALE AIRCRAFT
1/4 SCALE CARS
LARGE SCALE BOATS

Anexo 3. Servomotor HS755-HB.

PREPARED BY JUN HEE, LEE
UPDATE:FEB 20, 2003

GENERAL SPECIFICATION OF HS-755HB QUARTER SCALE SERVO

1.TECHNICAL VALUE		
CONTROL SYSTEM	:+PULSE WIDTH CONTROL 1500usec NEUTRAL	
OPERATING VOLTAGE RANGE	:4.8V TO 6.0V	
OPERATING TEMPERATURE RANGE	:-20℃ TO +60℃(-46℉ TO +86℉)	
TEST VOLTAGE	:AT 4.8V	AT 6.0V
OPERATING SPEED	:0.28sec/60°IAT NO LOAD	0.23sec/60°INO LOAD
STALL TORQUE	:11kg.cm(152.75oz.in)	13.2kg.cm(183.31oz.in)
STANDING TORQUE	:8.8kg.cm(122.20oz.in)/5°IHOLD OUT	10.5kg.cm(145.81oz.in)/5°IHOLD OUT
IDLE CURRENT	:8mA AT STOPPED	8mA AT STOPPED
RUNNING CURRENT	:230mA/60°IAT NO LOAD RUNNING	250mA/60°IAT NO LOAD RUNNING
STALL CURRENT	:1500mA	1800mA
DEAD BAND WIDTH	:5usec	5usec
OPERATING TRAVEL	:40°IONE SIDE PULSE TRAVELING 400usec	
DIRECTION	:CLOCK WISE/PULSE TRAVELING 1500 TO 1900usec	
MOTOR TYPE	:CORED/METAL BRUSH	
POTENTIOMETER TYPE	:6 SLIDER/INDIRECT DRIVE	
AMPLIFIER TYPE	:ANALOG AMPLIFIER	
DIMENSIONS	:59x29x50mm(2.32x1.14x1.96in)	
WEIGHT	:110g(3.88oz)	
BALL BEARING	:DUAL/MR106	
GEAR MATERIAL	:HEAVY DUTY RESIN	
HORN GEAR SPLINE	:24 SEGMENTS/Ø5.76	
SPLINED HORNS	:HEAVY DUTY/HD-IS, HD-O, HD-LS, HD-IL, HD-LL	
CONNECTOR WIRE LENGTH	:300mm(11.81in)	
CONNECTOR WIRE STRAND COUNTER	:60EA	
CONNECTOR WIRE GAUGE	:22AWG	

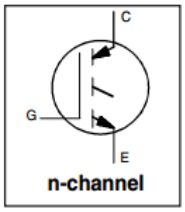


Anexo 4. IGBT IRG4PC50WPbF.

INSULATED GATE BIPOLAR TRANSISTOR

Features

- Designed expressly for Switch-Mode Power Supply and PFC (power factor correction) applications
- Industry-benchmark switching losses improve efficiency of all power supply topologies
- 50% reduction of Eoff parameter
- Low IGBT conduction losses
- Latest-generation IGBT design and construction offers tighter parameters distribution, exceptional reliability
- Lead-Free



n-channel

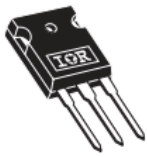
V_{CES} = 600V

$V_{CE(on)\ max.}$ = 2.30V

@V_{GE} = 15V, I_C = 27A

Benefits

- Lower switching losses allow more cost-effective operation than power MOSFETs up to 150 kHz ("hard switched" mode)
- Of particular benefit to single-ended converters and boost PFC topologies 150W and higher
- Low conduction losses and minimal minority-carrier recombination make these an excellent option for resonant mode switching as well (up to >300 kHz)



TO-247AC

Absolute Maximum Ratings

	Parameter	Max.	Units
V_{CES}	Collector-to-Emitter Breakdown Voltage	600	V
I_C @ T_C = 25°C	Continuous Collector Current	55	A
I_C @ T_C = 100°C	Continuous Collector Current	27	
I_{CM}	Pulsed Collector Current ①	220	
I_{LM}	Clamped Inductive Load Current ②	220	
V_{GE}	Gate-to-Emitter Voltage	± 20	V
E_{ARV}	Reverse Voltage Avalanche Energy ③	170	mJ
P_D @ T_C = 25°C	Maximum Power Dissipation	200	W
P_D @ T_C = 100°C	Maximum Power Dissipation	78	
T_J T_{STG}	Operating Junction and Storage Temperature Range	-55 to + 150	°C
	Soldering Temperature, for 10 seconds	300 (0.063 in. (1.6mm from case)	
	Mounting torque, 6-32 or M3 screw.	10 lbf•in (1.1N•m)	

