



**UNIVERSIDAD MICHOACANA DE SAN
NICOLÁS DE HIDALGO**

**DIVISIÓN DE ESTUDIOS DE POSGRADO
DE LA FACULTAD DE INGENIERÍA MECÁNICA**

MAESTRÍA EN CIENCIAS EN INGENIERÍA MECÁNICA

**DESARROLLO MECATRÓNICO DE UNA MANO
ARTIFICIAL CON PROPÓSITOS BIOMÉDICOS: SISTEMA
DE CONTROL**

TESIS

QUE PARA OBTENER EL TÍTULO DE:

MAESTRO EN CIENCIAS EN INGENIERÍA MECÁNICA

PRESENTA:

ING. CARLOS OROS MARTÍNEZ

ASESOR:

DR. IGNACIO JUÁREZ CAMPOS

COASESOR:

DR. MARCELO LÓPEZ PARRA

MORELIA, MICHOACÁN, FEBRERO 2009



AGRADECIMIENTOS

Quiero agradecer a mis padres por enseñarme el valor de la responsabilidad y constancia que es lo que me impulsó a cumplir una meta más en mi vida y que sin su apoyo y comprensión no hubiera llegado hasta aquí.

A mis amigos Arceo, Alejandro, Hugo y compañeros que me brindaron su amistad en los momentos difíciles y que intervinieron en mi crecimiento personal y profesional durante el desarrollo de este proyecto

Por último quiero agradecer al Dr. Ignacio Juárez Campos por confiar en que podía desarrollar este proyecto, y sobre todo al Dr. Marcelo López Parra, coasesor del presente trabajo que sin su valiosa colaboración y tiempo empleado en el proyecto, no se hubiera seguido la línea de trabajo que se utilizó para desarrollar del proyecto.

ÍNDICE

RESUMEN.....1

CAPÍTULO I. INTRODUCCIÓN

1.1 Justificación.....2

1.2 Establecimiento de hipótesis.....3

1.3 Objetivo general.....4

1.4 Objetivos específicos.....4

1.5 Estructura de la tesis.....5

1.6 Antecedentes

 1.6.1 Historia del reconocimiento de voz.....7

 1.6.1.1 Los inicios: años 50's.....7

 1.6.1.2 Los fundamentos: A mediados de los 60's.....7

 1.6.1.3 Las primeras soluciones: A principios de los 70's.....8

 1.6.1.4 Reconocimiento del habla continua: años 80's.....8

 1.6.1.5 Empieza el negocio: años 90's.....9

 1.6.1.6 Una realidad: año 2000.....9

 1.6.1.7 En la actualidad (año 2008).....9

1.7 Estado del arte del reconocimiento de voz.....10

 1.7.1 Aplicaciones del reconocimiento de voz.....15

1.8 Conclusiones del capítulo I.....16

CAPÍTULO II. DISEÑO DEL SOFTWARE

2.1 Información del reconocimiento de voz y del dispositivo electrónico.....	17
2.1.1 Procesos del reconocimiento de voz.....	17
2.1.2 Tipos de reconocimiento de voz.....	18
2.1.3 Los controladores digitales de señal (DSC o dsPIC's).....	18
2.1.3.1 Procesamiento digital de señales (DSP).....	19
2.1.3.2 Microcontroladores y DSP.....	20
2.1.3.3 El concepto DSC.....	20
2.1.3.4 Características generales de los dsPIC30F.....	21
2.1.3.5 Los modelos de la familia dsPIC30F.....	21
2.1.3.6 Encapsulado y diagramas de conexiones.....	22
2.1.3.7 Aplicaciones recomendadas.....	24
2.1.4 Selección del controlador digital de señal.....	26
2.2 Metodología para el reconocimiento de voz.....	27
2.2.1 Etapa I. Creación de la biblioteca para el reconocimiento de voz.....	28
2.2.2 Etapa II. Programa principal para el reconocimiento de voz.....	35
2.2.2.1 Construcción del proyecto principal.....	35
2.2.2.2 Programa para el dsPIC30F.....	39
2.2.3 Etapa III. Programa para el microcontrolador PIC.....	40
2.3 Programación del microcontrolador y el DSC.....	40

2.3.1 Programación del dispositivo PIC16F877A.....	41
2.3.1.1 Instalación del WinPic800.....	42
2.3.2 Programación del dsPIC30F.....	44
2.3.2.1 Instrucciones para conectar el ICD2.....	44
2.4 Conclusiones del capítulo II.....	48

CAPÍTULO III: DISEÑO DEL SISTEMA ELECTRÓNICO

3.1 Componente para los circuitos electrónicos.....	49
3.1.1 Componentes del hardware para el dsPIC30F.....	49
3.1.2 Componentes del hardware para el reconocimiento de voz.....	49
3.1.3 Componentes del hardware para el control de la mano artificial.....	51
3.2 Etapa I. Módulo <i>plug-in</i> para el dsPIC30F6014A.....	52
3.3 Etapa II. Circuito para el reconocimiento de voz.....	53
3.4 Etapa III. Circuito para el control de la mano artificial.....	60
3.5 Conclusiones del capítulo III.....	63

CAPÍTULO IV: EXPERIMENTOS Y RESULTADOS

4.1 Tamaño de muestra para el reconocimiento de voz.....	64
4.1.1 Nivel de precisión deseado.....	64
4.1.2 Limite de confianza.....	64
4.1.3 Selección de la muestra.....	65
4.1.4 Formula para n en el muestreo para proporciones.....	65

4.1.5 Estimación del tamaño de muestra para el reconocimiento de voz.....	66
4.2 Características de los micrófonos.....	67
4.3 Metodología para el muestreo.....	68
4.4 Resultados.....	71
4.5 Análisis de Resultados.....	78
CONCLUSIONES.....	79
BIBLIOGRAFÍA.....	82
ANEXO I. CONCEPTOS.....	85
ANEXO II. PROGRAMA PRINCIPAL DE LA BIBLIOTECA WordLibSR.h....	90
ANEXO III. PROGRAMA PARA EL RECONOCIMIENTO DE VOZ.....	91
ANEXO IV. PROGRAMA PARA EL MOVIMIENTO DE LA PRÓTESIS UTILIZANDO EL PIC16F877A.....	98

ÍNDICE DE FIGURAS

1.7.1 Kit manos libres CK-3000.....	11
1.7.2 Teléfono con reconocimiento de voz.....	12
1.7.3 Tarjeta <i>Beepcard</i> con reconocimiento de voz.....	13
1.7.4 Reconocimiento de voz aplicado en aviones.....	13
1.7.5 TV JVC con reconocimiento de aplausos y gestos.....	14
1.7.6 Robot Hitachi.....	15
2.1.1 Componentes de una aplicación.....	18
2.1.2 El micrófono transforma en una señal analógica.....	19
2.1.3 Modelos de encapsulados de la familia dsPIC30F.....	22
2.1.4 Campos en que se descomponen la nomenclatura del dsPIC30F.....	23
2.2.1 Descripción del reconocimiento de voz.....	28
2.2.2 Proceso para creación de la biblioteca.....	29
2.2.3 Entorno de <i>Word Library Builder</i>	30
2.2.4 Lista de la biblioteca principal.....	30
2.2.5 Selección de la <i>Keyword</i>	31
2.2.6 Perfiles de la biblioteca de ruido.....	32
2.2.7 Adiestramiento completo.....	32
2.2.8 Asignación de memoria.....	33
2.2.9 Finalizar datos de salida.....	34

2.2.10 Selección del dispositivo.....	35
2.2.11 Selección del lenguaje.....	36
2.2.12 Nombre del proyecto.....	37
2.2.13 Archivos agregados al proyecto.....	37
2.2.14 Finalizar proyecto.....	38
2.2.15 Archivos totales para el proyecto.....	39
2.3.1 Dispositivo para grabar el programa en el PIC.....	41
2.3.2 Entorno del WinPic800.....	41
2.3.3 Icono para programar el PIC.....	42
2.3.4 Configuración del Hardware.....	42
2.3.5 Configuración del software.....	43
2.3.6 Opciones para programar y compilar.....	43
2.3.7 Nombre del programa ejecutable.....	43
2.3.8 MPLAB ICD-2 graba y depura en circuitos los dsPIC.....	44
2.3.9 Configuraciones para construir el proyecto.....	45
2.3.10 Construcción del proyecto.....	46
2.3.11 Configuración para programar.....	47
3.1 DSPIC TQFP80 Elementos.....	52
3.2 DSPIC TQFP80 Pistas.....	52
3.3 DSPIC TQFP80 Integración.....	52
3.4 DSPIC PDIP28 Elementos.....	53

3.5 DSPIC PDIP28 Pistas.....	53
3.6 DSPIC PDIP28 Integración.....	53
3.7 Circuito de voz TQFP80 elementos.....	54
3.8 Circuito de voz TQFP80 pistas.....	54
3.9 Circuito de voz TQFP80 integración.....	55
3.10 Circuito de voz TQFP80 Puentes.....	55
3.11 Circuito de voz TQFP80 esquemático.....	56
3.12 Circuito de voz PDIP28 elementos.....	57
3.13 Circuito de voz PDIP28 pistas.....	57
3.14 Circuito de voz PDIP28 integración.....	58
3.15 Circuito de voz PDIP28 Puentes.....	58
3.16 Circuito de voz PDIP28 esquemático.....	59
3.17 Control mano artificial elementos.....	60
3.18 Control mano artificial pistas.....	60
3.19 Control mano artificial integración.....	61
3.20 Control mano artificial Puentes.....	61
3.21 Control mano artificial esquemático.....	62

ÍNDICE DE TABLAS

2.1.3.1 Características más importantes de la familia dsPIC30F.....	21
2.1.3.2 Símbolo, dirección y descripción de las patitas del dsPIC30F.....	23
2.1.4.1 Características del dsPIC30F6014A.....	26
2.2.1 Perfil de ruido predefinido.....	31
2.2.2 Archivos generados por la biblioteca.....	33
2.2.3 Código enviado por el dsPIC30F.....	40
3.1 Elementos del circuito para el dsPIC30F.....	49
3.2 Elementos del circuito para reconocimiento de voz.....	50
3.3 Elementos para control de prótesis.....	51
4.1 Probabilidad del nivel de confianza deseada.....	64
4.2 Características del micrófono <i>Logitech</i>	67
4.3 Características del micrófono <i>Plantronics</i>	67
4.4 Características del micrófono <i>Acteck</i>	68
4.5 Perfiles de ruido.....	70
4.6 Resultados obtenidos del muestreo.....	78
4.7 Mejores parámetros para el reconocimiento de voz.....	78

ÍNDICE DE DIAGRAMAS

2.2 Diagrama de flujo global para el reconocimiento de voz.....	27
4.1 Muestras tomadas para el micrófono A.....	69
4.2 Muestras tomadas para el micrófono B.....	69
4.3 Muestras tomadas para el micrófono C.....	70

ÍNDICE DE GRÁFICAS

4.1 Micrófono A - Eficiencia por palabra sin ruido.....	72
4.2 Micrófono A - Eficiencia <i>WordLib_6</i> para MIC 30 sin ruido.....	72
4.3 Micrófono A - Eficiencia por palabra con ruido.....	73
4.4 Micrófono A - Eficiencia <i>WordLib_6</i> para MIC 30 con ruido.....	73
4.5 Micrófono B - Eficiencia de cada palabra sin ruido.....	74
4.6 Micrófono B - Eficiencia <i>WordLib_6</i> para MIC 30 sin ruido.....	74
4.7 Micrófono B - Eficiencia de cada palabra con ruido.....	75
4.8 Micrófono B - Eficiencia <i>WordLib_6</i> para MIC 30 con ruido.....	75
4.9 Micrófono C - Eficiencia de cada palabra sin ruido.....	76
4.10 Micrófono C - Eficiencia <i>WordLib_4</i> para MIC30 sin ruido.....	76
4.11 Micrófono C - Eficiencia de cada palabra con ruido.....	77
4.12 Micrófono C - Eficiencia <i>WordLib_4</i> para MIC 30 con ruido.....	77

RESUMEN

En el presente proyecto se diseña y construye un dispositivo basado en el reconocimiento de voz, el cual está compuesto de un circuito electrónico y su respectivo software. Éstos transmiten las órdenes requeridas para el funcionamiento de la prótesis que será utilizada por personas con amputación de mano derecha.

El circuito electrónico está dividido en dos etapas: la primera etapa comienza cuando el micrófono capta las señales acústicas de la persona (las palabras) y las transmite al decodificador de voz. Una vez procesada éste las envía al dsPIC que analiza el espectro de voz y lo compara con las palabras guardadas previamente en el dispositivo (biblioteca de reconocimiento de voz). Cuando encuentra la palabra que más se asemeja a ésta, manda en código binario el número de la palabra reconocida al microcontrolador que es el encargado de accionar la mano artificial.

La biblioteca de reconocimiento de voz es el apartado en donde se generaron los códigos para las palabras que posteriormente serán comparadas con las señales acústicas de entrada. La biblioteca de reconocimiento de voz puede contener hasta un total de 100 palabras, si así lo requiere el usuario.

La segunda etapa inicia cuando el microcontrolador recibe el código de la palabra que ha sido reconocida. Cuando recibe el código, se ejecuta la rutina específica para cada palabra, mandando una señal para activar los servomotores, por ejemplo para abrir o cerrar la mano artificial.

Los resultados muestran que el reconocimiento de voz es factible, logrando alcanzar una eficiencia del 97% para ambientes sin ruido y del 93% con ruido. Aunque el reconocimiento de voz es discreto, solo requiere de una pausa de medio segundo entre las palabras y de medio segundo para reconocer la palabra y realizar el movimiento respectivo de la mano artificial.

CAPÍTULO I. INTRODUCCIÓN

1.1 JUSTIFICACIÓN

Para el año 2000 existían 1, 795, 300 personas discapacitadas en los Estados Unidos Mexicanos (*INEGI*). De éstas 813, 271 padecían alguna discapacidad motriz. En Michoacán, alrededor de 38, 240 personas han sufrido la amputación de manos y brazos [1].

Las principales causas de una amputación

Los traumatismos constituyen uno de los principales problemas de salud pública, ocupando el tercer lugar como causa de muerte y el segundo como causa de discapacidad [2].

Las principales fuentes de traumatismos son:

- Los accidentes automovilísticos y laborales
- Heridas por armas blancas y de fuego
- Las guerras
- Las enfermedades, por ejemplo, la diabetes.
- Muerte de tejido por envenenamiento, por ejemplo, mordedura de serpientes, etc.

Niveles de Amputación en Brazos y Manos

- A nivel de la muñeca (el que concierne a la investigación presentada)
- A nivel del codo
- A nivel del hombro

El problema consiste en diseñar y construir un dispositivo no invasivo que sea capaz de manipular una prótesis artificial y que será utilizada por pacientes con amputación a nivel del antebrazo. Un dispositivo no invasivo se trata de un producto o instrumento que no penetre parcial o completamente en el interior de algún cuerpo.

El modelo funcional será exclusivamente de la mano derecha, dicha mano se diseñará tomando como base las medidas de la mano derecha de un hombre de 50 años, mexicano de 1.70m de estatura, cuyo muñón se encuentra a 10 cm por debajo del codo.

En la actualidad existen un gran número de prótesis estéticas y funcionales, pero la gran mayoría del tipo funcional proviene del extranjero, de países como Alemania, Estados Unidos de Norteamérica y Japón. Por el hecho de ser importadas, éstas tienen un costo elevado. Este proyecto pretende bajar los costos de fabricación, haciendo de las prótesis funcionales más accesibles a más mexicanos, como obreros o trabajadores manuales que son las personas con mayor probabilidad de perder un miembro y que su salario no permite acceder fácilmente a las prótesis de importación.

Por esta razón actualmente en la Facultad de Ingeniería Mecánica y bajo el nombre del proyecto de investigación: *Desarrollo de Manos Artificiales para Propósitos Biomédicos*, se lleva a cabo el trabajo para producir una prótesis de la mano humana con las características siguientes:

1. Económicamente accesible a la población concerniente.
2. La mayor versatilidad posible equiparable a las funciones de la mano humana. No simplemente una prótesis cosmética.
3. Implementar un control a través de un dispositivo no invasivo.

1.2 ESTABLECIMIENTO DE HIPÓTESIS

En base a las investigaciones de reconocimiento de voz que se han desarrollado en la UNAM con diferentes tarjetas diseñadas para el reconocimiento de voz, así como en el artículo prótesis inteligente de brazo diseñado en la facultad de ingeniería y publicado en la revista gaceta (Número 3, 696, 2004) por la UNAM y a las aportaciones realizadas por el Ing. Ulises Peñuelas Ribas realizadas en la visita que se hizo al centro de diseño Mecánico e Innovación Tecnológica de la UNAM (CDM), con la ayuda del Dr. Marcelo López Parra coasesor del presente proyecto, lo que dio origen a la planificación de la siguiente hipótesis [3].

Es posible controlar una mano artificial a través de comandos por voz, mediante un dispositivo compuesto por un circuito electrónico y un software, quienes analizarán los espectros de voz y proveerán las rutinas necesarias para el funcionamiento de la mano artificial que será utilizada por un paciente con amputación de mano derecha.

1.3 OBJETIVO GENERAL

- Diseñar un sistema de control compuesto de un software y un hardware, analizando los espectros de voz para su programación. El dispositivo estará encargado de convertir las señales sonoras (palabras) en señales digitales mediante las cuales se accionará una prótesis de mano artificial utilizada por pacientes con amputación a nivel del antebrazo.

1.4 OBJETIVOS ESPECÍFICOS

- Analizar el estado del arte en sistemas que convierten señales sonoras en digitales para diferentes propósitos, enfatizando los de acciones mecánicas o eléctricas.
- Confeccionar un software que convierta la señal sonora en forma de palabras a señal digital, que pueda ser interpretada por un circuito electrónico.
- Construir un circuito electrónico que convierta la señal digital en acciones realizadas por motores encargados de accionar una mano artificial.
- Establecer la coordinación entre el software, el circuito electrónico y la mano artificial, de modo que responda a los intereses del paciente con amputación a nivel del antebrazo.

1.5 ESTRUCTURA DE LA TESIS

La estructura de la tesis se presenta dividida en cuatro capítulos, conclusiones y anexos, en los capítulos se abarca desde los antecedentes del reconocimiento de voz, el diseño de las placas para el circuito y los experimentos realizados para conocer la eficiencia de este dispositivo. Los anexos muestran los códigos de los programas.

Capítulo I. Antecedentes

Ya que el reconocimiento de voz no es relativamente nuevo, se inicia este capítulo aprendiendo de los inicios y características de los primeros dispositivos diseñados para el reconocimiento de voz.

Por otro lado, también se mencionan los avances tecnológicos y aplicaciones más recientes, en cuanto a reconocimiento de voz se refiere.

Capítulo II. Diseño del software

En este capítulo se describen las características generales del por qué se eligió un dsPIC para el reconocimiento de voz. También se describen las familias, tipos de encapsulados y las aplicaciones en las cuales se pueden implementar este tipo de dispositivos.

Además, se detalla la creación de la biblioteca para el reconocimiento de voz, así como el desarrollo del programa para el reconocimiento de voz. Por último se desarrolla el programa para el microcontrolador que transmitirá las órdenes para los movimientos de la mano.

Capítulo III. Diseño del circuito electrónico

Describe la metodología y los componentes necesarios para la construcción del dispositivo, tanto para el reconocimiento de voz como para la tarjeta que transmitirá las órdenes para los movimientos de la prótesis. Por último se muestran los circuitos que se utilizan para la construcción de los dispositivos.

Capítulo IV. Experimentación y Resultados

Este capítulo estudia los diferentes niveles de ruido, así como la precisión y la sensibilidad que se obtiene utilizando diferentes micrófonos con diferentes características.

Para esto se realizó un muestreo con los parámetros antes mencionados: para conocer y seleccionar los mejores niveles de eficiencia en cada micrófono y para obtener una mejor precisión del reconocimiento de voz. Los resultados obtenidos se muestran en gráficas para cada uno de los micrófonos.

Conclusiones

En esta sección se resumen las conclusiones más relevantes extraídas de la presente tesis.

Anexo I

Describe los conceptos básicos para tener una mejor comprensión del tema.

Anexo II

En éste se anexa el código generado por el programa *Word Library Bulder*, el código define las características utilizadas para crear el programa, también define el total de las palabras y el orden de éstas.

Anexo III

Además de mostrar los códigos realizados para integrar la biblioteca, las configuraciones del dsPIC y el programa principal para el reconocimiento de voz, se describen algunas funciones de los comandos utilizados.

Anexo IV

En este anexo se muestra el código que se realizó para dar el movimiento a los servomotores una vez que reciba el código mandado por el dsPIC.

1.6 ANTECEDENTES

1.6.1 Historia del reconocimiento de voz

La historia del reconocimiento de voz inicia en 1950 con múltiples paradigmas de trabajos y resultados. Inclusive muchas de las técnicas utilizadas con éxito debieron esperar más de 10 años para pasar de la teoría a la práctica o a laboratorios [4].

La historia de los sistemas de reconocimiento de voz se puede resumir con las siguientes etapas:

1.6.1.1 Los inicios: años 50's

Entre 1940 y 1950, los laboratorios Bell crearon la primera máquina basada en plantillas que permitía identificar para un solo hablante, las pronunciaciones de los primeros 10 dígitos del inglés realizadas de forma aislada. Requería un extenso entrenamiento a la voz de la persona, pero una vez logrado, tenía entre 95 y 99% de certeza. Por lo que surgió la esperanza de que el reconocimiento de voz fuera simple y directo.

1.6.1.2 Los fundamentos: A mediados de los 60's

La mayoría de los investigadores reconoce que era un proceso mucho más confuso y sutil de lo que se había anticipado. Intentan reducir los alcances y se centran en sistemas más específicos:

- Dependientes del Locutor
- Flujo discreto de habla (con espacios / pausas entre palabras)
- Vocabulario pequeño (menor o igual a 50 palabras)

Estos sistemas empiezan a incorporar técnicas de normalización del tiempo (minimizar la diferencia en la velocidad del habla). Además, ya no buscaban la exactitud perfecta en el reconocimiento.

Después la compañía privada IBM (*International Business Machines*) trabaja en reconocimiento de voz continuo, pero los resultados no llegan hasta 1970.

1.6.1.3 Las primeras soluciones: A principios de los 70's

Se produce el primer producto de reconocimiento de voz, el VIP100 de *Threshold Technology Inc.* (utilizaba un vocabulario pequeño, dependiente del locutor, y reconocía palabras discretas).

Durante este mismo periodo, el programa DARPA (*Defence Advance Research Agency*), impulsó en Estados Unidos el desarrollo de mejores sistemas de reconocimiento para el habla continuo y vocabularios de tamaño medio y grande, con independencia del locutor.

Los sistemas empiezan a incorporar los siguientes módulos que ayudan a mejorar la calidad de los sistemas [5]:

- Análisis léxico (Conocimiento léxico)
- Análisis sintáctico (Estructura de palabras)
- Análisis semántico (Significado)
- Análisis pragmático (Intención)

En los 70's muchas de las aportaciones de estos proyectos vienen más por la parte de la estructura software de los sistemas basados en el conocimiento que por los avances exclusivos en el reconocimiento de voz [6].

1.6.1.4 Reconocimiento del habla continua: años 80's

Se aplicaron los conceptos de DTM (*Dynamic Time Warping*). Se produce un importante cambio de paradigma de comparación de plantillas hacia el modelado estadístico/probabilístico como un gran avance de aproximación al reconocimiento de voz.

A mitad de los 80's se hizo masiva una técnica que revolucionó el campo de reconocimiento, los modelos ocultos de Márkov o HMM (*Hidden Markov Model*), que obtuvo excelentes resultados en el modelado de señales de voz y virtualmente indispensable hoy en día.

Se reintroduce el uso de redes neuronales artificiales (*ANN-Artificial Neural Networks*), que habían vencido algunos obstáculos de tipo conceptual y de recursos necesarios para su implementación. También se comenzó a experimentar con reconocimiento continuo de vocabularios largos independientes del locutor.

1.6.1.5 Empieza el negocio: años 90's

En los 90's se comenzó a hacer énfasis en interfaces de lenguaje natural, y recuperación de la información en grandes documentos de voz. Continúo la investigación de reconocimiento continuo en vocabularios grandes y a usarse masivamente a través de redes telefónicas. También se utilizó en el estudio de sistemas en condiciones de ruido.

Durante la primera mitad de los 90's se dio la investigación de sistemas híbridos HMM-ANN (*Hidden Markov Model-Artificial Neural Networks*), que también han dado excelentes resultados, siendo la excelencia de los motores de reconocimiento de voz de hoy en día [4].

1.6.1.6 Una realidad: año 2000

Integración de aplicaciones por teléfono y sitios de Internet dedicados a la gestión de reconocimiento de voz (*Voice Web Browsers*).

Aparece el estándar VoiceXML (*Voice Extensible Markup Language*). Es un lenguaje de etiquetas basado en XML que permite describir servicios de voz independiente de la aplicación en la que corran. De esta manera no es necesario conocer detalles específicos de una plataforma para entender el funcionamiento del sistema de diálogo. El lenguaje VoiceXML describe la interacción hombre-máquina a partir de los siguientes elementos [7]:

- Salida de texto-a-voz
- Salida de audio grabado
- Reconocimiento de entrada hablada
- Reconocimiento de tonos DTMF (*Dual Tone Multi-frequency*)
- Grabación de entrada hablada
- Control de flujo de diálogo
- Funciones de telefonía (transferencia de llamada, desconexión, etc.)

1.6.1.7 En la actualidad (año 2008)

En México el desarrollo del reconocimiento de voz no ha sido tan grande como lo ha sido en países como Alemania, Estados Unidos de Norteamérica y Japón, sin embargo, en la facultad de ingeniería en la Universidad Nacional Autónoma de México ha tenido avances importantes. Por ejemplo en proyectos como prótesis inteligente de brazo, donde se intentaron utilizar instrucciones por voz o impulsos mioeléctricos para accionar la prótesis [3].

El Dr. Marcelo López Parra, coasesor del presente proyecto, concertó una vista al centro de diseño mecánico e innovación tecnológica UNAM con el Ing. Ulises Peñuelas Ribas, que trabajo con reconocimientos de voz utilizando varias tarjetas de uso comerciales para el reconocimiento de voz. Gracias a las aportaciones y observaciones que el Ing. Ulises nos hizo, se opto por tomar el reconocimiento de voz como línea de investigación para el desarrollo del proyecto.

1.7 ESTADO DEL ARTE DEL RECONOCIMIENTO DE VOZ

En la actualidad, existen en el mercado un gran número de aplicaciones que realizan el reconocimiento de voz mediante ordenadores, dentro de los cuales se pueden mencionar los siguientes productos:

FreeSpeech 2000

Las principales características de la nueva versión del programa de Philips son su velocidad y su capacidad de aprendizaje. A medida que el usuario se familiariza con el programa, verá cómo los márgenes de error de éste van disminuyendo [8].

Características y propiedades:

- Requiere: Pentium/MMX 166 o equivalente; 48 MB en RAM; 100MB de espacio en disco duro; tarjeta de sonido compatible con *Sound Blaster*; Microsoft Windows 95,98 o NT con SP3.
- Soporte para dictado en 14 idiomas.
- Conversión de voz a texto
- **91%** en precisión
- Dentro de sus desventajas, carece de una tarjeta de consulta rápida e incluye poca información de los comandos.

L & H Voice Xpress Professional, Version 4

Ofrece las características básicas que se podría esperar de un programa de reconocimiento de voz, es posible dictar en cualquier aplicación, crear macros de voz y controlar las aplicaciones con comandos de voz. El punto fuerte del programa son sus comandos intuitivos con lenguaje natural para *Microsoft Word*, Excel y el sencillo procesador de palabras *Voice Xpress*. Pero sus características de corrección son limitadas; por ejemplo, carecen de reproducción de audio.

Características y propiedades [9]:

- Requiere: Pentium II; 48 MB en RAM con Microsoft Windows 95 ó 98, 64 MB con Windows NT; 200 MB de espacio en disco duro.
- 94% en precisión

Dragon Naturally Speaking Preferred 9

Software de reconocimiento por voz que permite redactar documentos, mensajes de correo electrónico e informes tres veces más rápido que con el teclado.

Características y propiedades [10]:

- Requiere: Pentium III/500 o equivalente; 256 MB en RAM; 50MB de espacio en disco duro; tarjeta de sonido de 16 bits compatible con *Creative Labs Sound Blaster*; Microsoft Windows 98 , XP Home y Profesional
- 99% de precisión.
- Soporte para el lenguaje natural en aplicaciones Microsoft Word.

Además, con el fin de resolver problemas específicos o simplemente hacer más fácil la vida de las personas, actualmente se están desarrollando nuevos productos con reconocimiento de voz, algunos ejemplos de éstos son:

Kit Manos Libres Bluetooth Parrot CK-3000 Evolución

El CK-3000 es uno de los sistemas de manos libres para coche de mayor aceptación disponible en el mercado. Gracias a la tecnología de radio *Bluetooth*, el conductor conserva su teléfono en el bolsillo o en el bolso. Basta con girar la llave de contacto, el CK-3000 detecta el teléfono y todas las llamadas telefónicas se efectuarán en modo manos libres. Al llamar, se habla al interlocutor sin dejar el volante. Se llama al interlocutor pronunciando sencillamente su nombre.



Fig. 1.7.1 Kit manos libres CK-3000

Características y propiedades [11]:

- Difusión de la conversación por los altavoces, lo que ofrece una gran calidad sonora.
- Las personas pueden utilizar una instalación única a partir de sus propios teléfonos móviles.
- Disponible en 6 idiomas (inglés, alemán, francés, italiano, español, holandés).

Reconocimiento de voz en teléfonos celulares

La compañía *Nuance Communications* ha desarrollado un sistema de reconocimiento de voz lo suficientemente sofisticado como para dictarle un email completo, decir el nombre del destinatario y enviarlo con la voz [12].



Fig. 1.7.2 Teléfono con reconocimiento de voz

Características y propiedades [13]:

- Los celulares ya no son sólo dispositivos para conversar, “hablar” es lo último en una lista de funciones compuesta por la música, los vídeos, las fotografías y muchas aplicaciones propias de una computadora.

Tarjeta de crédito con reconocimiento de voz

Esta novedosa tarjeta tiene por nombre *Beepcard*, y entre sus características están la de un altavoz de papel delgado con micrófono y un chip de reconocimiento de voz, así que ahora se tendrá que hablar a la tarjeta para que ésta le reconozca, si logra identificar su voz y concuerda con la que ha sido grabada, se envía una señal por medio de la computadora (con un micrófono) o del teléfono, ésta a su vez le dará la autorización a la tarjeta para hacer la transacción que el usuario desea [14].



Fig. 1.7.3 Tarjeta Beepcard con reconocimiento de voz

Software de reconocimiento de voz en los aviones

Hoy por hoy, los pilotos recurren al panel de control para controlar los aviones. Sin embargo esto no continuará igual por mucho tiempo más, pues la Fuerza Aérea de los Estados Unidos planea introducir un software de reconocimiento de voz que permitirá a los pilotos maniobrar con mayor libertad a la hora de comandar la nave.



Fig. 1.7.4 Reconocimiento de voz aplicado en aviones

Características y propiedades [15]:

- Esta tecnología comenzará a aplicarse a lo largo de 2009 en algunos jets como el *F-35 Lightning II Joint Strike Fighter* evitando el clásico control a través del panel.
- Si bien desde hace tiempo la fuerza aérea de ese país ha probado diferentes sistemas, por el momento se están realizando las pruebas a través de *DynaSpeak*, un software desarrollado por las compañías SRI Internacional y *Adacel Systems* que funcionará mediante un micrófono que se incorpora a la

máscara de oxígeno del piloto y le permitirá la realización de las acciones a través de comandos simples.

- El sistema compatibiliza con la computadora de abordo y brindará datos relacionados con la comunicación y la navegación que el piloto escuchará a través de su casco.

TV con reconocimiento de aplausos y gestos

El nombre de la tecnología es “TV con reconocimiento de aplausos y gestos”, fue creada y presentada por JVC, brinda un sistema de control distinto, pues será con aplausos y gestos la forma en la cual se maneja la TV.



Fig.1.7.5 TV JVC con reconocimiento de aplausos y gestos

Características y propiedades [16]:

- Para activar el “control”, se tendrán que dar tres aplausos, el menú se abre, cuando se quiera seleccionar algo, se tendrá que ir moviendo nuestra mano, pues una cámara está situada en la parte superior de la TV.
- Una vez que se realicen los aplausos se activa y captará el movimiento por el menú. Para seleccionar algo se debe acercar la mano.
- No todo se controla de la misma manera, por ejemplo para el volumen se hace una secuencia de aplausos, la cual puede ser personalizada.

Robot Hitachi con reconocimiento de voz (julio 2008)

El último robot de Hitachi, llamado *EMIEW2*, es lo suficientemente inteligente para diferenciar entre tres voces humanas que suenan simultáneamente gracias a 14 micrófonos montados en su cabeza que ofrece un reconocimiento de voz realmente avanzado.



Fig. 1.7.6 Robot Hitachi

Para ayudar a esto, las funciones de reconocimiento consisten en recibir la fuente del sonido por separado y crear un patrón. El pequeño robot mide 80cm [17].

1.7.1 Aplicaciones del reconocimiento de voz

Medicina

- Permite descripciones habladas de los hallazgos obtenidos en un examen físico u otras observaciones.
- Entrada "on-line" de datos de laboratorio.
- Desarrollo de redes neuronales multicapa, entrenadas y probadas utilizando palabras aisladas, pronunciadas por pacientes con disartria (trastorno del habla).

Operadores automáticos

- Los proveedores de servicios telefónicos ofrecen a sus clientes la posibilidad de obtener un número de teléfono manteniendo un diálogo completo con un sistema activado por la voz.
- Capacidad de entender el habla de quienes realizan las llamadas, incluso de aquellos que hablan con acentos extranjeros, y responder a consultas comunes durante las 24 horas del día, los 7 días de la semana.

Teléfonos Celulares

- Integración de interfaces de audio, voz y video en un único teléfono y en una sola sesión.

Discapacitados

- Para aquellas personas que presentan grandes dificultades en el uso del teclado y el ratón estándar.

Empresas importantes actualmente trabajando en el reconocimiento de voz [10]

- Philips
- Lernout & Hauspie
- Sensory Circuits
- Dragon Systems
- Speechworks
- Vocalis
- Dialogic
- Novell
- Microsoft
- NEC, Siemens, Intel

1.8 CONCLUSIONES DEL CAPÍTULO I

Tuvieron que pasar alrededor de 20 años para obtener el primer producto de reconocimiento de voz, lo que nos indica que no era un proceso sencillo si no que era mucho más confuso de lo que los primeros investigadores supusieron y no se obtuvieron mejores resultados hasta que se implementaron algoritmos basados en los modelos ocultos de Markov y las redes neuronales, así como los sistemas híbridos entre estos dos modelos.

Si bien en la actualidad existen un gran número de productos específicos para el reconocimiento de voz aun queda un largo camino por recorrer. Ya sea mejorando los algoritmos de comparación utilizados para el reconocimiento de voz o en el diseño de filtros que mejoraría los espectros de voz tomados por el micrófono.

CAPÍTULO II: DISEÑO DEL SOFTWARE

Este capítulo muestra los tres procesos que se requieren para procesar la voz y que el programa debe realizar para obtener un reconocimiento exitoso. Además, muestra los distintos tipos de reconocimiento de voz que existen. También muestra la información del dispositivo seleccionado para el reconocimiento de voz, información como la definición de los controladores digitales de señal (DSC), características, diferencias entre un DSC y un DSP (procesador digital de señal), tipos de modelos, encapsulados y el por qué se eligió un DSC como dispositivo principal para el reconocimiento de voz.

El diseño del software consta de tres etapas: la primera etapa inicia en el subtema 2.2.1 con la creación de la biblioteca, que contiene los códigos de las palabras que se compararán con los espectros de voz entrantes, la segunda (subtema 2.2.2) es el programa principal para el reconocimiento de voz que proveerá a la tercera etapa de un código binario de la palabra reconocida. La tercera etapa se muestra en el subtema 2.2.3 y consta del programa que ejecutara la rutina necesaria para cada código binario que recibió por la segunda etapa.

Por último se muestran los pasos a seguir para poder programar la segunda y la tercera etapa y así poder integrar el sistema.

2.1 INFORMACIÓN DEL RECONOCIMIENTO DE VOZ Y DEL DISPOSITIVO ELECTRÓNICO

2.1.1 Procesos del reconocimiento de voz

El reconocimiento de voz generalmente es utilizado como una interfaz entre el humano y la computadora mediante un software.

Para que la voz se transforme en datos debe pasar por los siguientes procesos, pasos con los cuales el programa debe cumplir para un reconocimiento exitoso [18]:

1. Pre-procesamiento: Convierte la entrada de voz a una forma que el reconocedor pueda procesar.
2. Reconocimiento: Identifica lo que se dijo (traducción de señal a texto).
3. Comunicación: Envía lo reconocido al sistema (Software/Hardware) que lo requiere.

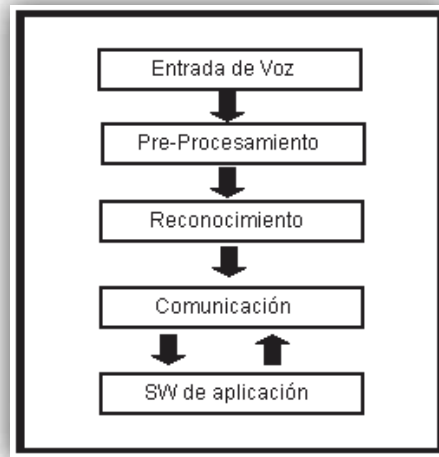


Fig. 2.1.1 Componentes de una aplicación

2.1.2 Tipos de reconocimiento de voz

Sistema Dependiente del Locutor: El software debe ser entrenado para que reconozca cada usuario individual. Sin embargo, este sistema consume mucho tiempo en el ingreso del diccionario en la computadora.

Sistema Independiente del Locutor: El software reconoce palabras de la mayoría de los que hablan sin entrenamiento previo. Para eso usa modelos. Sin embargo, un acento fuerte podría confundir al sistema.

Reconocimiento Discreto del Habla: El locutor deberá hacer pausas entre las palabras para que la computadora pueda reconocer dónde termina cada palabra.

Reconocimiento Continuo del Habla: El que habla puede usar un flujo de conversación normal.

Lenguaje Natural: El locutor puede llegar a conversar con su computadora y decirle algo como: ¿Cuándo podríamos enviarle una docena del artículo 2000 en color azul a Olavarría? ¡Y obtener una respuesta! [19].

2.1.3 Los controladores digitales de señal (DSC o dsPIC's)

Durante las últimas décadas, la investigación en el campo del reconocimiento automático del habla se ha venido desarrollando de una forma intensa, empujada por los avances en procesamiento de señal, algoritmos, arquitecturas y plataformas de cómputo.

DsPIC es un nombre genérico que se utiliza para referirse a los Controladores Digitales de Señales (DSC) que ha diseñado la compañía *Microchip Technology Inc.* Para facilitar a los usuarios de los microcontroladores PIC (Circuito integrado programable) la transición al campo de las aplicaciones de los Procesadores Digitales de Señales (DSP), que han presentado grandes avances en áreas como las comunicaciones, los sensores, el procesamiento de la imagen y del sonido.

Los DSC son dispositivos que combinan la arquitectura y la programación de los microcontroladores PIC de 16 bits con los recursos hardware y software necesarios para soportar las prestaciones fundamentales de los DSP.

2.1.3.1 Procesamiento Digital de Señales (DSP)

El mundo que nos rodea está caracterizado por contar con un gran número de señales analógicas o continuas que varían entre un valor mínimo y otro máximo pasando por valores finitos. La temperatura, la luz y la presión atmosférica son algunos ejemplos de magnitudes analógicas típicas en nuestro entorno.

Desde que en 1971 Intel comercializó el primer microprocesador, las posibilidades y ventajas del procesamiento digital se incrementaron de forma exponencial. Un microprocesador es la parte más compleja e importante de un computador digital. En nuestros días, existe una gran variedad de circuitos integrados con millones de transistores y otros elementos repartidos en pocos mm² de silicio que configuran potentes computadores digitales a un costo accesible.

Un claro ejemplo de procesamiento digital de señales es el tratamiento de la señal de sonido captada por un micrófono, el cual queda ilustrado en la figura 2.1.2.

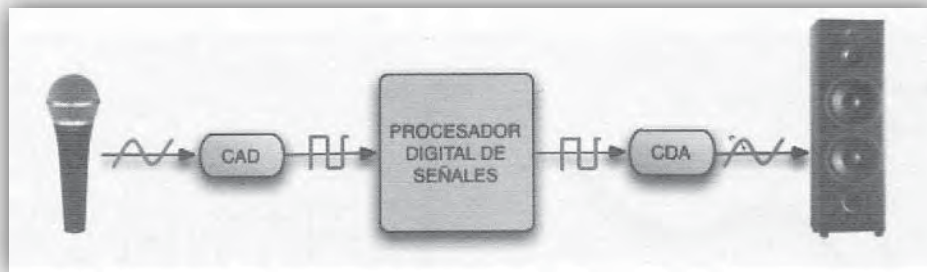


Fig. 2.1.2 El micrófono transforma en una señal analógica las presiones acústicas de la voz. Dicha señal se aplica a un CAD (convertidor analógico a digital), cuya salida se trata en un procesador digital de señales según el algoritmo implementado en el programa con el fin de eliminar ruido de la señal obtenida del micrófono. El resultado de este procesamiento se aplica a un CDA (convertidor digital a analógico) para obtener de nuevo una señal analógica que se reproduce en un altavoz.

2.1.3.2 Microcontroladores y DSP

Los microcontroladores clásicos, denominados de forma reducida MCU (*Micro Controller Unit*), son circuitos integrados que contienen un procesador digital completo junto a diversos periféricos auxiliares que facilitan el desarrollo de las aplicaciones a las que se dedican. Su parecido con los DSP es muy grande, pero las diferencias que los distinguen hacen que sus campos de aplicación sean diferentes.

Las instrucciones aritméticas complejas de los MCU (*Micro Controller Unit*) se ejecutan en varios ciclos, mientras que las de los DSP sólo precisan uno. En los DSP siempre se dispone de convertidores AD rápidos y precisos. Dado el carácter marcadamente matemático de los programas para DSP, éstos están preparados para ser programados con lenguajes de alto nivel, como el lenguaje C. La velocidad y el rendimiento de los DSP son muy superiores a los habituales en los MCU.

2.1.3.3 El concepto DSC

La compañía de Microchip ha unido toda la potencia y posibilidades de los microcontroladores de 16 bits con las prestaciones más interesantes de los DSP para fabricar un nuevo circuito integrado denominado DSC.

El DSC es muy parecido a un microcontrolador MCU de 16 bits en cuanto a la arquitectura, repertorio de instrucciones y precio, pero con el rendimiento y las prestaciones de un DSP.

Existen varias formas de medir la velocidad de un procesador digital de señales, aunque el parámetro de medida más usual es el tiempo de ciclo de instrucción, que es el tiempo necesario para ejecutar la instrucción más rápida del procesador. Su inverso dividido por un millón da lugar al rendimiento del procesador en millones de instrucciones por segundo (MIPS).

Estos dispositivos se caracterizan por alcanzar un rendimiento de 40 MIPS (los Pic18F de 16 bits alcanzan una rapidez de 10 MIPS) e integrar memoria flash de alta calidad junto a novedosos recursos hardware y manteniendo la compatibilidad de los diversos encapsulados.

Los DSC se comercializan en diferentes dispositivos agrupados en dos familias:

1. Familia dsPIC30F (El que concierne al proyecto)
2. Familia dsPIC33F

2.1.3.4 Características Generales de los dsPIC30F

Tabla 2.1.3.1 Características más importantes de la familia dsPIC30F

RECURSO	RANGO DE VALORES
Memoria de programa FLASH	12kb - 144kb
Memoria de datos RAM	512Bytes - 8kb
Memoria de datos EEPROM	1kb - 4kb
Patillaje de encapsulado	18 - 80 patitas
Temporizadores de 16 bits	Hasta 5
Módulo de captura	Hasta 8 entradas
Módulo Comparador/PWM	Hasta 8 salidas
Módulo PWM de control de motores	De 6 a 8
Convertor A/D de 10 bits	500 kbps, hasta 16 canales
Convertor A/D de 12 bits	100 kbps, hasta 16 canales
UART	1 - 2
SPI™ (8 -16 bits)	1 - 2
I ² C™	1 Módulo
Interfaz CODEC	1
CAN	1 - 2

El voltaje de alimentación admite un rango comprendido entre 2,5 y 5,5 VDC (Voltaje de corriente directa). Se tolera una temperatura interna entre -40 °C y 85 °C y una externa entre -40 °C y 125 °C. El rendimiento alcanza los 30 MIPS con un voltaje de alimentación entre 4,5 y 5,5 VDC.