Electrical Characteristics @ T_J = 25°C (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions
$V_{(BR)CES}$	Collector-to-Emitter Breakdown Voltage	600	—	—	V	V_{GE} = 0V, I_C = 250µA
$V_{(BR)CES}$	Emitter-to-Collector Breakdown Voltage ④	18	—	—	V	V_{GE} = 0V, I_C = 1.0A
$\Delta V_{(BR)CES}/\Delta T_J$	Temperature Coeff. of Breakdown Voltage	—	0.41	—	V/°C	V_{GE} = 0V, I_C = 5.0mA
$V_{CE(ON)}$	Collector-to-Emitter Saturation Voltage	—	1.93	2.3	V	I_C = 27A — V_{GE} = 15V
		—	2.25	—		I_C = 55A — See Fig.2, 5
		—	1.71	—		I_C = 27A , T_J = 150°C
$V_{GE(th)}$	Gate Threshold Voltage	3.0	—	6.0		V_{CE} = V_{GE}, I_C = 250µA
$\Delta V_{GE(th)}/\Delta T_J$	Temperature Coeff. of Threshold Voltage	—	-11	—	mV/°C	V_{CE} = V_{GE}, I_C = 1.0mA
g_{fe}	Forward Transconductance ⑤	27	41	—	S	V_{CE} = 100 V, I_C = 27A
I_{CES}	Zero Gate Voltage Collector Current	—	—	250	µA	V_{GE} = 0V, V_{CE} = 600V
		—	—	2.0		V_{GE} = 0V, V_{CE} = 10V, T_J = 25°C
		—	—	5000		V_{GE} = 0V, V_{CE} = 600V, T_J = 150°C
I_{GES}	Gate-to-Emitter Leakage Current	—	—	±100	nA	V_{GE} = ±20V

Switching Characteristics @ T_J = 25°C (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions
Q_g	Total Gate Charge (turn-on)	—	180	270	nC	I_C = 27A
Q_{ge}	Gate - Emitter Charge (turn-on)	—	24	36		V_{CC} = 400V — See Fig.8
Q_{gc}	Gate - Collector Charge (turn-on)	—	63	95		V_{GE} = 15V
$t_{d(on)}$	Turn-On Delay Time	—	46	—	ns	
t_r	Rise Time	—	33	—		T_J = 25°C
$t_{d(off)}$	Turn-Off Delay Time	—	120	180		I_C = 27A, V_{CC} = 480V
t_f	Fall Time	—	57	86		V_{GE} = 15V, R_G = 5.0Ω
E_{on}	Turn-On Switching Loss	—	0.08	—	mJ	Energy losses include "tail"
E_{off}	Turn-Off Switching Loss	—	0.32	—		See Fig. 9, 10, 14
E_{ts}	Total Switching Loss	—	0.40	0.5		
$t_{d(on)}$	Turn-On Delay Time	—	31	—	ns	T_J = 150°C,
t_r	Rise Time	—	43	—		I_C = 27A, V_{CC} = 480V
$t_{d(off)}$	Turn-Off Delay Time	—	210	—		V_{GE} = 15V, R_G = 5.0Ω
t_f	Fall Time	—	62	—		Energy losses include "tail"
E_{ts}	Total Switching Loss	—	1.14	—	mJ	See Fig. 10,11, 14
L_E	Internal Emitter Inductance	—	13	—	nH	Measured 5mm from package
C_{ies}	Input Capacitance	—	3700	—	pF	V_{GE} = 0V
C_{oes}	Output Capacitance	—	260	—		V_{CC} = 30V — See Fig. 7
C_{res}	Reverse Transfer Capacitance	—	68	—		f = 1.0MHz

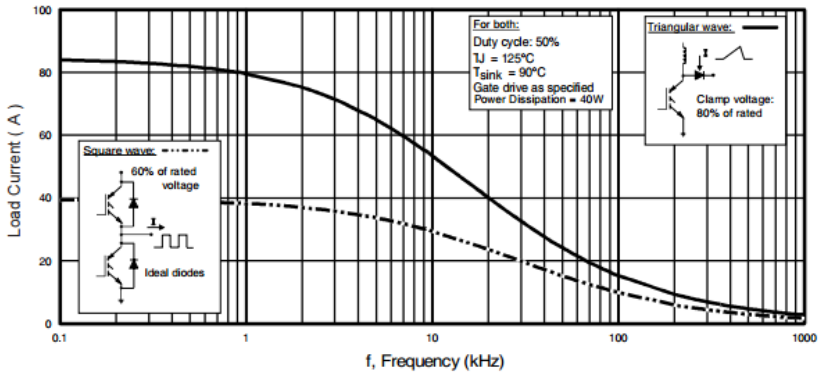


Fig. 1 - Typical Load Current vs. Frequency
(Load Current = I_{RMS} of fundamental)