Otra característica importante en los dsPIC30F es la de admitir hasta 45 fuentes distintas de petición de interrupción con 7 niveles de prioridad, de las cuales 5 son externas. Hay modelo de dsPIC30F que disponen de hasta 54 patitas de E/S programables y con otras funciones multiplexadas con un consumo de 25mA [20].

2.1.3.5 Los modelos de la familia dsPIC30F

La familia de dsPIC30F se clasifica en 19 modelos según la marca comercial de la empresa Microchip. Actualmente se fabrican y comercializan tres categorías atendiendo a su aplicación más apropiada:

1. Dispositivos dsPIC30F de propósito general
2. Dispositivos dsPIC30F para el control de sensores
3. Dispositivos dsPIC30F control de motores y sistemas de alimentación

2.1.3.6 Encapsulado y diagramas de conexiones

Con la finalidad de soportar todo tipo de diseños y necesidades los modelos dsPIC30F están encapsulados desde 18 hasta 80 patitas, algunos con doble hilera de patitas, tipo PDIP y SPDIP y SOIC. Otros están preparados para el montaje superficial como los tipos TQFP y QFN (Figura 2.1.3).

- PDIP-----Plastic Dual In-Line
- SPDIP---Shrink (Encoger) Plastic Dual in-line
- SOIC----Plastic Small Outline
- TQFP----Plastic Thin Quad Flatpack
- QFN-----Plastic Quad FlatPack

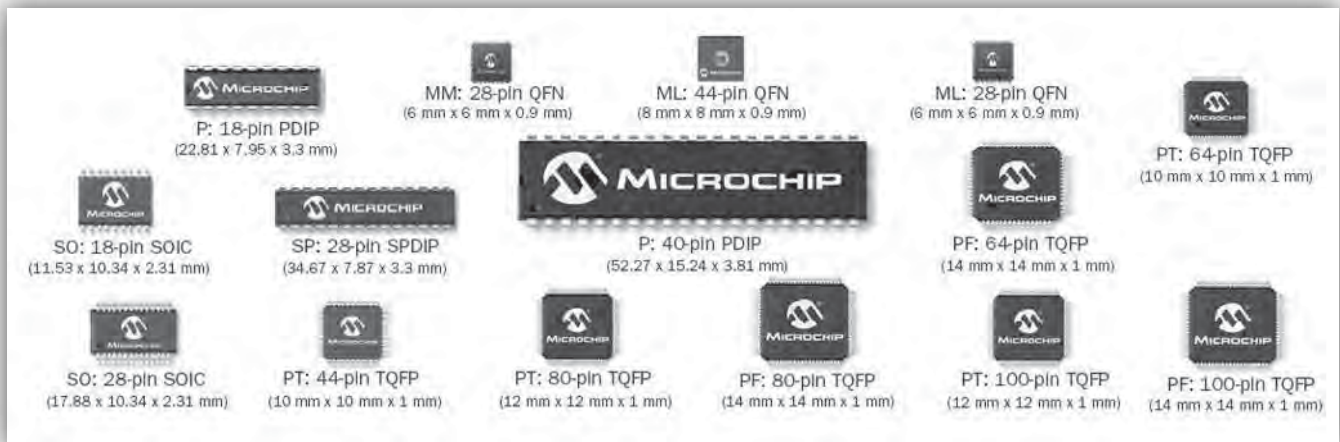


Fig. 2.1.3 Modelos de encapsulado de la familia dsPIC30F

La nomenclatura de los dsPIC30F formada por números y letras expresan las especificaciones particulares y su significado se describe en la figura 2.1.4.

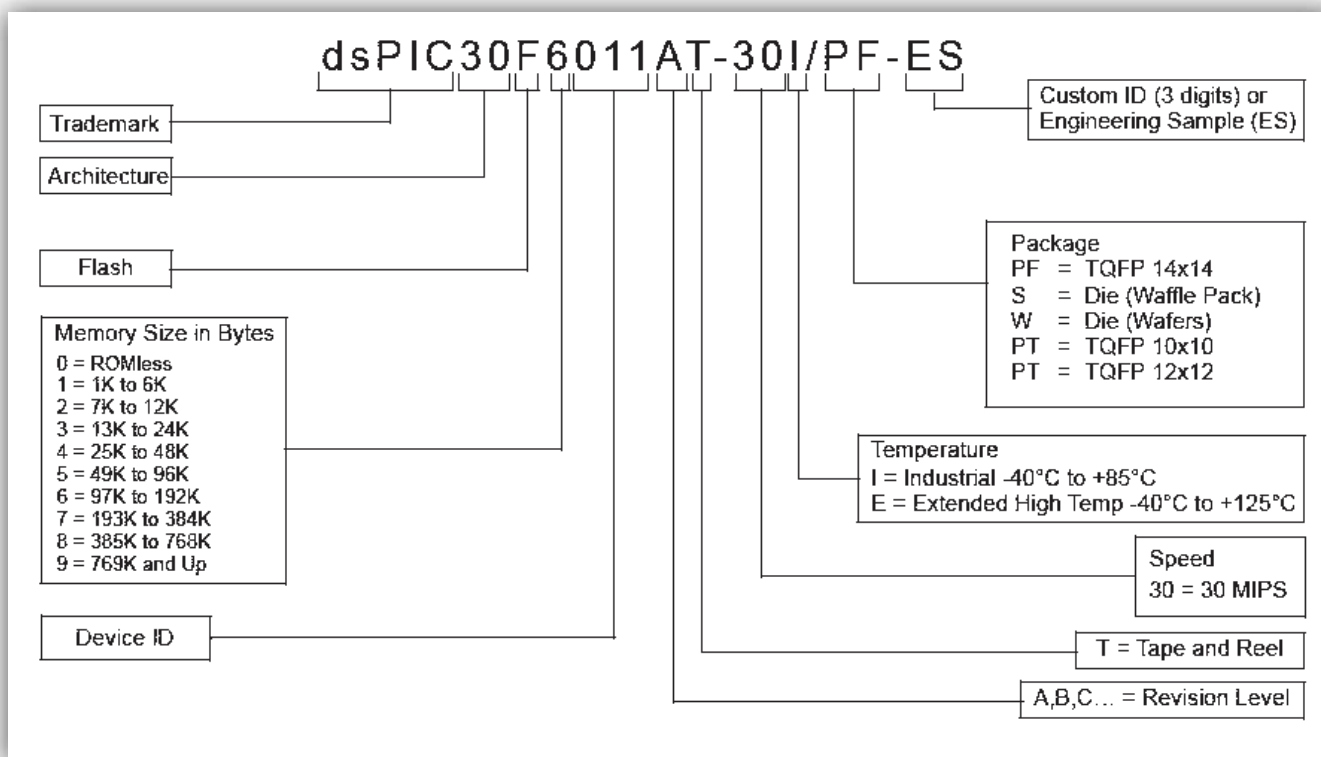


Fig. 2.1.4 Campos en que se descompone la nomenclatura de la familia dsPIC30F y significado de cada uno de ellos. También es aplicable a los dsPIC33F

En la tabla 2.1.3.2 Se describe el símbolo, el significado y la dirección de las patitas más representativas en los dsPIC30F.

Tabla 2.1.3.2 Símbolo, dirección y descripción de las patitas más significativas de los encapsulados de los dsPIC30F.

NOMBRE DE LAS PATITAS	DIRECCIÓN	DESCRIPCIÓN
RA_n	E/S	PATITAS DE E/S DIGITAL DEL PUERTO A
RB_n	E/S	PATITAS DE E/S DIGITAL DEL PUERTO B
RC_n	E/S	PATITAS DE E/S DIGITAL DEL PUERTO C
RD_n	E/S	PATITAS DE E/S DIGITAL DEL PUERTO D
RE_n	E/S	PATITAS DE E/S DIGITAL DEL PUERTO E
RF_n	E/S	PATITAS DE E/S DIGITAL DEL PUERTO F
RG_n	E/S	PATITAS DE E/S DIGITAL DEL PUERTO G
V_{DD}	ALIMENTACIÓN	VOLTAJE POSITIVO PARA LÓGICA Y E/S
V_{SS}	ALIMENTACIÓN	TIERRA PARA LÓGICA Y E/S
AV_{DD}	ALIMENTACIÓN	VOLTAJE POSITIVO PARA MÓDULO ANALÓGICO

Tabla 2.1.3.2 (continuación) Símbolo, dirección y descripción de las patitas más significativas de los encapsulados de los dsPIC30F.

NOMBRE DE LAS PATITAS	DIRECCIÓN	DESCRIPCIÓN
AV_{SS}	ALIMENTACIÓN	TIERRA PARA MÓDULO ANALÓGICO
V_{REF+}	ALIMENTACIÓN	ENTRADA VOLTAJE REFERENCIA + ANALÓGICO
V_{REF-}	ALIMENTACIÓN	ENTRADA VOLTAJE REFERENCIA - ANALÓGICO
INT_n	E	INTERRUPCIÓN EXTERNA n
AN_n	E	CANALES ANALÓGICOS DE ENTRADA
$CLK1/CLK0$	E/S	ENTRADA/SALIDA RELOJ EXTERNO
CN_n	E	ENTRADAS DE NOTIFICACIÓN DE CAMBIO
$COFS-CSCK-CSDI-CSD0$	E/S	PATITAS PARA INTERFAZ DE CONVERSIÓN DE DATOS
$C1RX-C1TX-C2RX-C2TX$	E/S	PATITAS DE RECEPCIÓN/TRANSMISIÓN DE CAN1 Y CAN2
EMU_{xx}	E/S	PATITAS PARA EL CANAL DE COMUNICACIÓN ICD
IC_n	E	ENTRADAS DEL MÓDULO DE CAPTURA
$OSC1-OCS2$	E/S	ENTRADA/SALIDA DEL OSCILADOR DE CRISTAL
$SOSC1-SOSC2$	E/S	ENTRADA/SALIDA OSCILADOR 32KHz BAJA POTENCIA
$OCFA-OCFB$	E	ENTRADAS A-B FALTA COMPARADOR
OC_n	S	CANALES DE SALIDA DEL COMPARADOR
$PGD-PGC$	E/S	PATITAS PARA PROGRAMACIÓN SERIE
T_nCK	E	ENTRADA RELOJ EXTERNO TIMER n
$U1RX-U1TX-U2RX-U2TX$	E/S	RECEPCIÓN/TRANSMISIÓN UART1 Y UART2
$SS1-SS2-SCK1-SCK2-SDI1-SDI2-SDO1-SDO2$	E/S	PATITAS PARA MÓDULOS SPI1 Y SPI2

2.1.3.7 Aplicaciones recomendadas

El fabricante (*Microchip*) ha clasificado en 6 áreas las posibles aplicaciones de los DSC [20].

A) CONTROL DE MOTORES

- Bombas industriales
- Lavadoras y aspiradoras
- Tensores de cinturones de seguridad
- Calefacción, ventilación y aire acondicionado
- Apertura de puertas

B) CONTROL DE SENSORES

- Sensores de torsión
- Sensores de presión
- Sensores de vibración
- Sensores de golpes
- Sensores químicos y de gases

C) AUTOMOCIÓN

- Control de potencia
- Control de estabilidad y balanceo
- Cancelación de ruido en el habitáculo
- Bolsas de aire (*Airbag*)
- Control de combustible

D) CONECTIVIDAD EN INTERNET

- Monitorización remota
- Contadores automáticos de agua y gas
- Instrumentación médica remota
- Sistemas de seguridad
- Diagnóstico industrial remoto

E) AUDIO

- Reconocimiento de voz
- Altavoces telefónicos
- Cancelación de ruidos
- Reproducción de sonido digital
- Auriculares para la eliminación de ruidos
- Puertos de comunicación para manos libres
- Micrófono activado por la voz

F) GESTIÓN Y MONITORIZACIÓN DE SISTEMAS DE ALIMENTACIÓN

- Convertidores AC-DC
- Convertidores DC-AC
- Fuentes de alimentación interrumpibles (UPS)
- Inversores
- Vehículos eléctricos
- Gestión de alimentación y ahorro de consumo
- Detección de fallos

2.1.4 Selección del controlador digital de señal (DSC)

Para la selección de un microcontrolador PIC24 de 16 bits o un dsPIC (Controlador digital de señal), se consideran los siguientes factores en el diseño:

- La velocidad del microcontrolador (MIPS, frecuencia de reloj para los canales de comunicaciones, etc.)
- Los tamaños de la memoria (memoria del programa/*Flash*, RAM, datos EEPROM)
- El número de patitas de entrada/salida (I/O) requeridas
- Los periféricos necesarios (A/D, Input Capture, Output Compare, Motor Control PWM, SPI™, UART, etc.)
- Tamaño físico (factor de forma/encapsulado)
- Niveles de voltaje de la entrada análoga

Las cualidades requeridas para su diseño de 16 bits son un factor crítico para el equilibrio del sistema en costo/valor para la solución requerida. El balance en el número de patitas de entrada/salida (I/O), los tamaños de la memoria, la frecuencia de reloj/CPU, los rangos de operación del voltaje y la temperatura toda contribuye al costo del dispositivo [21].

Las características del dsPIC seleccionado según los parámetros antes mencionados, se muestran en la tabla 2.1.4.1

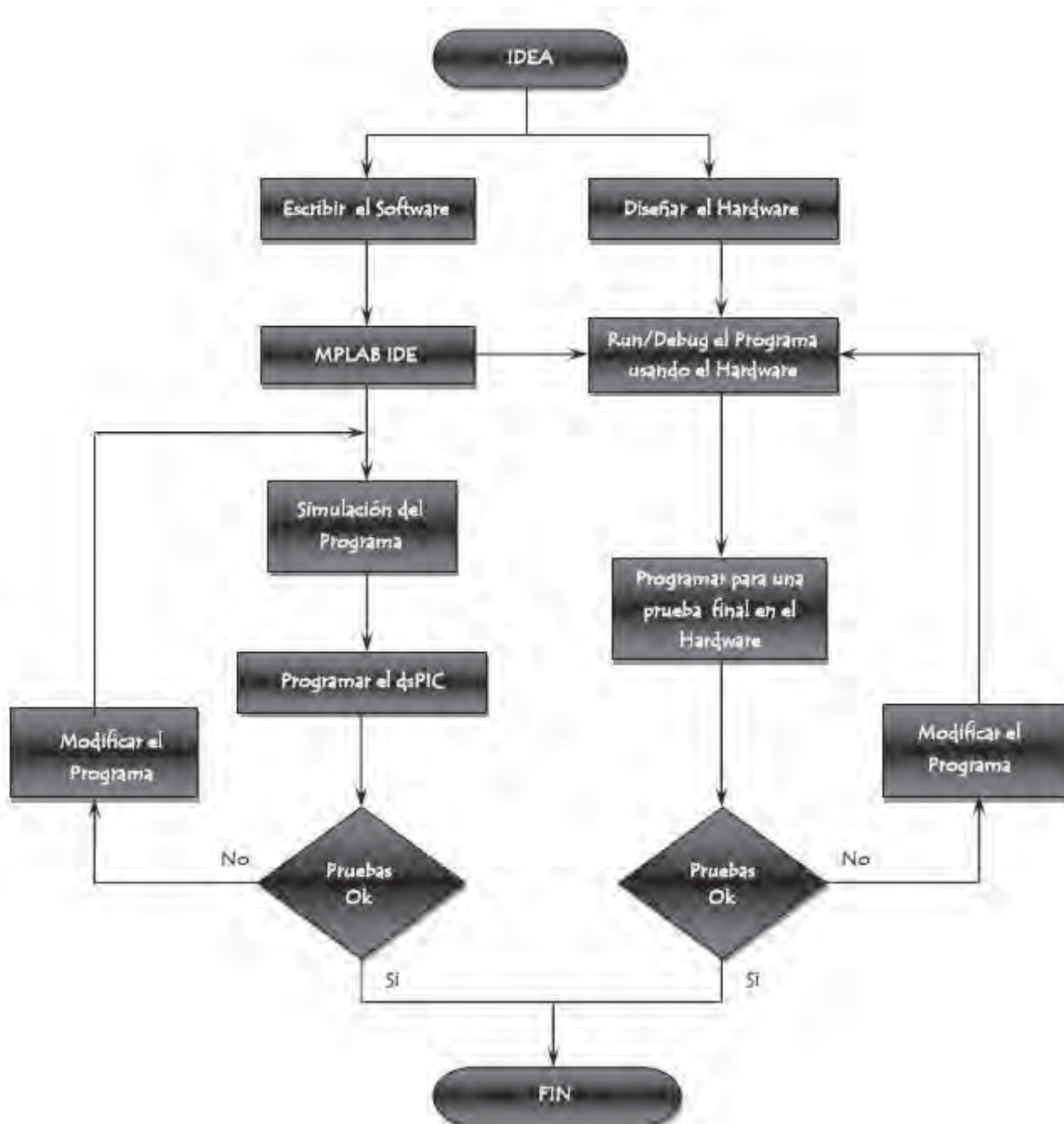
Tabla 2.1.4.1 Características del dsPIC30F6014A

DSC Seleccionado		dsPIC30F6014A-20E/PT
Velocidad del microcontrolador		30 MIPS
Tamaño de memoria	Flash	144 KBytes
	RAM	8 192 Bytes
	Datos EEPROM	4096 Bytes
Número de pines		80 Pines, de éstos 68 son I/O
Periféricos necesarios		16 channels - A/D 12-Bit
		DCI (I2S y AC'97)
		1 - I ² C
		8 - Control PWM
Tipo de encapsulado		TQFP (Thin Quad Flat Pack)
Niveles de voltaje		2.3 a 5.5 volts

2.2 METODOLOGÍA PARA EL RECONOCIMIENTO DE VOZ

Para el diseño del software y el circuito se propuso una metodología de acuerdo al diagrama de flujo que aparece en el manual *Beginners Guide to the Microchip PIC* por Nigel Gardner [22].

Diagrama 2.2 Diagrama de flujo global para el reconocimiento de voz



2.2.1 Etapa I. Creación de la biblioteca para el reconocimiento de voz

Siguiendo el diagrama de flujo, primero se desarrollará el software para el reconocimiento de voz, iniciando con la creación de la biblioteca.

La biblioteca de reconocimiento de voz para el dsPIC30F proporciona un interfaz entre el audio del usuario y el programa de aplicación, permitiéndole al usuario controlar la aplicación de palabras articuladas individualmente que estarán contenidas dentro de una librería predefinida. Las palabras elegidas para la biblioteca son específicamente relevantes para la interacción entre el programa de aplicación y el usuario. El programa aplicado toma las acciones mostradas en la figura 2.2.1.

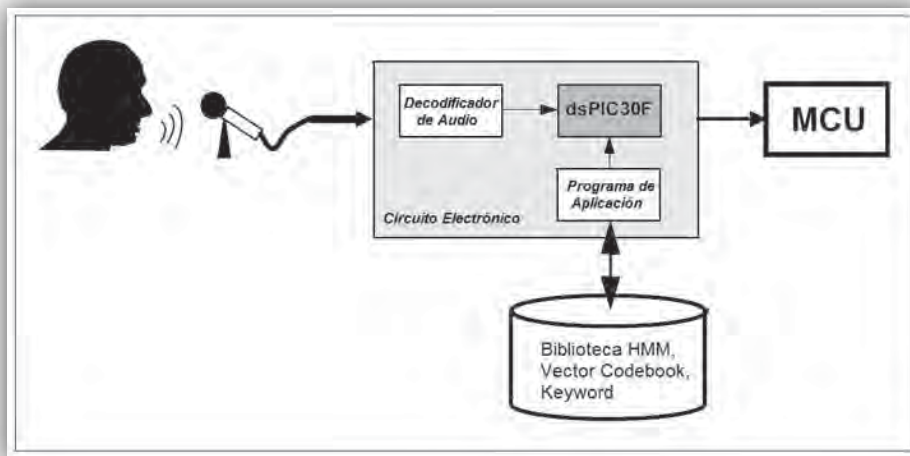


Fig. 2.2.1 Descripción del reconocimiento de voz

La biblioteca de reconocimiento usa un algoritmo de reconocimiento por palabra basado en el modelo oculto de Markov (*HMM-Hidden Markov Model*). Una vez que la palabra es procesada por la transformada rápida de Fourier se analiza en un marco usando un algoritmo *RASRA-PLP* y cuantificado dentro de vectores característicos de sonido para después ser comparados con el vector *codebook*. Después los vectores característicos son examinados para determinar que palabra del modelo HMM se empareja lo más cerca posible a la señal de entrada, obteniendo como resultado el código de la palabra de entrada.

La biblioteca del reconocimiento de voz para el dsPIC30F funciona en condiciones ruidosas y libres de ruido. La interferencia puede incluir ruido blanco, ruido de la oficina o cualquier mezcla de ruido. El cociente de la relación señal/ruido (SNR) no debe ser menor a 15 dB.

La biblioteca puede contener hasta un total de 100 palabras si así lo requiere el usuario. Esta biblioteca se construye con un palabra clave (*Keyword*) que dependiendo del modo usado, se podrá utilizar como una auto prueba para el circuito y para activar o iniciar una sesión de reconocimiento. Además, la activación de esta palabra clave es recomendada por seguridad para prevenir accidentes respondiendo a palabras cuando no debería hacerlo.

Para un reconocimiento exitoso se requiere que las palabras sean separadas por una pausa de por lo menos medio segundo, pero no mayor a cinco segundos, ya que el software pasara a modo inactivo (*sleep*) y una nueva sesión de reconocimiento deberá ser iniciada.

La empresa de Microchip dispone de una herramienta específicamente útil para el reconocimiento de voz. Se trata de la herramienta *dsPIC30F Speech Recognition Word Library Builder (SW300010)*. Puede encontrarse en su sitio de internet www.microchip.com una versión de prueba válida por dos meses [23].

Hay tres tareas primarias involucradas para la creación de la biblioteca como se resume en figura 2.2.2 [24].

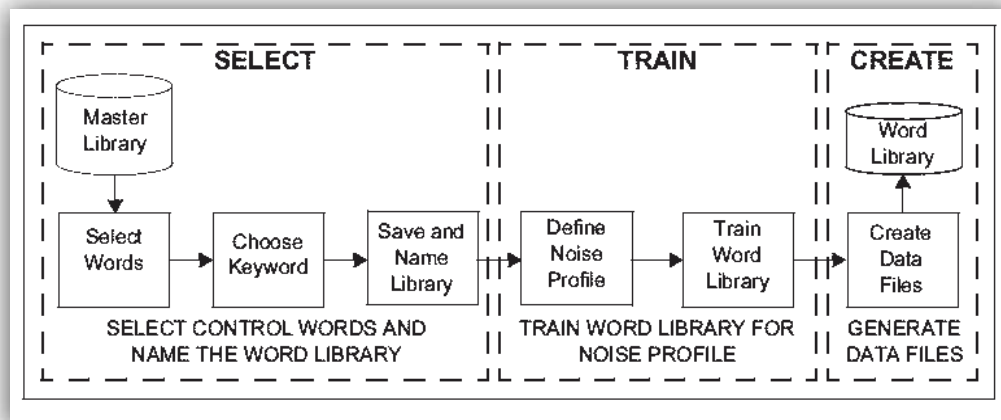


Fig. 2.2.2 Proceso para creación de la biblioteca

1. Seleccionar las palabras específicas de la biblioteca principal (Fig. 2.2.3), identificar la *keyword* y nombrar la biblioteca.
2. Definir el perfil del ruido de acuerdo al ambiente donde se desarrollara la aplicación y adiestrar la biblioteca para este ambiente.
3. Crear los archivos necesarios para la biblioteca de reconocimiento y para la aplicación del usuario.

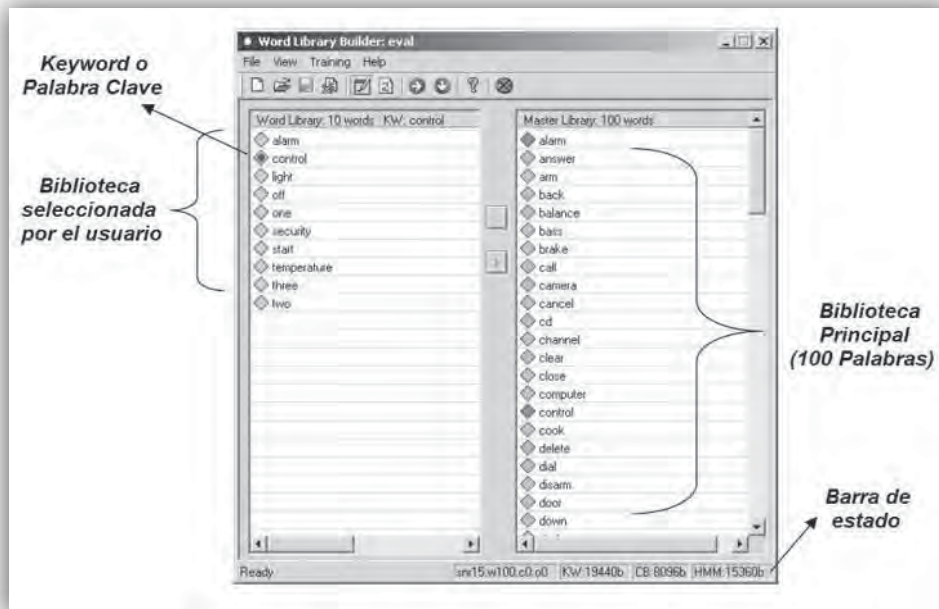


Fig. 2.2.3 Entorno de Word Library Builder

Selección de las palabras

Una vez instalado el demo de *Word Library Builder*, se abre la aplicación y se genera un nuevo proyecto, como se muestra en la figura 2.2.4.

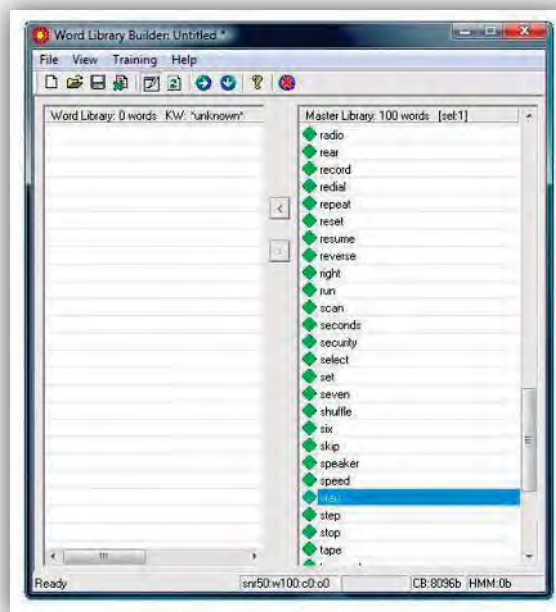


Fig. 2.2.4 Lista de la biblioteca principal

Se seleccionan las palabras y se escoge con doble clic la palabra que se utilizará como *keyword*, éste se indicará con un icono rojo (figura 2.2.5).

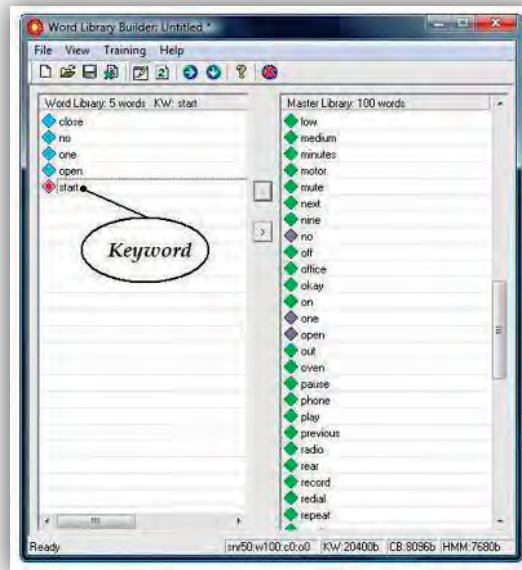


Fig. 2.2.5 Selección de la *Keyword*

Adiestramiento de las palabras

Después de guardar la biblioteca, se debe adiestrar a la biblioteca para las especificaciones de ruido de acuerdo al ambiente donde se desarrollara. Este proceso determina qué vector *codebook* ofrece las mejores características de emparejado para las palabras seleccionadas y el perfil de ruido. A su vez, el vector *codebook* seleccionado se usa para entrenar el algoritmo basado en el modelo oculto de *Markov* para cada palabra.

El perfil de ruido predefinido y sus características se muestran en la tabla 2.2.1 [24].

Tabla 2.2.1 Perfil de ruido predefinido

Profile	SNR level (dB)	Noise type (%)		
		White	Car	Office
1 (White – 50 dB)	50	100	0	0
2 (White – 15 dB)	15	100	0	0
3 (Car – 15 dB)	15	0	100	0
4 (Office – 15 dB)	15	0	0	100
5 (White/Car – 15 dB)	15	50	50	0
6 (White/Office – 15 dB)	15	50	0	50
7 (Car/Office – 15 dB)	15	0	50	50
8 (Mixed – 15 dB)	15	34	33	33

Para continuar con el adiestramiento se selecciona el menú *Training>Train Word Library*. Después se selecciona el tipo de perfil de ruido *User* o *Standard*. Como lo muestra la figura 2.2.6.

1. Si se selecciona *Standard*. Se podrá seleccionar uno de los perfiles predefinidos.
2. Si se selecciona *User*. Este permitirá la asignación arbitraria del nivel SNR (*Sign Noise Ratio*) y el tipo de ruido con los siguientes límites definidos:

El cociente de la relación señal/ruido (SNR) para cualquier ruido puede ser entre 15 dB y 50 dB

La suma de los porcentajes para todos los tipos de ruido (*White + Car + Office*) debe ser igual al 100%

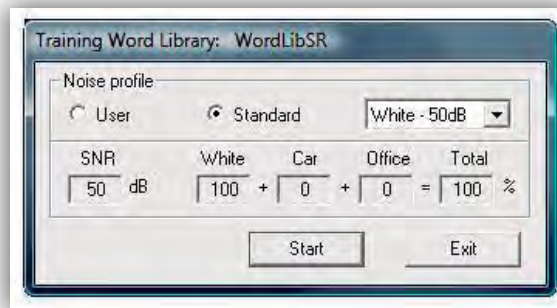


Fig. 2.2.6 Perfiles de la biblioteca de ruido

Una vez seleccionado el tipo de ruido se da clic en Start. Después aparecerá una ventana con el dialogo: *Training is Completed!* (figura 2.2.7), se da OK



Fig. 2.2.7 Adiestramiento completo

Crear archivos de salida

La última tarea para crear la biblioteca es crear los archivos necesarios para la interfaz con el reconocimiento de voz del dsPIC30F y la aplicación para lo que fue creado el reconocimiento. Se generan cuatro archivos en el proceso de adiestramiento (Tabla 2.2.2)

Tabla 2.2.2 Archivos generados por la biblioteca

[Nombre del archivo]_int.s	Contiene los datos para el vector <i>codebook</i> , el Modelo Oculto de <i>Markov</i> y la Palabra clave que se asignarán a la memoria interna.
[Nombre del archivo]_ext.s	Contiene los datos para el vector <i>codebook</i> , el Modelo Oculto de <i>Markov</i> y la Palabra clave que se asignarán a la memoria externa.
[Nombre del archivo].h	Archivo principal que contiene un conjunto de constantes para cada palabra en la biblioteca. Este archivo es utilizado para interpretar el valor regresado por la función <i>SR_Recognizer()</i> de la biblioteca de reconocimiento de voz del dsPIC30F.
[Nombre del archivo]_voc.s	Contiene los datos de la cadena ASCII (<i>American Standart Code for Information interchange</i>) de cada palabra en la biblioteca.

Se selecciona *Training>Create Data*. El diálogo de asignación de memoria se abre como lo muestra la figura 2.2.8.

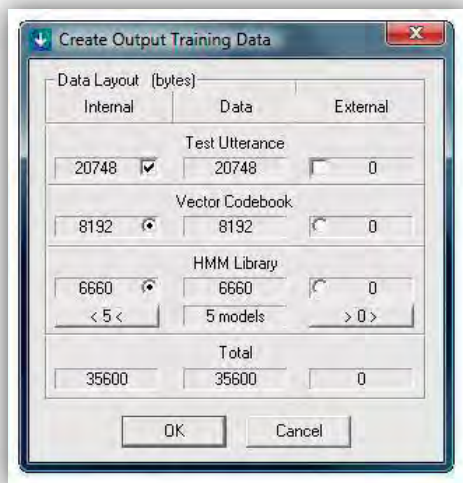


Fig. 2.2.8 Asignación de memoria

En esta ventana (Fig. 2.2.8) se decide si el archivo de la pronunciación de la palabra clave será incluido o no. Si el archivo es incluido, se decide si se coloca en la memoria interna o externa. También permite asignar el vector *codebook* a la memoria interior o externa. Los archivos se pueden asignar en la memoria externa o interna o una mezcla de ambas memorias.

Para crear los archivos finales se da clic en *OK*. Cuando se complete la tarea se debe finalizar la acción con *OK* (Figura 2.2.9)

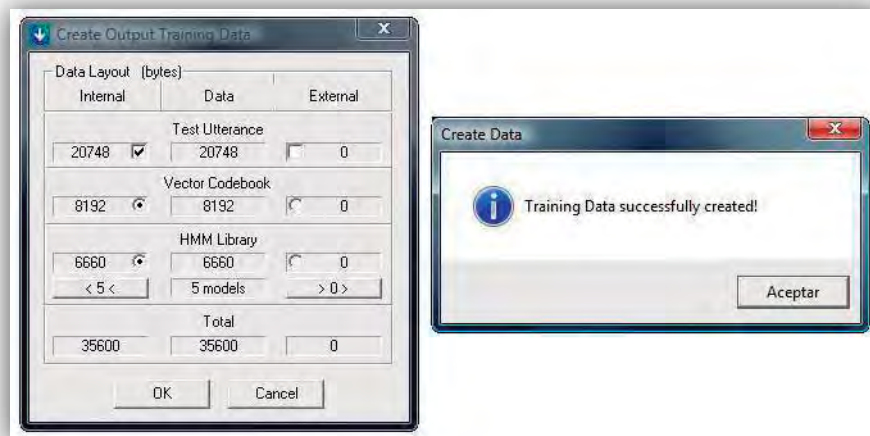


Fig. 2.2.9 Finalizar datos de salida

Para una biblioteca bien seleccionada se puede esperar una exactitud mayor al 90%, aunque también depende de otros parámetros, tales como:

- La omisión de palabras que son similares, para la selección de una buena biblioteca
- La correcta pronunciación de las palabras es vital
- El tamaño de la biblioteca también afecta, a menos palabras mayor exactitud
- La sensibilidad del micrófono que se utilice para la pronunciación

Las extensiones de la biblioteca finales que se utilizan en el programa para el reconocimiento de voz son:

Word Library Builder	WordLibSR_voc.s
	WordLibSR_int.s
	WordLibSR.h

En el Anexo II se puede ver el código principal creado por la biblioteca.

2.2.2 Etapa II. Programa principal para el reconocimiento de voz

2.2.2.1 Construcción del proyecto principal para el dsPIC30F

El proyecto se creó utilizando la herramienta de microchip *MPLAB IDE v8.10* versión estudiantil. Esta versión está disponible en su sitio en internet [25].

Una vez instalado el programa se utiliza la herramienta *Project Wizard*, que ayuda a la creación de nuevos proyectos. Antes de empezar es recomendable crear una carpeta lo más cerca posible al disco duro.

Selección del dispositivo

1. Iniciar *MPLAB IDE*.
2. Del menú *Project*, seleccionar *Project Wizard*.
3. Seleccionar el dispositivo a utilizar, en este caso el dsPIC30F6014A, clic *Siguiente*> (Figura 2.2.10).

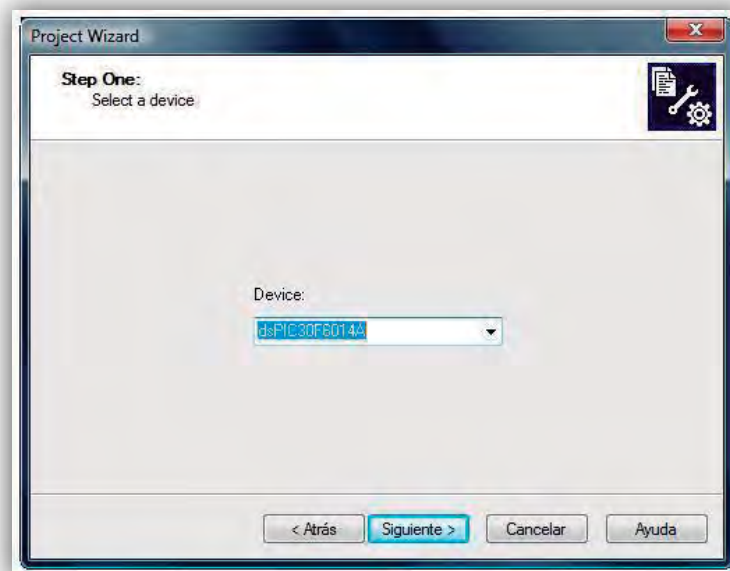


Fig. 2.2.10 Selección del dispositivo

Selección del Lenguaje

1. Del menú *Active Toolsuite*, seleccionar *Microchip C30 Toolsuite*.
2. En el menú *Toolsuite contents*, seleccionar las siguientes direcciones:

MPLAB ASM30 Assambler (pic30-as.exe)	c:\Program Files\Microchip\MPLAB C30\bin\pic30-as.exe
MPLAB C30 C Compiler (pic30-gcc.exe)	c:\Program Files\Microchip\MPLAB C30\bin\pic30-gcc.exe
MPLAB LINK 30 Object Linker (pic30-ld.exe)	c:\Program Files\Microchip\MPLAB C30\bin\pic30-ld.exe
Lib30 Archiver (pic30-ar.exe)	c:\Program Files\Microchip\MPLAB C30\bin\pic30-ar.exe

3. Clic siguiente para continuar, con la creación del proyecto (figura 2.2.11)

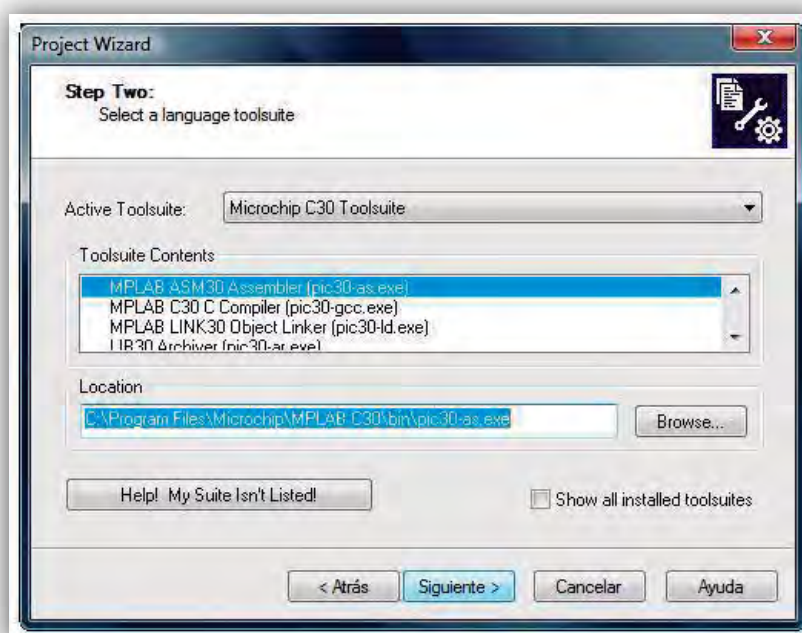


Fig. 2.2.11 Selección del lenguaje

Nota: Si no aparece en *Active Toolsuite* Microchip C30 Toolsuite, seleccionar el menú *Project – Set Language Tool Locations*, en *Registered Tools* se busca *Microchip C30 Toolsuite* y se selecciona, ahora si aparecerá en el menú de *Project Wizard*.

Nombrar el proyecto

Para el paso tres, se selecciona *Browse*, para nombrar el proyecto y la ubicación de los archivos que se generan para el proyecto (figura 2.2.12).

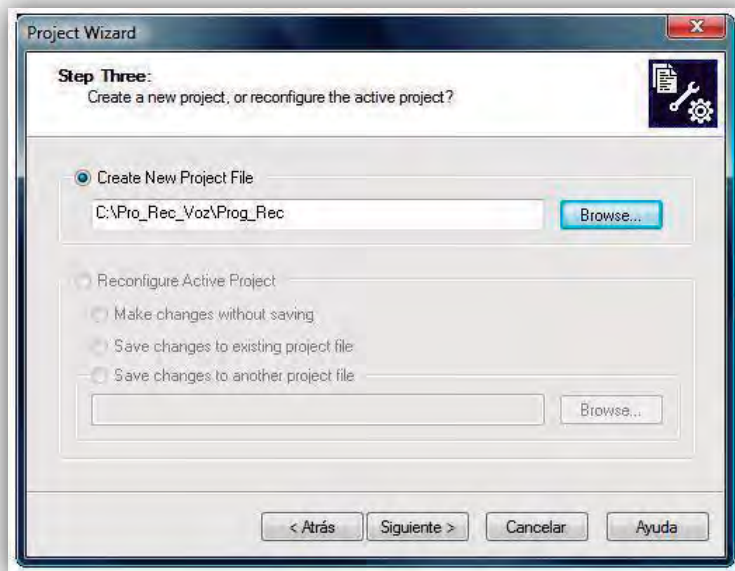


Fig. 2.2.12 Nombre del Proyecto

Agregar archivos al proyecto

Los siguientes archivos se agregarán al proyecto ya que son necesarios para el reconocimiento de voz. En la ventana de la izquierda se buscan los siguientes archivos para ser agregados con `add>>` a la ventana de la derecha, para continuar se da clic en siguiente (figura 2.2.13).

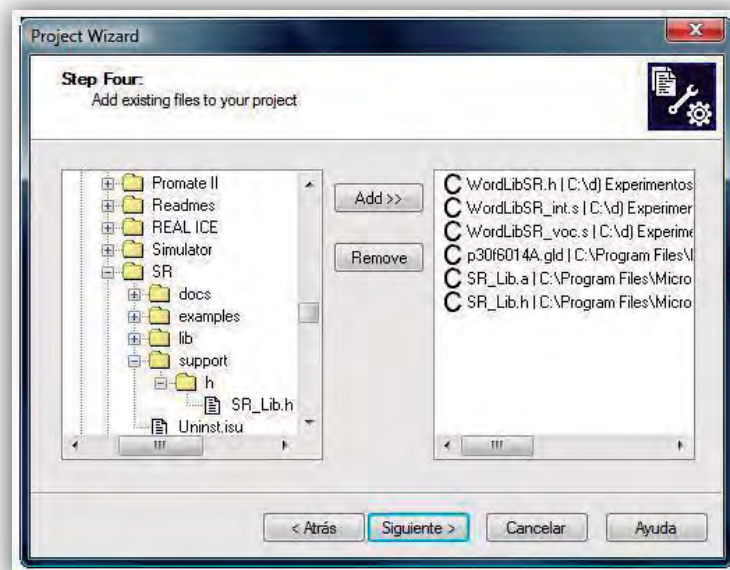


Fig. 2.2.13 Archivos agregados al proyecto

Archivos creados por la biblioteca

Archivo	Ubicación
WordLibSR_int.s	Carpeta de la biblioteca (punto 2.6)
WordLibSR_voc.s	Carpeta de la biblioteca (punto 2.6)
WordLibSR.h	Carpeta de la biblioteca (punto 2.6)

Archivos de MPLAB IDE

Archivo	Ubicación
p30f60a4A.gld	c:\Program Files\Microchip\MPLAB C30\support\ dsPIC30F\gld
SR_Lib.a	c:\Program Files\Microchip\MPLAB IDE\SR\lib\ SR_Lib.a
SR_Lib.h (Arch_Arranq_Lib.h)	c:\Program Files\Microchip\MPLAB IDE\SR\support\ h\SR_Lib.h

Por último aparecerá una ventana donde resumen los parámetros del proyecto generado, como lo muestra la figura 2.2.14, clic para finalizar el proyecto.

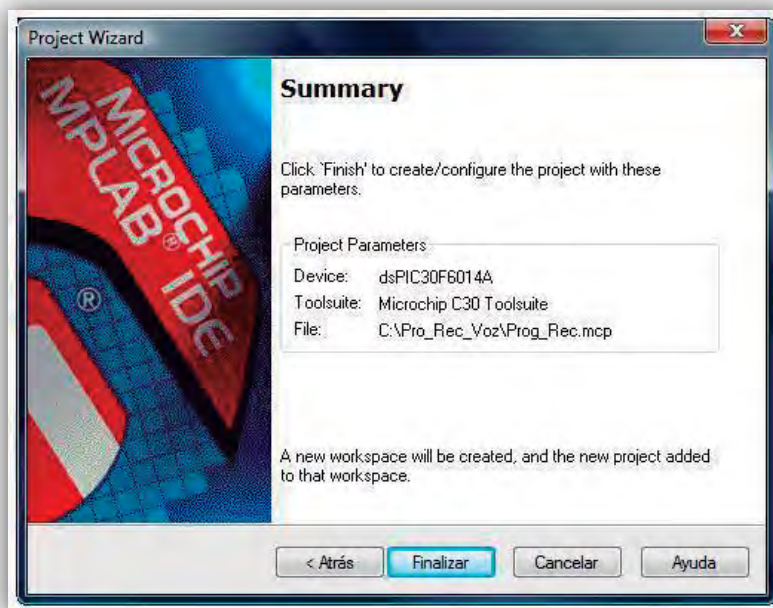


Fig. 2.2.14 Finalizar proyecto

2.2.2.2 Programa para el dsPIC30F

Una vez creado el proyecto, el siguiente paso es realizar los programas para configurar el dsPIC30F y para integrar los archivos de la librería principal.