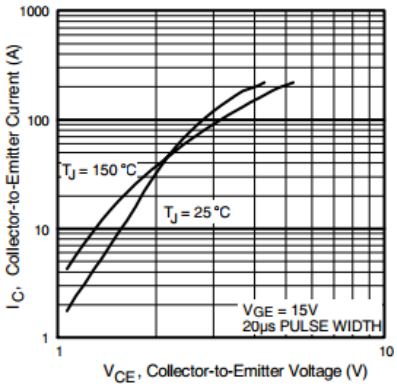


Fig. 2 - Typical Output Characteristics

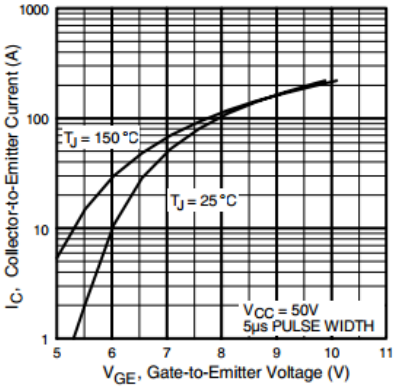


Fig. 3 - Typical Transfer Characteristics

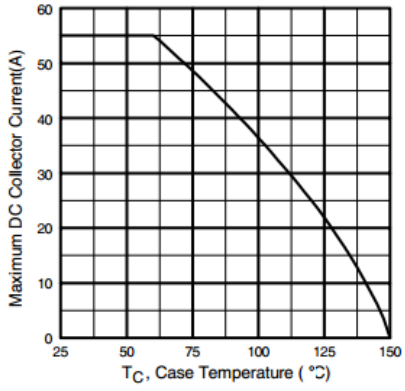


Fig. 4 - Maximum Collector Current vs. Case Temperature

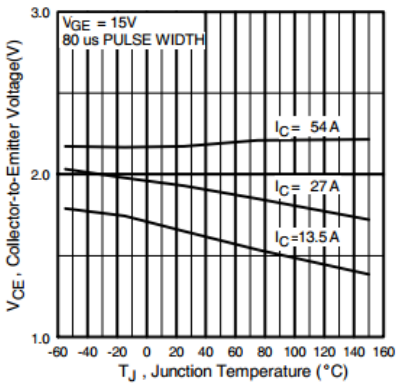


Fig. 5 - Typical Collector-to-Emitter Voltage vs. Junction Temperature

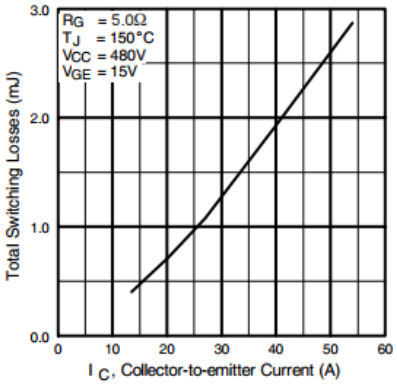


Fig. 11 - Typical Switching Losses vs. Collector-to-Emitter Current

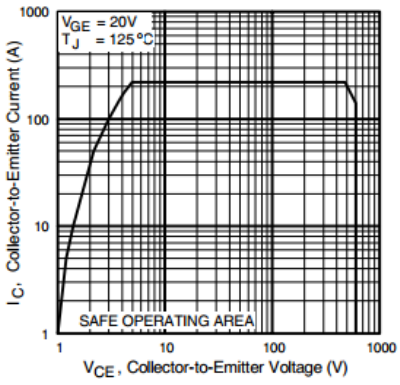


Fig. 12 - Turn-Off SOA

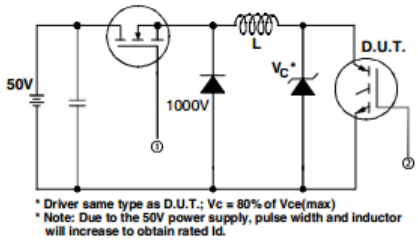


Fig. 13a - Clamped Inductive Load Test Circuit

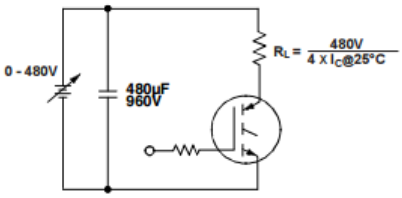


Fig. 13b - Pulsed Collector Current Test Circuit

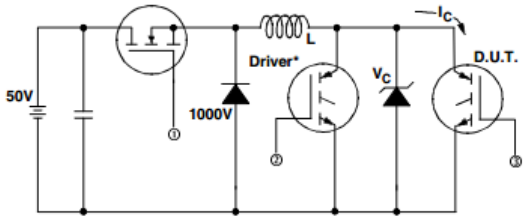


Fig. 14a - Switching Loss Test Circuit