Para que el dispositivo seleccionado trabaje y pueda tener un reconocimiento exitoso se necesitan de los siguientes programas, los códigos se muestran en el anexo III. La descripción de cada programa se muestra a continuación:

- Programa **Config_Dip_SR.c** - Este programa contiene los ajustes de interrupción.
- Programa **Config_Param_SR.c** - Ajusta los parámetros de inicialización para el SR (reconocimiento de voz).
- Programa **Variables_SR.h** - En esta parte se realiza la declaración de las variables.
- Programa **Prog_Rec.s** - Programa principal y declaraciones globales. Contiene la integración de los códigos anteriores, reconoce la palabra que entra y manda el código al PIC (Circuito integrado programable).

El total de archivos agregados y creados para el reconocimiento de voz, se muestra en la figura 2.2.15

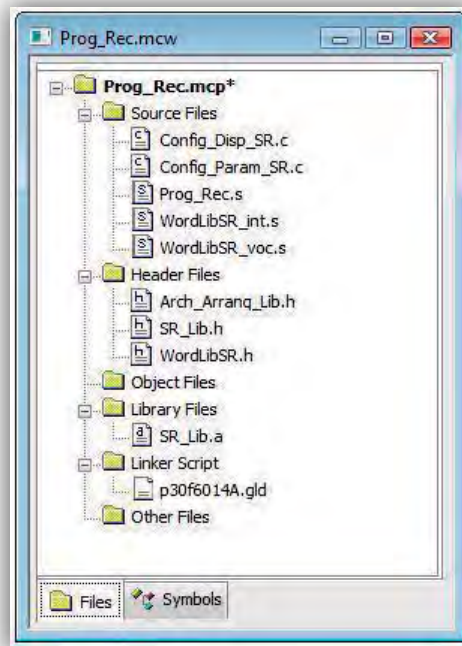


Fig. 2.2.15 Archivos totales para el proyecto

2.2.3 Etapa III. Programa para el microcontrolador PIC

Para el control de los servomotores se optó por utilizar un dispositivo también de microchip, el microcontrolador PIC16F877A. Este microcontrolador contiene los pines necesarios para controlar los 11 servomotores que conforman los 11 tendones de la prótesis y estos a su vez controlan los 11 grados de libertad con los que cuenta la prótesis.

Una vez que el dsPIC30F procese y reconozca la voz, éste enviara un código (tabla 2.2.3) por los puertos RD0, RD1, RD2, RD3, el microcontrolador recibirá el código por los puertos RC0, RC1, RC2, RC3 y se realizara la rutina programada para cada palabra.

Tabla 2.2.3 Código enviado por el dsPIC30F

Palabra	Código (Binario)	Código (decimal)	Acción
START	0001	1	Inicia Rec. de voz
CLOSE	0010	2	Cierra la mano
NO	0011	3	Mueve el Abductor del dedo índice
ONE	0100	4	Abre el dedo índice, cierra los demás
OPEN	0101	5	Abre la mano
UNKNOWN	0110	6	No hace nada

Para realizar el programa se utilizó la herramienta *Proton IDE Special Edition versión 1.1.2.8*, para mayor información del paquete consultar el sitio de internet [26].

Aunque este es un programa secundario, no por eso es menos importante para la elaboración del proyecto que involucra el reconocimiento de voz. El código que se realizó para los movimientos de la mano artificial se muestra en el Anexo IV.

2.3 PROGRAMACIÓN DEL MICROCONTROLADOR Y EL DSC (CONTROLADOR DIGITAL DE SEÑALES)

A continuación se muestran los pasos a seguir para grabar los programas en los dispositivos, tanto para el movimiento de la prótesis como para el DSC para el reconocimiento de voz.

2.3.1 Programación del Dispositivo PIC16F877A

La herramienta que se utiliza para programar el PIC16F877A se muestra en la figura 2.2.16



Fig. 2.3.1 Dispositivo para grabar el programa en el PIC

El software que se utiliza para enlazar el programa con el dispositivo para grabar el PIC es el *WinPic800 v 3.59*, este programa es de libre distribución compatible con Windows 95/98/NT/2000/ME/XP, para mayor información consultar su sitio en internet (Figura 2.3.2) [27].

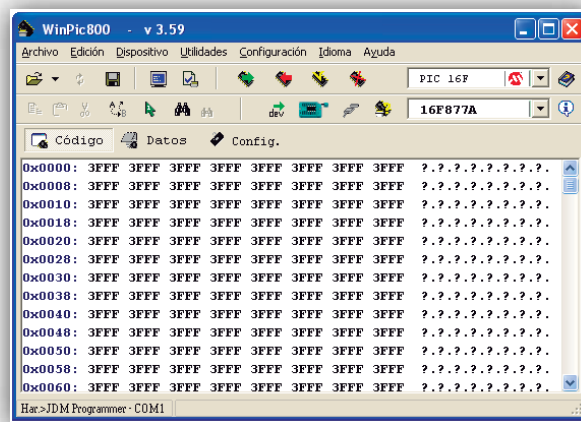


Fig. 2.3.2 Entorno del WinPic800

Para programar el PIC16F877A, una vez instalado el *winPic800*, se abre el programa *Proton IDE (Integrated Development Environment)* y se crea o carga el programa, una vez hecho esto se coloca el PIC en el socket respectivo del dispositivo para grabar *PIC's* (Circuitos integrados programables) (figura 2.3.1), se conecta el programador de *pic's* con la computadora utilizando el conector DB9, se oprime F10 o se da clic el icono que muestra la figura 2.3.3 para compilar y programar al PIC, automáticamente se abrirá *WinPic800* y se programará el PIC.

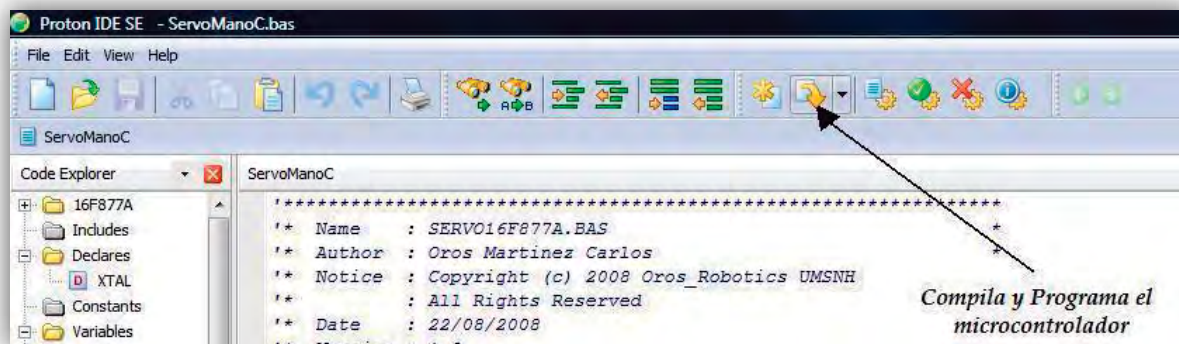


Fig. 2.3.3 Icono para programar el PIC

2.3.1.1 Instalación del WinPic800

1. Seleccionar el icono *WinPic800V3.59.exe*
2. Copiar la ruta donde se instalara, de preferencia guardar en *c:\Archivos de programa\winpic800*.
3. Configuración del Hardware: *Configuración> Hardware>* Cambiar las siguientes casillas, como lo muestra la figura 2.3.4.

Selección del Hardware	JDM Programmer
Addr	COM1
Confirmar cambios	Clic (para finalizar)

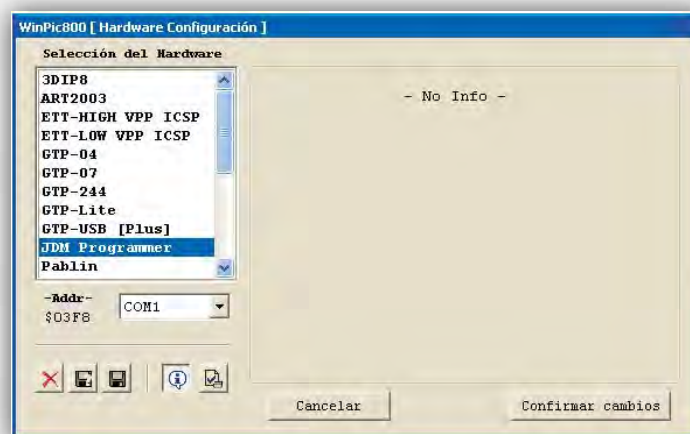


Fig. 2.3.4 Configuración del Hardware

4. Configuración del Software: *Configuración> Software> Dispositivo*, en esta casilla se marcan las dos opciones que aparecen (fig. 2.3.5).

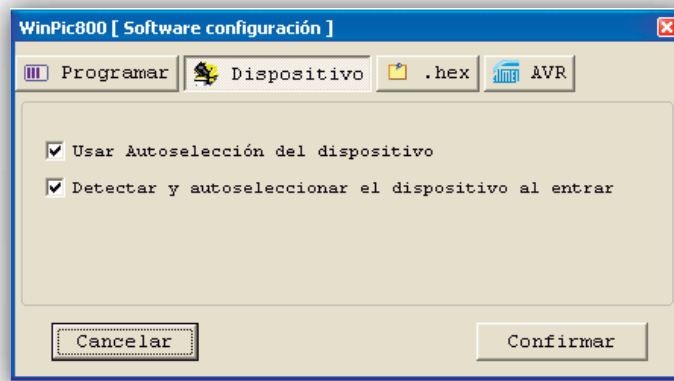


Fig. 2.3.5 Configuración del Software

- Para configurarlo con el programa *Proton IDE* (Herramienta que se utilizó para hacer el programa), en el menú *View (del Proton IDE) > Compile and Program Options > Programmer > Install New Programmer > Create a custom programmer entry*, en la ventana *Display Name* poner *WinPic800*, clic *Next* (figura 2.3.6)

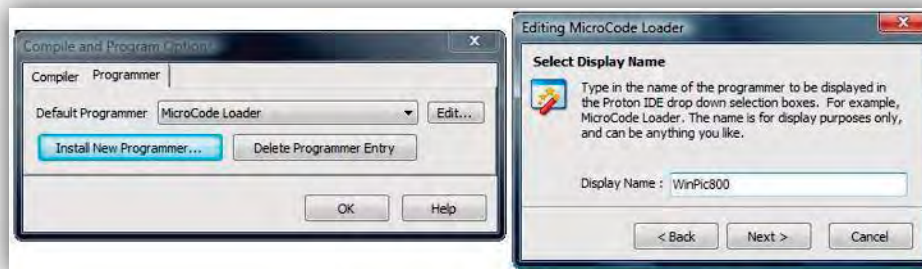


Fig. 2.3.6 Opciones para programar y compilar

- Ahora en *Programmer Filename* se pone *winpic800.exe*, clic *Next* (Figura 2.3.7)

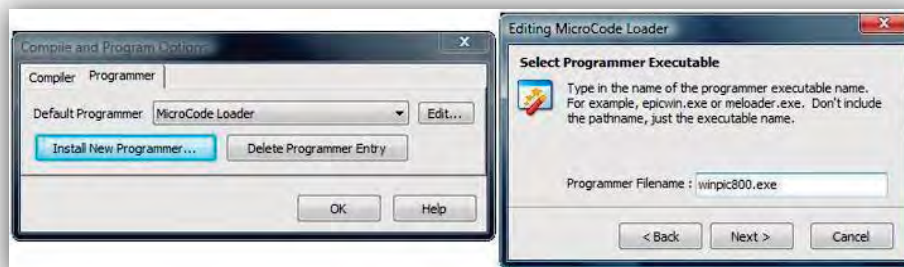


Fig. 2.3.7 Nombre del Programa ejecutable

- En búsqueda manual, ir a *C:\Archivos de Programas\WinPic800*, en parámetros poner:

Parámetros	\$hex-filename\$ -p
Y por último	Clic para finalizar

2.3.2 Programación del dsPIC30F

Una herramienta muy eficaz de microchip es el programador y depurador en circuito *MPLAB ICD-2* que además de grabar diversos microcontroladores y dsPIC depura el código desarrollado en lenguaje ensamblador o C (figura 2.3.8).



Fig. 2.3.8 MPLAB ICD-2 graba y depura en circuitos los dsPIC

Para programar el dsPIC30F6014A sobre el circuito se utiliza esta herramienta (Fig. 2.3.8), siguiendo las siguientes instrucciones.

2.3.2.1 Instrucciones para conectar el ICD2

1. Abrir el proyecto que se creó para el reconocimiento de voz, "*[Nombre del Programa].mcp*".
2. *Building the code*, Crear los archivos de salida "*[Nombre del programa].hex*" y "*[Nombre del programa].cof*", el archivo *.hex* que contiene los datos

necesarios para programar el dispositivo y el archivo *.cof* contiene información adicional que permite depurar a nivel del código fuente, esto se hace con el siguiente menú:

- En la barra de menú ir a *Project>Build Options...>Project*, en la pestaña de *Directories and search Paths*, buscar *Include Search Path* y poner la siguiente dirección, como lo muestra la figura 2.3.9.

C:\Program Files\Microchip\MPLAB ASM30 Suite\Soport\dsPIC30F\Inc

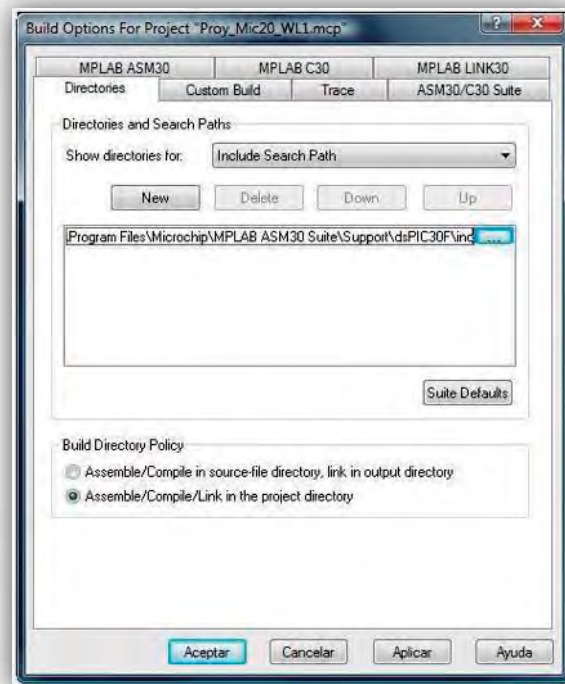


Fig. 2.3.9 Configuraciones para construir el proyecto

3. Enlace para el ICD-2, en la misma barra de herramientas que en el punto anterior, seleccionar la pestaña *MPLAB LINK30*, seleccionar *Link for ICD2*, clic en aplicar y después en aceptar.
4. Construir el proyecto, Seleccionar el menú *Project>Build All*, esperar hasta que salga la ventana con la leyenda *BUILD SUCCEEDED* (Fig. 2.3.10).

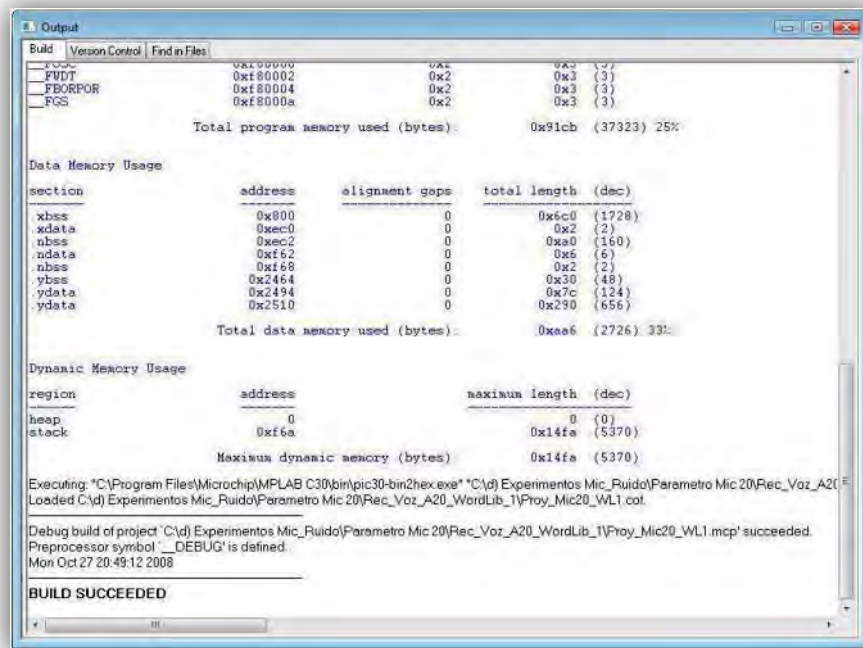


Fig. 2.3.10 Construcción del proyecto

- Configuración del dispositivo, en el menú *configure>Configuration Bits*, cambiar las siguientes configuraciones:

Oscillator	XT w/PLL 4x
Watchdog Timer	Disabled

- Configuración de la comunicación a la PC a través del USB, instalar el driver para el ICD2 según la siguiente dirección:

c:\program Files\Microchip\MPLAB IDE\ICD2\Drivers\ddicd2.htm

El nombre del documento ".html" es "ddicd298.htm" para windows 98, "ddicd2me.htm" para windows ME y "ddicd2.htm" para windows 2000 y XP.

Nota: No dejar que Windows instale el driver, el usuario debe seleccionar la ruta de éste.

- Conexión del *MPLAB ICD2*, conectar el *ICD2* a la PC con el USB, conectar el *ICD2* al circuito del dsPIC30F, con el *short Rj-11 (telephone)* y por último aplicar energía.

8. Habilitar la conexión para el *MPLAB ICD2*, seleccionar en el menú:

- *Debugger> Select Tool> MPLAB ICD2*
- *Debugger> Settings> Program> Allow ICD2 to select memories and ranges (seleccionarla)*
- *Debugger> Connect> MPLAB ICD2 Ready*

9. Programación de los parámetros para el dispositivo *dsPIC30F*:

- *Programmer> Select Programmer> MPLAB ICD2*
- *Programmer> Settings> Program> Allow ICD2 to select memories and ranges seleccionarla, (Fig. 2.3.11).*
- *Programmer> Connect> MPLAB ICD2 Ready*

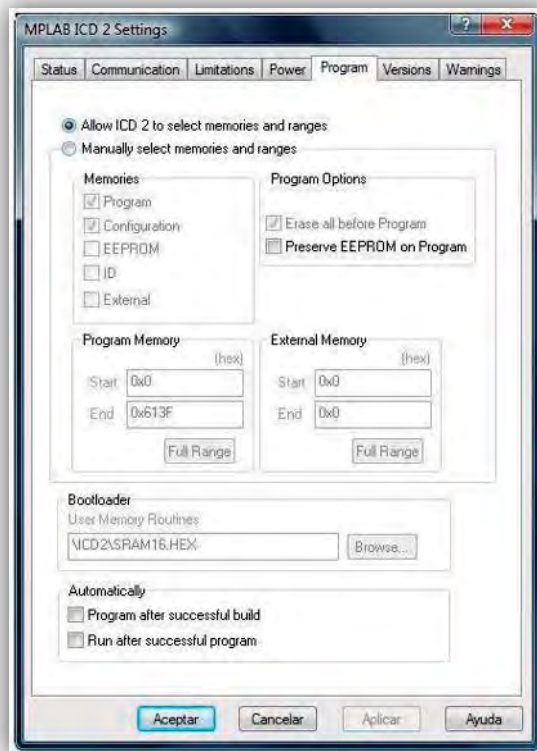


Fig. 2.3.11 Configuración para programar

Nota: Nunca se debe seleccionar al mismo tiempo la casilla *Debugger> select Debugger> MPLAB ICD2* con la de *Programmer> Select Programmer> MPLAB ICD2*.

10. Programar el dispositivo *dsPIC30F*: por último se selecciona la pestaña *Programmer>Program*, esperar a que finalicé y listo, dispositivo programado.

2.4 CONCLUSIONES DEL CAPÍTULO II

Al inicio del capítulo se describen los tipos de reconocimiento de voz que existen, el reconocimiento discreto del habla es el tipo al que pertenece este proyecto ya que el dispositivo utilizado necesita de por lo menos medio segundo de separación entre las palabras para poder reconocerlas.

Si bien el DSC no puede ser comparado con un DSP, la velocidad a la que reconoce las palabras es bastante buena (medio segundo aproximadamente) y como el DSC pertenece a la familia de Microchip, su programación es un poco similar a la de los microcontroladores con los que se había trabajado en proyectos anteriores por lo que aprender a programar el DSC no fue tan complicado.

CAPÍTULO III: DISEÑO DEL CIRCUITO ELECTRÓNICO

Este capítulo inicia con la descripción de los componentes necesarios para construir el circuito electrónico del reconocimiento de voz (etapa II), después muestra los componentes para la creación del circuito para el control de mano artificial (etapa III), por último se muestran los esquemas y diagramas para construir las tres etapas en las que se dividieron los circuitos electrónicos.

La primera etapa es un módulo tipo socket diseñado para hacer más fácil la conexión del dsPIC30F con el circuito diseñado para el reconocimiento de voz y para poder separarlos en caso de que el dsPIC se dañe y se tenga que sustituir por otro. La segunda etapa es la encargada del reconocimiento de voz, convirtiendo las órdenes sonoras (palabras) en señales digitales. La tercera etapa es transformar las señales digitales en acciones realizadas por servomotores encargadas de mover la mano artificial.

3.1 COMPONENTES PARA LOS CIRCUITOS ELECTRÓNICOS

3.1.1 Componentes del hardware para el dsPIC30F

La tabla 3.1 muestra los materiales para la construcción del módulo *PLUG-IN* tipo socket (etapa I), que se conecta a la segunda etapa.

Tabla 3.1 Elementos del circuito para el dsPIC30F

Símbolo	Material	Cantidad	
-----	dsPIC30F6014A-20E/PT	1	
-----	Soldadura 0.8 mm aleación 63/37	3 metros	
-----	Marcador permanente	1	
-----	Cloruro férrico (botella de 220 ml)	1	
-----	Lija para metal 300 ó 400	1	
-----	Baquelita dimensiones 5x5 cm	2	
J14, J15	Tira de pines pitch 2.54 cm	Normal	2
J1, J2, J3, J4		Doble	4
-----	Punta para soldar de 0.8 mm	2	

3.1.2 Componentes del hardware para el reconocimiento de voz

La tabla 3.2 muestra los materiales necesarios para la construcción del circuito electrónico, para la segunda etapa que se encarga del reconocimiento de voz.

Tabla 3.2 Elementos del circuito para reconocimiento de voz

Símbolo	Material		Cantidad
-----	Módulo dsPIC30F6014A-20E/PT (de la etapa I)		1
SI3000	Decodificador de voz SI3000		1
X1	Cristal HC-49US de 7.3728 Mhz		1
DILOSC	Oscilador Canned Dil-14 de 6.144 Mhz		1
DIL OSC	Socket para oscilador 14 pines		1
J14, J15	Tira de pines pitch 2.54 cm	Normal	2
J1, J2, J3, J4		Doble	4
-----	Punta para soldar de 0.8 mm		2
-----	Soldadura 0.8 mm aleación 63/37		1 rollo
-----	Cloruro férrico (botella de 220 ml)		1
-----	Lija para metal 300 ó 400		1
-----	Alambre calibre 22 (rojo/negro)		2 metros
Reset y Codec	Push botón 2 terminales		2
R6, R7	Resistencias	30 ohm, 1/16w	4
R1, R11		100 ohm, 1/2w	2
R8, R9		470 ohm, 1/2w	4
R10		1K ohm, 1/2w	2
R5		2.2k ohm, 1/4w	2
R2, R3		10K ohm, 1/2w	4
1, C2		Capacitor cerámico 18 pf, 50v	
C3, C5, C8, C9	Capacitor Electrolítico de aluminio de 10uf, 16v, +_20%		6
C4, C6, C7, C10	Capacitor cerámico SMD (SMT) 0805 de 0.1uf, 50v		6
C11, C12	Capacitor Electrolítico 100uF 16v		2
RJ-12 ICD2	Conector de teléfono RJ-12 jack Pinout		1
LG, LR	Led (Rojo, Verde) de 3 mm		4
Mic In, SPKR Out	PCB Mounted 3.5mm stereo jack socket		2
TB1	PCB Terminal Block 4 tornillos		1
-----	Batería de 9 volts		1
Batery Holder	Socket para batería de 9 volts		1
JMP1	Jumper (Hembra/Macho) 3 Pines		2
7805CV	Regulador 7805CV		1
-----	Baquelita	5x5 cm	2
		10x15 cm	2
		10x10 cm	2

Descripción de los principales elementos que configuran la segunda etapa.

- Un decodificador de voz *SI3000 (DCI Si3000 Voiceband Codec with Microphone/Speaker Drive)*, éste recibe la voz a través del micrófono y la transmite al dsPIC.
- Un Microcontrolador avanzado dsPIC (*DSC-controlador digital de señal*) dsPIC30F6014A de uso general, contiene el programa desarrollado para el reconocimiento de voz y las órdenes para accionar la prótesis.
- Un conector ICD2 RJ12 *Jack Pinout* para programar el dsPIC una vez instalado en el circuito.
- Una batería de 9 volts que dotará de energía a los elementos electrónicos.
- Los componentes básicos para el funcionamiento del dsPIC y el DCI, resistencias, capacitores, cristales, etc.

3.1.3 Componentes del hardware para el control de la mano artificial

La tabla 3.3 muestra los elementos necesarios para la elaboración del circuito que estará a cargo del movimiento de la mano artificial (tercera etapa).

Tabla 3.3 Elementos para control de prótesis

Símbolo	Nombre elemento		Cantidad
U1	PIC16F877A-I/P		1
TB1	PCB Terminal Block 4 Tornillos		1
J1-J11	Tira de pines Pitch 2.54 cm		4
RESET	Push Botón de 2 terminales		1
U1	Socket Para circuito integrado 40 pines		1
C3	Capacitor Electrolítico 100uF 16volts		2
C1, C2	Capacitor cerámico 22pF 16 volts		2
7805CV	Regulador de voltaje 7805CV		1
X1	Cristal	20 MHz	1
D1	Led de 3mm	Rojo	2
D2		Verde	2
D3		Azul	2
D4		Amarillo	2
R1	Resistencias	100 ohm	2
R3,R4,R5,R6		470 ohm	6
R2		10k ohm	2
JMP1	Jumper (Hembra/Macho) 3 Pines		2

3.2 ETAPA I. MÓDULO PLUG-IN PARA EL dsPIC30F6014A

Para la elaboración de los dibujos y esquemas para las tres etapas, se utilizó el programa *Proteus 6 Profesional*, el cual contiene las herramientas *ISIS* y *ARES Profesional*, propiedad de *Labcenter Electronics*, se utilizó una versión de prueba, para mayor información de este programa consultar su sitio en internet [28].

Se crearon dos módulos diferentes **A** y **B** para el dsPIC30F, el **módulo A** se muestra en las figuras 3.1, 3.2, 3.3, cuyas dimensiones son 4.5 x 4.5 cm, a este módulo se le conectan las 80 patitas (las figuras 3.1, 3.2, y 3.3 muestran el tamaño real de los módulos).

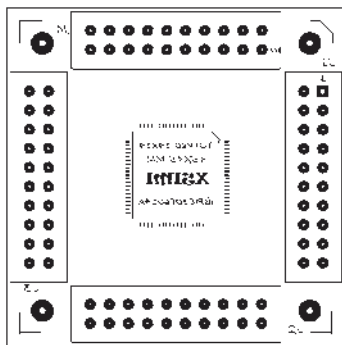


Fig. 3.1 DSPIC TQFP80 Elementos

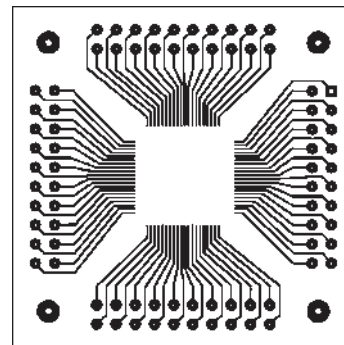


Fig. 3.2 DSPIC TQFP80 Pistas

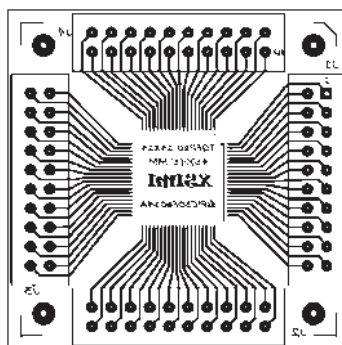


Fig. 3.3 DSPIC TQFP80 Integración

El **módulo B**, es mucho más sencillo, ya que solo cuenta con la conexión a las patitas utilizadas para el reconocimiento de voz y para su programación, las dimensiones de éste son 4 x 3.5 cm, las figuras 3.4, 3.5, 3.6 muestran los diferentes

esquemas de este módulo, los tamaños mostrados en las figuras son los reales del dispositivo.

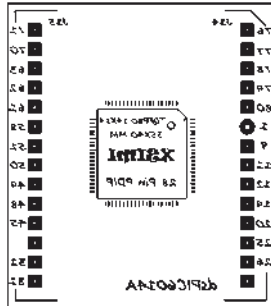


Fig. 3.4 DSPIC PDIP28 Elementos

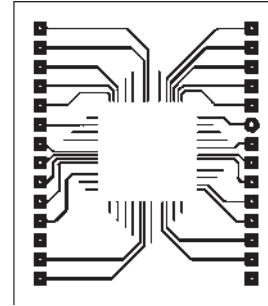


Fig. 3.5 DSPIC PDIP28 Pistas

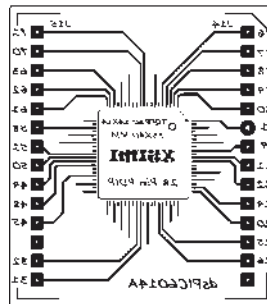


Fig. 3.6 DSPIC PDIP28 Integración

3.3 ETAPA II. CIRCUITO PARA RECONOCIMIENTO DE VOZ

Ya que se crearon dos módulos para el dsPIC30F, también se crearon sus respectivos circuitos, las figuras 3.7, 3.8, 3.9, 3.10 y 3.11 muestran los diferentes esquemas para el **módulo A** de 80 patitas.

Al igual que en la etapa I, los dibujos y esquemas que se presentan en la etapa II son los tamaños reales que tendrán los dispositivos. Dimensiones de esta placa 11 x 9 cm.

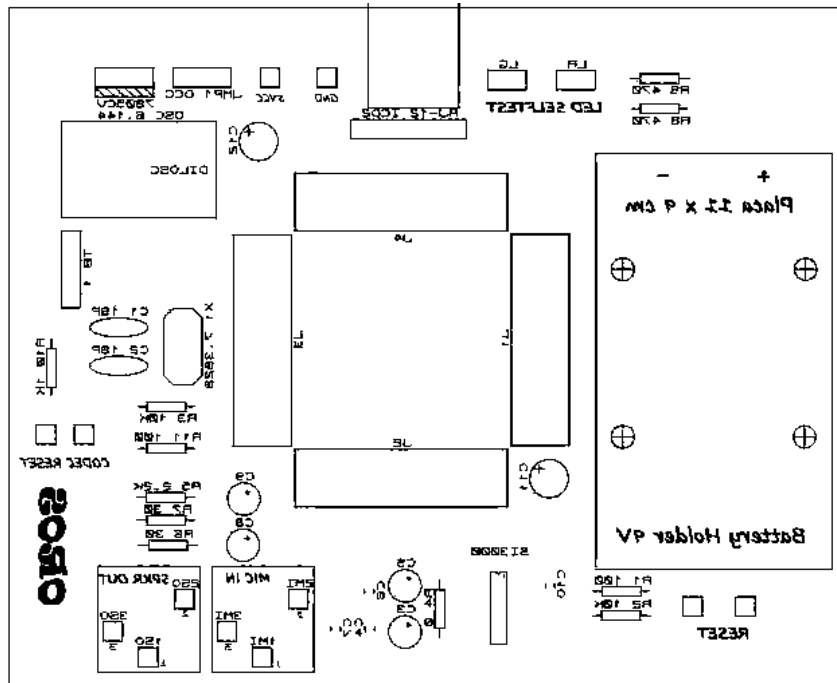


Fig. 3.7 Circuito de voz TQFP80 Elementos

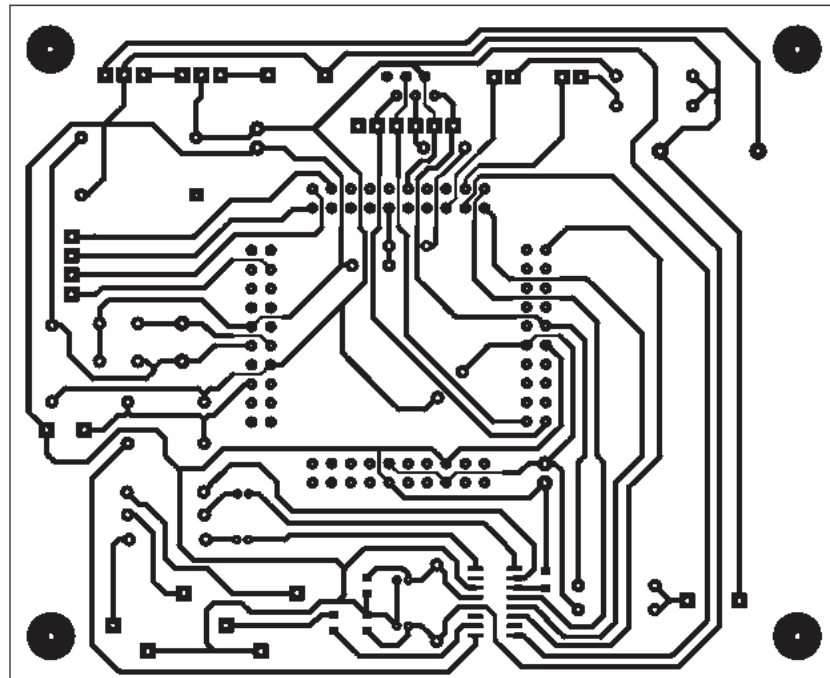


Fig. 3.8 Circuito de voz TQFP80 Pistas

Circuito para el **módulo B**, Las figuras 3.12, 3.13, 3.14, 3.15 y 3.16 muestran los dibujos y esquemas para el módulo II. Las dimensiones de la placa son 11 x 8.2 cm.

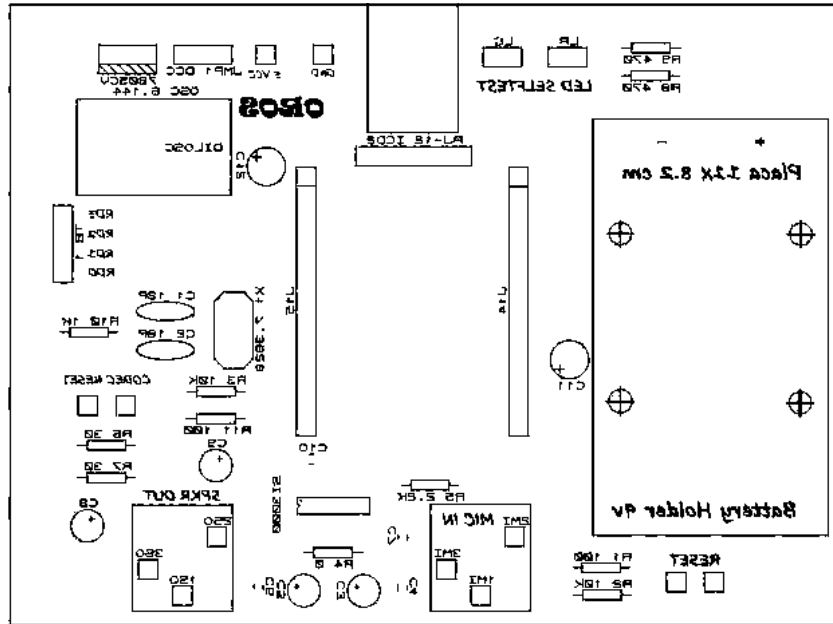


Fig. 3.12 Circuito de voz PDIP28 Elementos

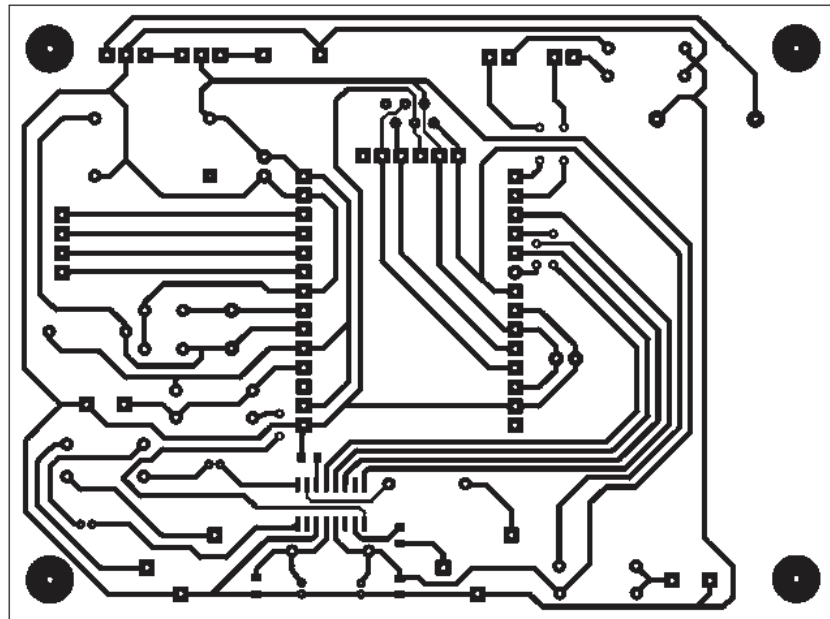


Fig. 3.13 Circuito de voz PDIP28 Pistas

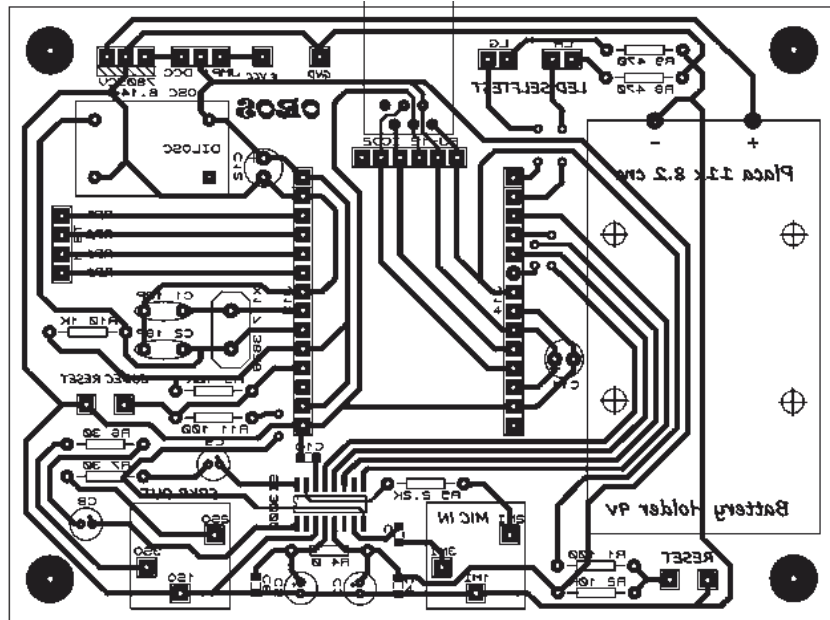


Fig. 3.14 Circuito de voz PDIP28 Integración

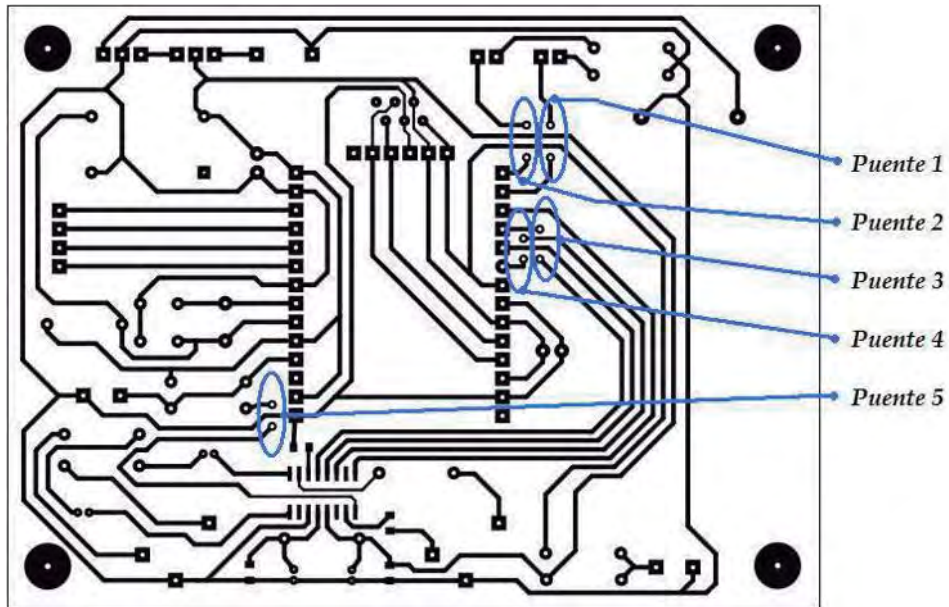


Fig. 3.15 Circuito de voz PDIP28 Puentes

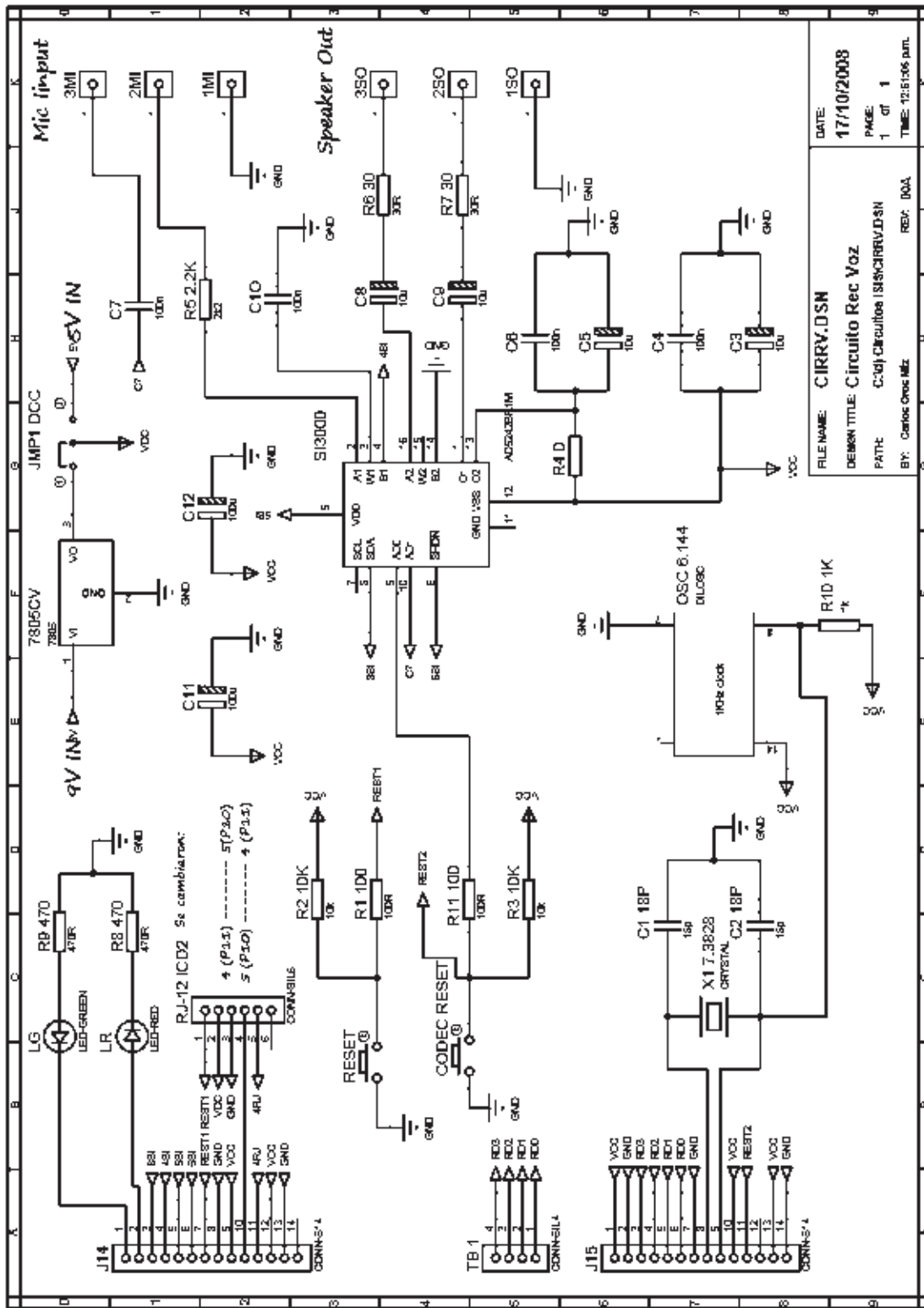


Fig. 3.16 Circuito de voz PDIP28 Esquemático

3.4 ETAPA III. CIRCUITO PARA EL CONTROL DE LA MANO ARTIFICIAL

Este diseño utiliza el PIC16F877A, las dimensiones de esta placa son de 6.5 x 7 cm, y es mucho más sencillo que el utilizado para el reconocimiento de voz, ya que los microcontroladores no son tan sofisticados como el dsPIC, que necesita de más elementos electrónicos para su funcionamiento.

Las figuras 3.17, 3.18, 3.19, 3.20 y 3.21 muestran los esquemas y dibujos (tamaños reales) para la fabricación del circuito.

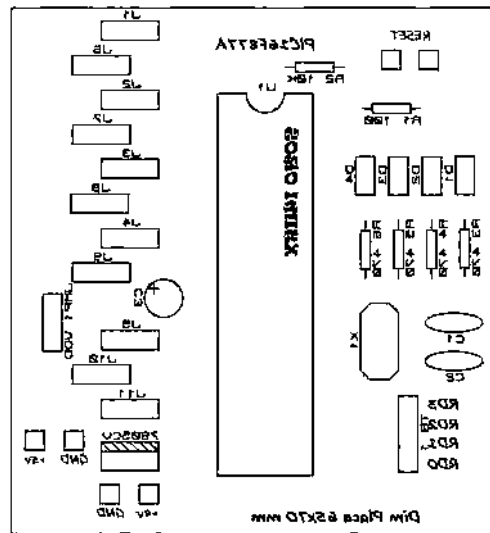


Fig. 3.17 Control mano artificial elementos

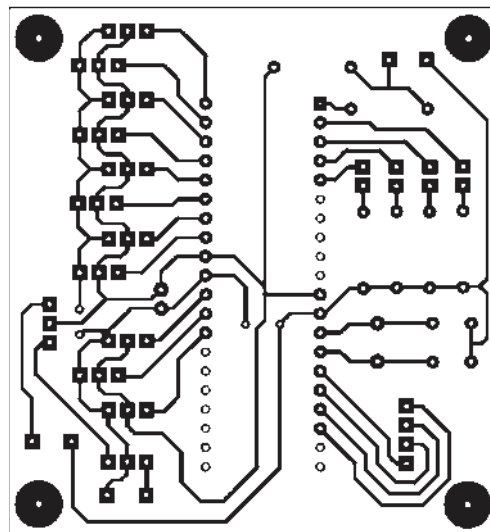


Fig. 3.18 Control mano artificial pistas

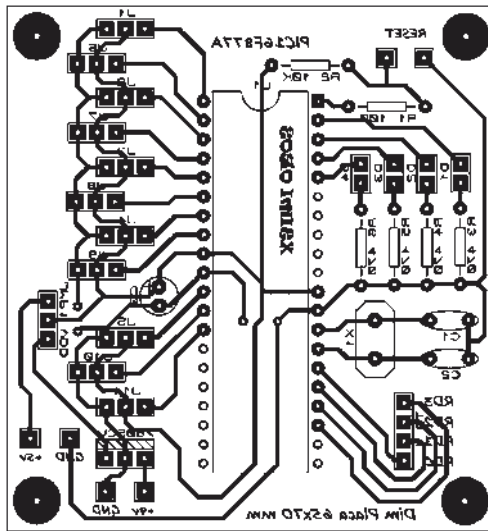


Fig. 3.19 Control mano artificial integración

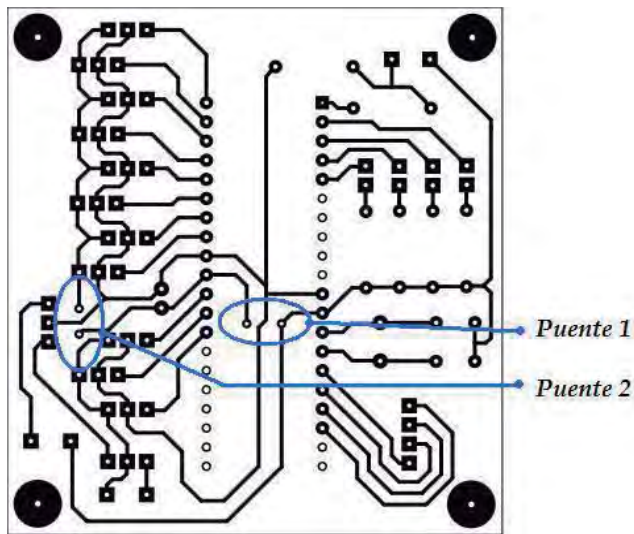


Fig. 3.20 Control mano artificial Puentes

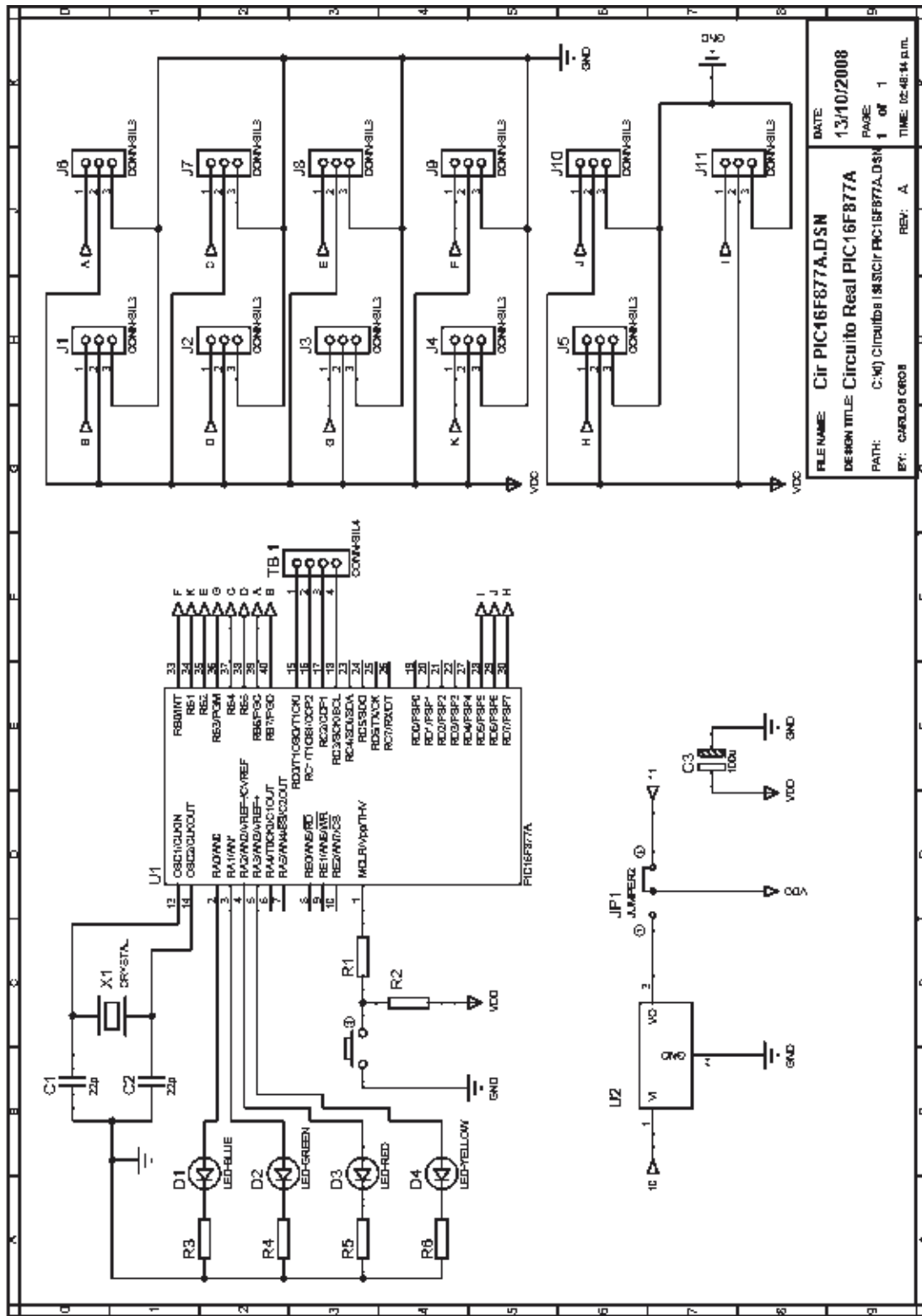


Fig. 3.21 Control mano artificial esquemático

3.5 CONCLUSIONES DEL CAPÍTULO III

Se utilizaron dos etapas para el reconocimiento de voz ya que el encapsulado del DSC no permite colocar un socket para intercambiar el dispositivo en caso de daño, por lo que se diseñó un módulo Plug-in para así poder colocar el socket, el socket se observa en la figura 3.7 en los elementos J1, J2, J3 y J4 para el circuito TQFP80 y en la figura 3.12 el socket son los elementos J12 y J14.

Los circuitos diseñados en las etapas II y III no se lograron crear a la perfección, se necesitaran colocar puentes para unir las líneas de cobre que por lo complejo de los circuitos no se lograron unir, tal y como lo muestran las figuras 2.10, 3.15 y 3.20.

CAPÍTULO IV: EXPERIMENTOS Y RESULTADOS

Este capítulo inicia con la información requerida para poder entender que es y cómo se realiza una toma de muestra para poder realizar un análisis estadístico. Una vez interpretado esto, se calcula el tamaño de muestra para poder estimar los parámetros necesarios para mejorar el reconocimiento de voz. También se estudia el impacto que tiene la selección del micrófono, afecta o no la sensibilidad de éste y por último la eficiencia del sistema en ambientes con y sin ruido.

4.1 TAMAÑO DE MUESTRA PARA EL RECONOCIMIENTO DE VOZ

4.1.1 Nivel de precisión deseado

Los resultados de las encuestas siempre están sujetos a cierta incertidumbre debido a que sólo una parte de la población ha sido investigada y a los errores de medición. Esta incertidumbre puede ser reducida tomando muestras más grandes y usando métodos de medición adecuados, métodos como el muestreo aleatorio simple, el muestreo simple con remplazo, el muestreo simple sin remplazo, el muestreo estratificado y/o el muestreo por racimos. Desgraciadamente esto, por lo general, cuesta tiempo y dinero. En consecuencia, es un paso importante la especificación del nivel de precisión deseado. Este paso es la responsabilidad de la persona que va a usar los datos.

4.1.2 Límites de confianza

El símbolo t es la variable estándar de la distribución *t de student* correspondiente a la probabilidad del nivel de confianza deseada, los valores más comunes son:

Tabla 4.1 Probabilidad del nivel de confianza deseada

Probabilidad del nivel de confianza (%)	50	80	90	95	99
Valor de t	0.67	1.28	1.64	1.96	2.58

Si el tamaño de muestra es menor a 60, los intervalos de confianza pueden ser tomados de la tabla *t de student* con $(n-1)$ grados de libertad, siendo éstos los grados de libertad en la varianza estimada S^2 . La distribución *t de student* se ajusta exactamente sólo si los datos están normalmente distribuidas y “N” es infinito (N es el número total de la población que será investigada).

4.1.3 Selección de la muestra

Existe, actualmente, una gran variedad de métodos para seleccionar una muestra. Por cada plan que es considerado, se puede hacer estimaciones del tamaño de la muestra, partiendo de un conocimiento del nivel de precisión deseada. Los costos relativos y el tiempo involucrado para cada plan son también comparados antes de tomar una decisión.

En una encuesta por muestreo se eligen ciertas propiedades que se intentan medir y registrar para cada unidad que venga dentro de la muestra. Estas propiedades de la muestra son identificadas como características o más simplemente, atributos.

En la planeación de una encuesta por muestreo, siempre se alcanza un punto en el que se debe tomar una decisión acerca del tamaño de la muestra. La decisión es importante. Ya que una muestra demasiado grande implica un desperdicio de recursos y una muestra demasiado pequeña disminuye la utilidad de los resultados.

4.1.4 Fórmula para n (tamaño de muestra) en el muestreo para proporciones

Las unidades son clasificadas en dos clases C y C' . Se ha convenido en algún margen de error d en la proporción estimada p de unidades pertenecientes a la clase C y hay un pequeño riesgo " α ", que se está dispuesto a correr, de que el error real sea mayor que d ; es decir, se quiere:

$$\Pr(|p - P| \geq d) = \alpha$$

Se supone un *muestreo simple aleatorio* y se considera que p se distribuye normalmente. La fórmula para n (tamaño de la muestra) en el muestreo para proporciones está dada como:

$$n = \frac{\frac{t^2 PQ}{d^2}}{1 + \frac{1}{N} \left(\frac{t^2 PQ}{d^2} - 1 \right)} \quad (4.1)$$

Para uso práctico se sustituye un estimador adelantado p de P . Si " N " es grande, una primera aproximación es:

$$n_0 = \frac{t^2 pq}{d^2} = \frac{pq}{V} \quad (4.2)$$

Donde: $V = \frac{d^2}{t^2} = \text{Varianza deseada de la proporción muestral}$

En la práctica primero se calcula n_0 . Si n_0/N es despreciable, n_0 es una aproximación satisfactoria del tamaño de la muestra (n) de la ecuación (4.1). Si no es así, es evidente de la comparación de (4.1) y (4.2), que n se obtiene como sigue:

$$n = \frac{n_0}{1 + (n_0 - 1)/N} = \frac{n_0}{1 + (n_0/N)} \quad (4.3)$$

La fórmula para n_0 es válida también si d , p , y q están todas expresadas como porcentajes en lugar de proporciones. Puesto que el producto pq aumenta conforme p tiende a $1/2$, o 50%, un estimador conservador de n se obtiene escogiendo para p el valor más cercano a $1/2$ dentro del intervalo de valores que se piensa puede tomar p . Por ejemplo si parece probable que p tome valores entre 5 y 9%, suponemos 9% para la estimación de n [29].

4.1.5 Estimación del tamaño de muestra (n) para el reconocimiento de voz

La Proporción estimada se espera entre 94% y 98%, por lo que se toma el más cercano al 50%. De la tabla 4.1 para una seguridad del 95% se encuentra $t = 1.96$

Los datos necesarios para utilizar la ecuación 4.2 son:

$d =$ Margen de error = $\pm 3\% = 0.03$

$p =$ Proporción estimada = 94%

$q = 1 - p = 0.06$

Sustituyendo los datos en la ecuación 4.2

$$n_0 = \frac{(1.96)^2(0.94)(0.06)}{(0.03)^2} = \frac{(3.8416)(0.0564)}{0.0009} = \frac{0.21666624}{0.0009} = 240.74$$

Suponiendo que las palabras son $N=1000$. Se encuentra el tamaño de muestra (n) para un número finito.

$$n = \frac{240.74}{1 + \left(\frac{240.74}{1000}\right)} = \frac{240.74}{1.24074} = 194.03$$

$n = 194$

Cerca del 20 % de un total de 1000 palabras debe ser muestreado con el fin de obtener la precisión deseada.

4.2 CARACTERÍSTICAS DE LOS MICRÓFONOS

Se utilizaron tres micrófonos para tomar las muestras, en un ambiente con y sin ruido: Las tablas 4.2, 4.3 y 4.4 muestran las características de estos micrófonos utilizados.

Tabla 4.2 Características del micrófono Logitech

Micrófono A		
<i>Logitech ClearChat Style</i> , el micrófono cuenta con supresión de ruido filtra y elimina ruidos de fondos no deseados		
Marca	Logitech	
Modelo	981-000043	
Potencia (sensibilidad)	Audífono	-59 dBV/ μ Bar
	Micrófono	-39 dBV/Pa +/- 4dB
Frecuencia	Audífono	20-20 000 Hz
	Micrófono	100-10 000 Hz
Modo de sensibilidad MIC20 y MIC30, para el reconocimiento de voz		

Tabla 4.3 Características del micrófono Plantronics

Micrófono B		
<i>Voice Gaming-Music</i> , micrófono con supresión de ruido filtra y elimina ruidos no deseados		
Marca	Plantronics	
Modelo	.AUDIO 345	
Potencia (sensibilidad)	Audífono	106 dBpl/Mw +/- 4 dB
	Micrófono	-39 dB/Pa +/- 5 dB
Frecuencia	Audífono	20-20kHz
	Micrófono	100-8kHz
Modo de sensibilidad MIC20 y MIC30, para el reconocimiento de voz		

Tabla 4.4 Características del micrófono Acteck

Micrófono C		
Audífono - Mic Acteck p/Oído		
Marca	Acteck	
Modelo	HF-400 Manos Libres Audífono c/Micrófono	
Potencia (sensibilidad)	Audífono	-98 dB +/- 4 dB
	Micrófono	-98 dB +/- 3 dB
Frecuencia	Audífono	20-20kHz
	Micrófono	20-20kHz
Modo de sensibilidad MIC20 y MIC30, para el reconocimiento de voz		

4.3 METODOLOGÍA PARA EL MUESTREO

Aunque el dispositivo reconoce las palabras con el micrófono A, B, o C, ya sea en el modo de sensibilidad de 20 ó 30, se realizó un muestreo para obtener los mejores parámetros, tales como:

- Seleccionar el mejor modo de sensibilidad
- Seleccionar el mejor ruido predefinido
- Conocer y probar la eficiencia
- Seleccionar y conocer las diferencias entre los distintos micrófonos

Se realizaron 16 experimentos por micrófono (ver las figuras 4.1, 4.1 y 4.3). Para cada experimento se tomaron 194 muestras por palabras analizando un total de 5 palabras por experimento:

- Palabra 1: Start (Keyword)
- Palabra 2: close
- Palabra 3: No
- Palabra 4: One
- Palabra 5: Open

Por último se muestran los mejores resultados obtenidos con y sin ruido.

Diagrama 4.1 Muestras tomadas para el micrófono A

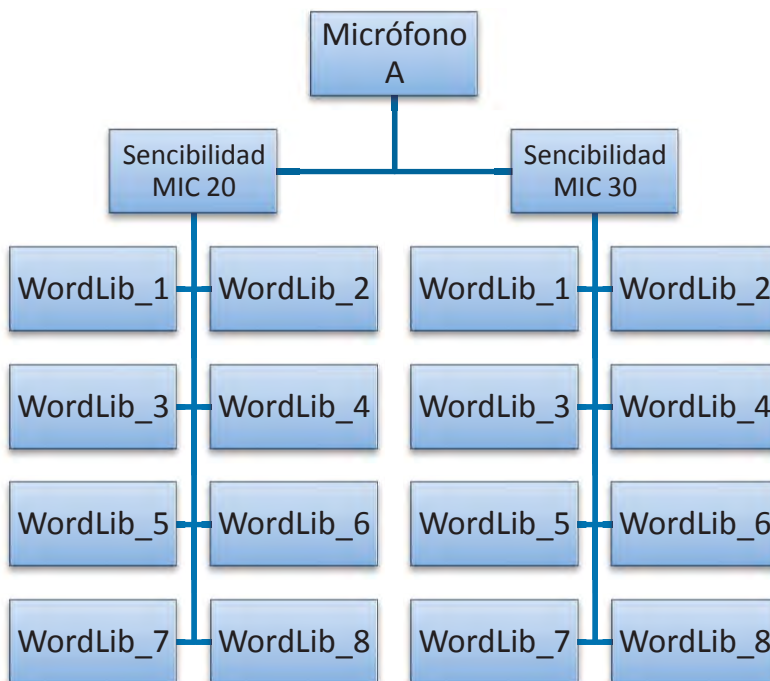


Diagrama 4.2 Muestras tomadas para el micrófono B

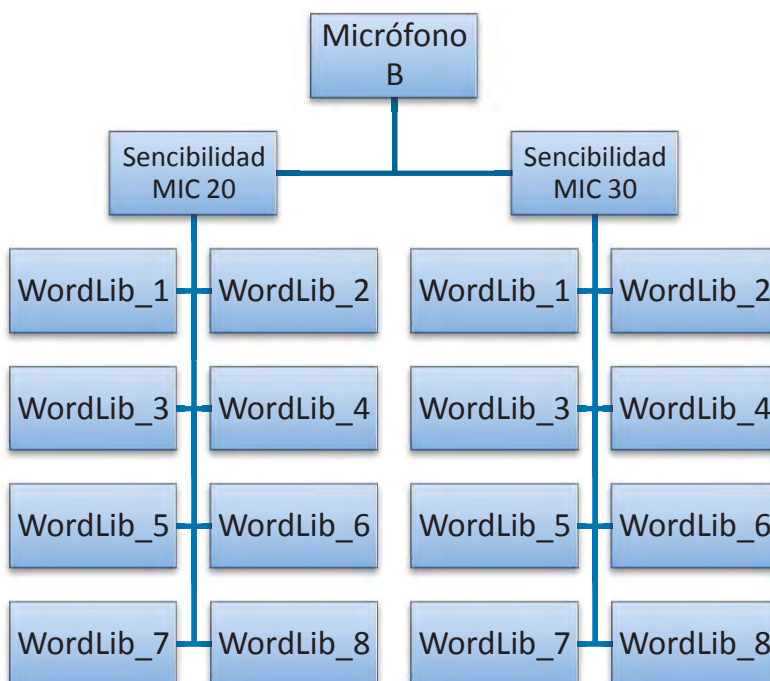
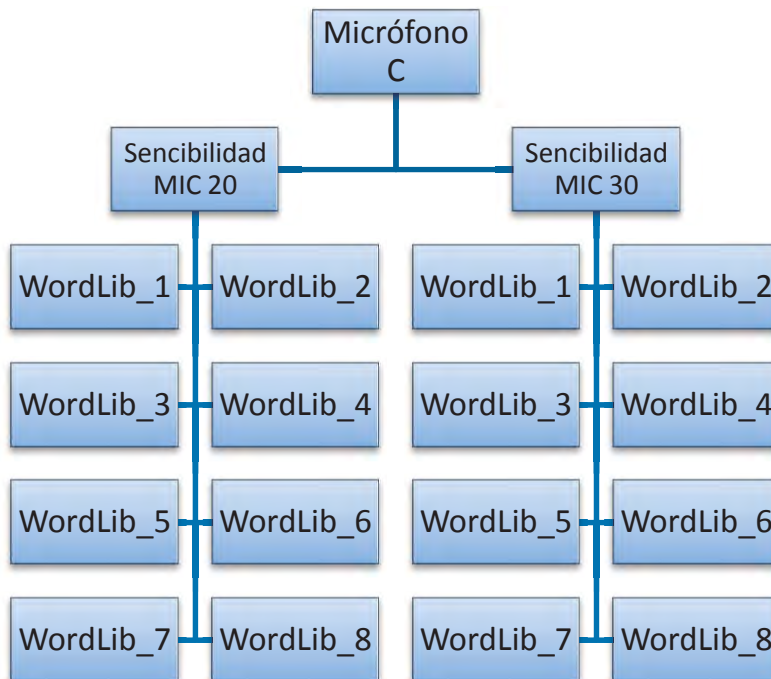


Diagrama 4.3 Muestras tomadas para el micrófono C



Las bibliotecas (*WordLib_n*) contienen el perfil de ruido predefinido que se utiliza, como filtro para reducir el ruido del medio donde se desarrolla el reconocimiento de voz y así obtener mejores resultados. Las características de las bibliotecas se muestran en la siguiente tabla 4.5 [24]:

Tabla 4.5 Perfiles de ruido

Biblioteca	Perfil Ruido (dB)	Nivel SNR (dB)	Tipo de Ruido (%)		
			White	Car	Office
WordLib_1	White - 50 dB	50	100	0	0
WordLib_2	White - 15 dB	15	100	0	0
WordLib_3	Car - 15 dB	15	0	100	0
WordLib_4	Office - 15 dB	15	0	0	100
WordLib_5	White/Car -15 db	15	50	50	0
WordLib_6	White/Office - 15 dB	15	50	0	50
WordLib_7	Car/Office - 15 dB	15	0	50	50
WordLib_8	Mixed - 15 dB	15	34	33	33

4.4 RESULTADOS

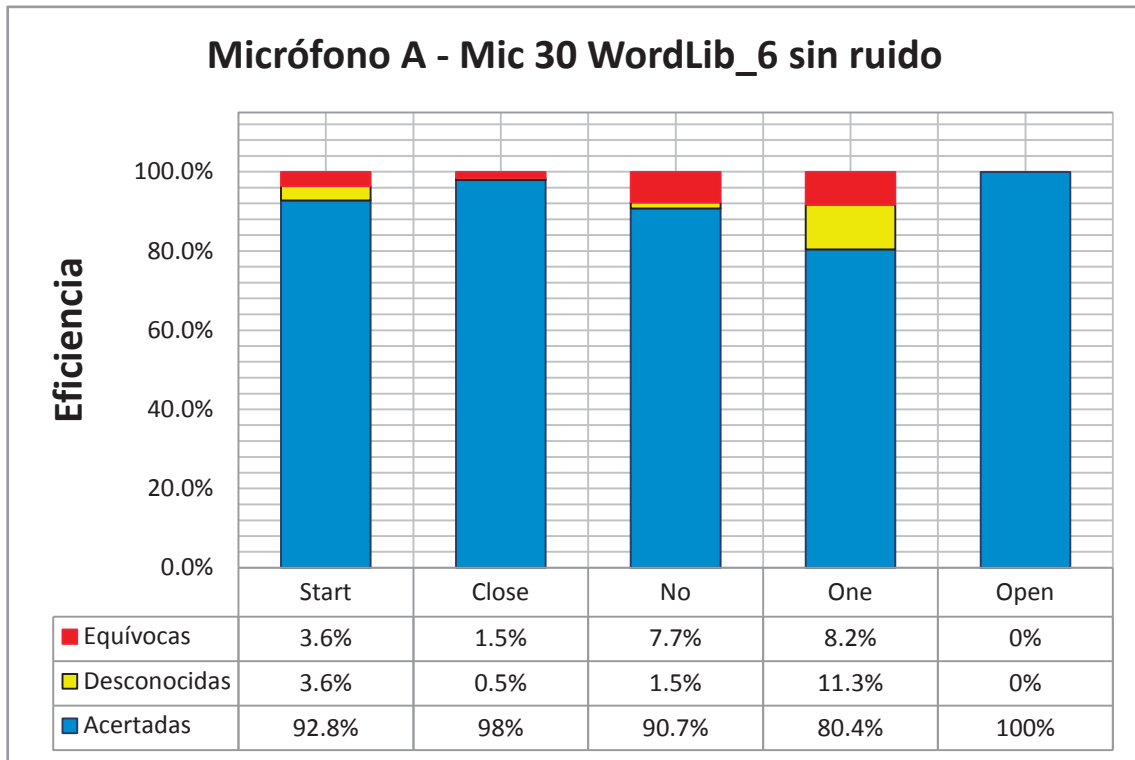
A continuación se muestran las gráficas con los mejores resultados obtenidos para cada micrófono en ambientes con y sin ruido externo.

Micrófono A

El mejor resultado se obtuvo con la biblioteca *WordLib_6* para una sensibilidad de *MIC 30*, en un ambiente sin ruido, de acuerdo a los resultados se diseñaron dos gráficas, la primera (gráfica 4.1) indica la eficiencia de cada palabra, así como los dos diferentes errores que obtuvo el sistema:

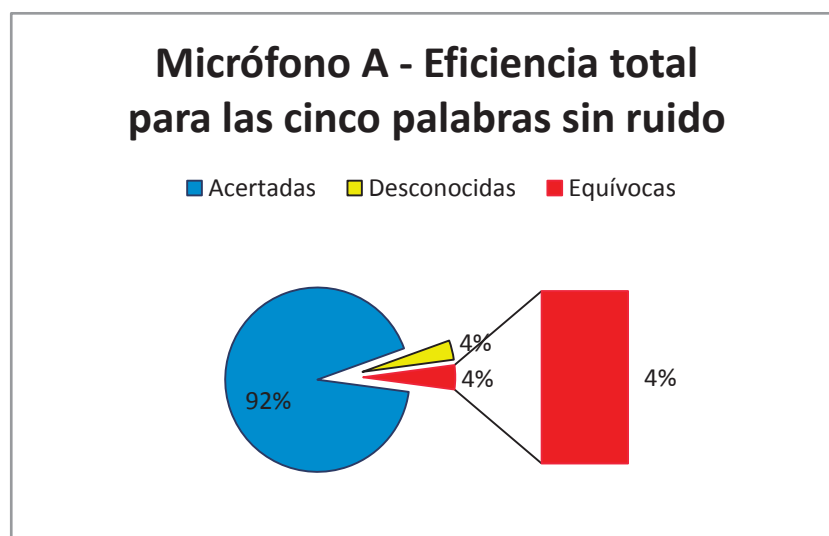
- *Equívoca* cuando el dispositivo confunde la palabra con alguna otra de las que se encuentra en la biblioteca.
- *Desconocida* cuando el dispositivo no entendía la palabra tomada por el micrófono.

Gráfica 4.1 Micrófono A - Eficiencia por palabra sin ruido



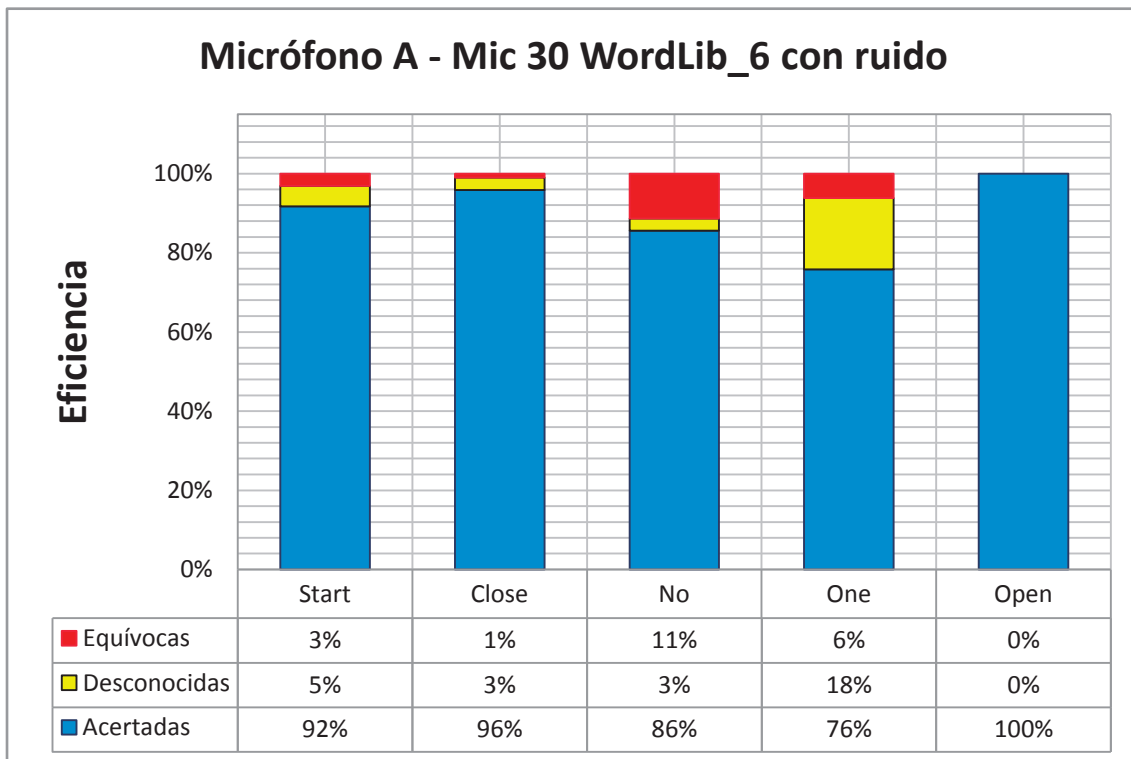
La gráfica 4.2 muestra la eficiencia total de la biblioteca *WordLib_6* para las 5 palabras.

Gráfica 4.2 Micrófono A - Eficiencia *WordLib_6* para MIC 30 sin ruido

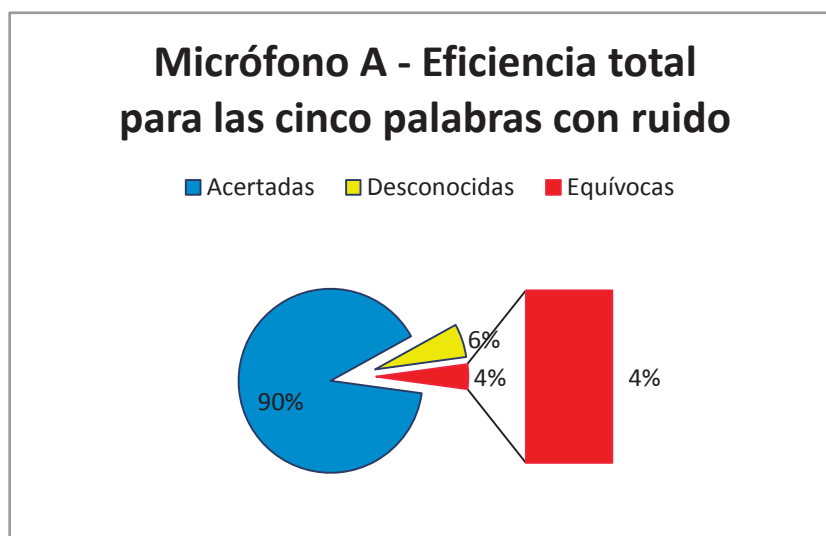


Se tomo una segunda muestra pero ahora con un ambiente ruidoso para ver el efecto que éste provocaba en el reconocimiento. La gráfica 4.3 muestra el comportamiento por palabra y la gráfica 4.4 la eficiencia total con ruido.

Gráfica 4.3 Micrófono A - Eficiencia por palabra con ruido



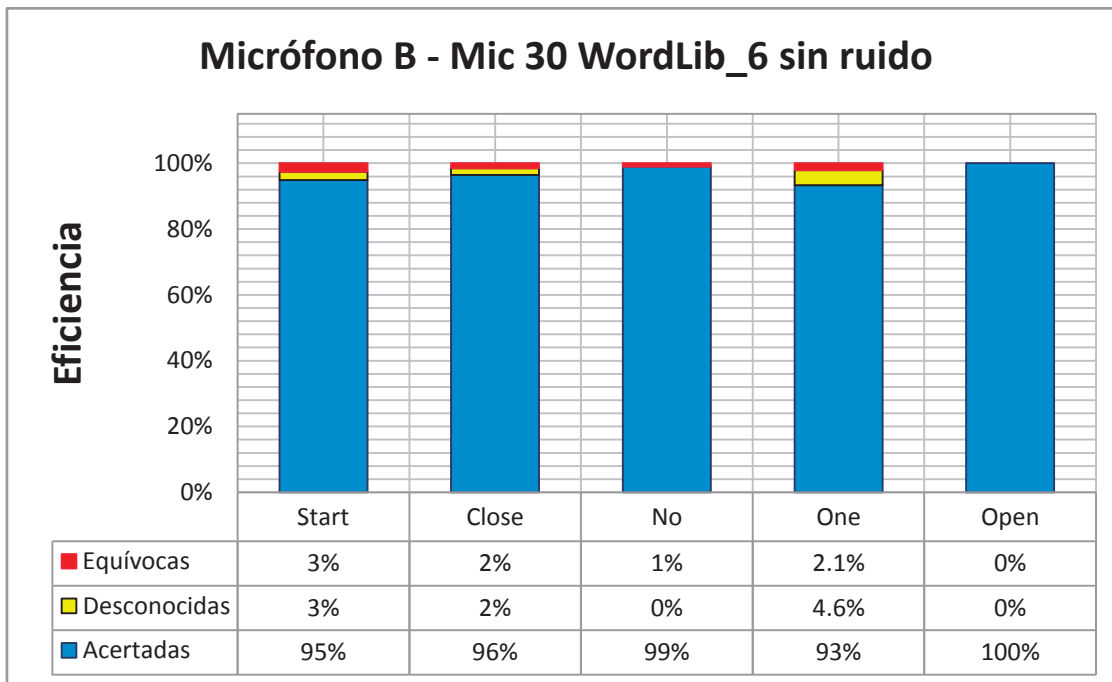
Gráfica 4.4 Micrófono A - Eficiencia WordLib_6 para MIC 30 con ruido



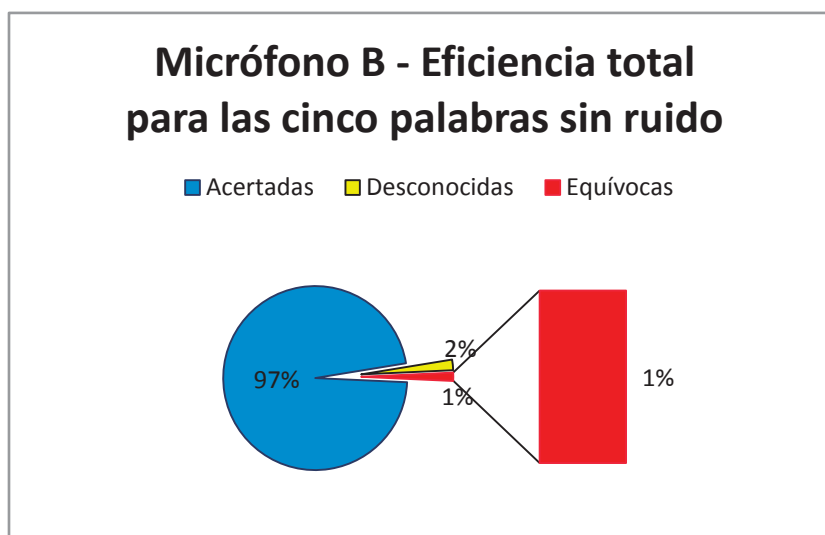
Micrófono B

La biblioteca que obtuvo el mejor resultado de reconocimiento de voz sin ruido es *WordLib_6* para una sensibilidad de *MIC 30*, obteniendo una eficiencia de 97%. La grafica 4.5 muestra la eficiencia por palabra y la 4.6 la eficiencia total sin ruido.

Grafica 4.5 Micrófono B - Eficiencia de cada palabra sin ruido

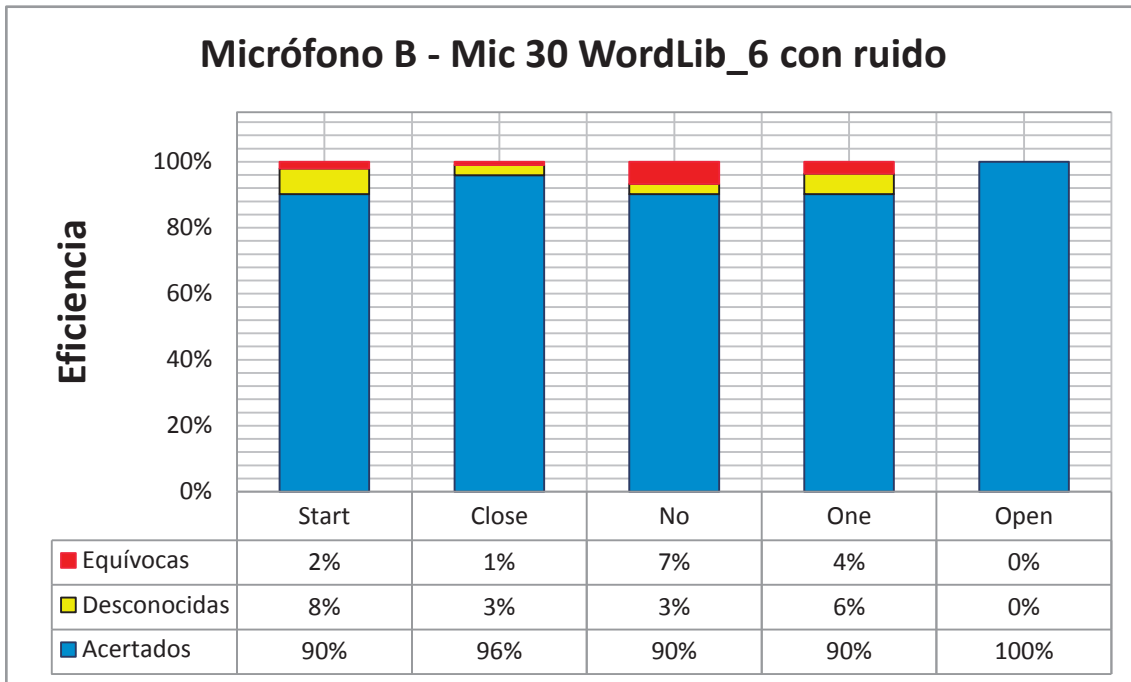


Gráfica 4.6 Micrófono B - Eficiencia WordLib_6 para MIC 30 sin ruido



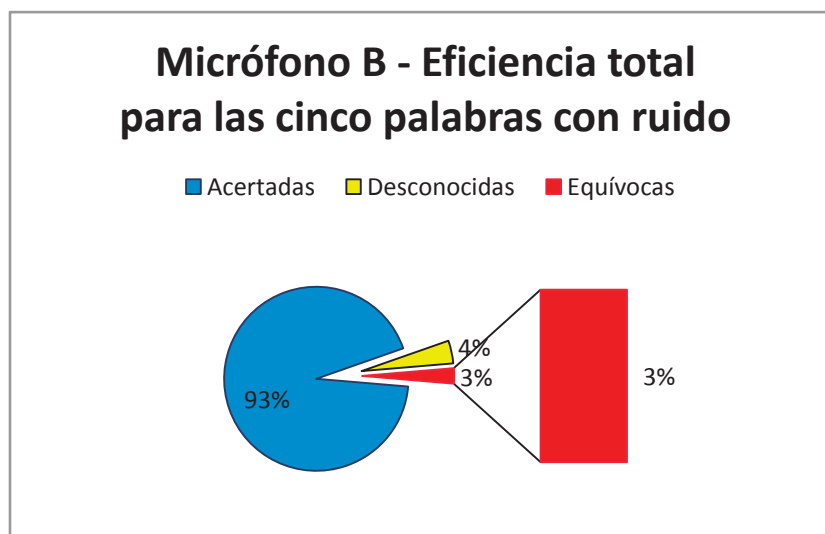
La segunda toma de muestras se hizo en un ambiente con ruido. La gráfica 4.7 muestra el desempeño de esta biblioteca en una situación de extremo ruido.

Gráfica 4.7 Micrófono B - Eficiencia de cada palabra con ruido



La gráfica 4.8 muestra la disminución de 4% de la eficiencia comparado con la muestra sin ruido.

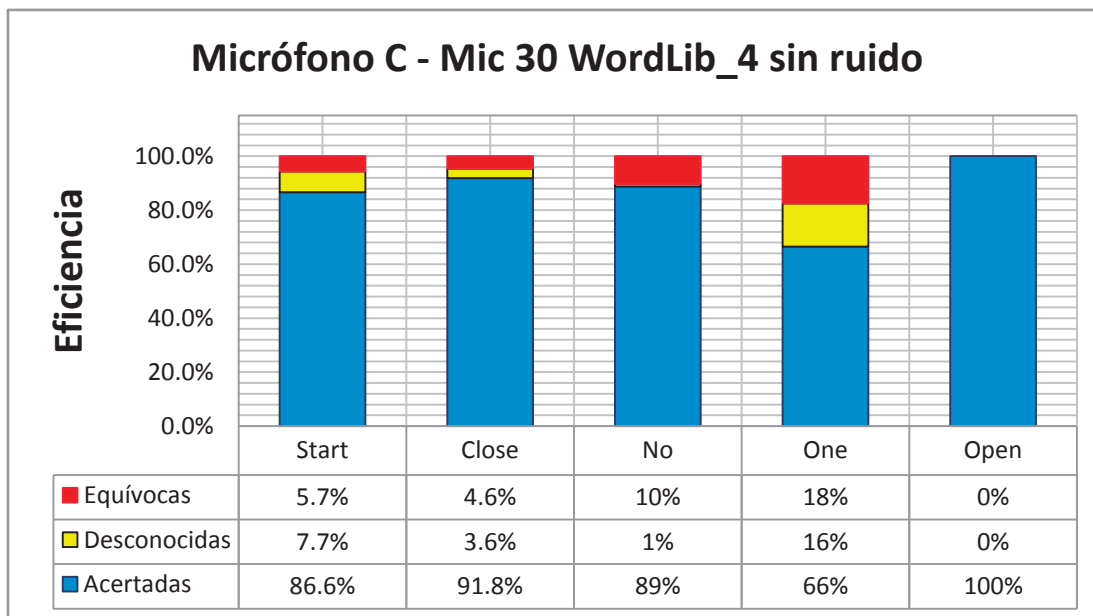
Gráfica 4.8 Micrófono B - Eficiencia WordLib_6 para MIC 30 con ruido



Micrófono C

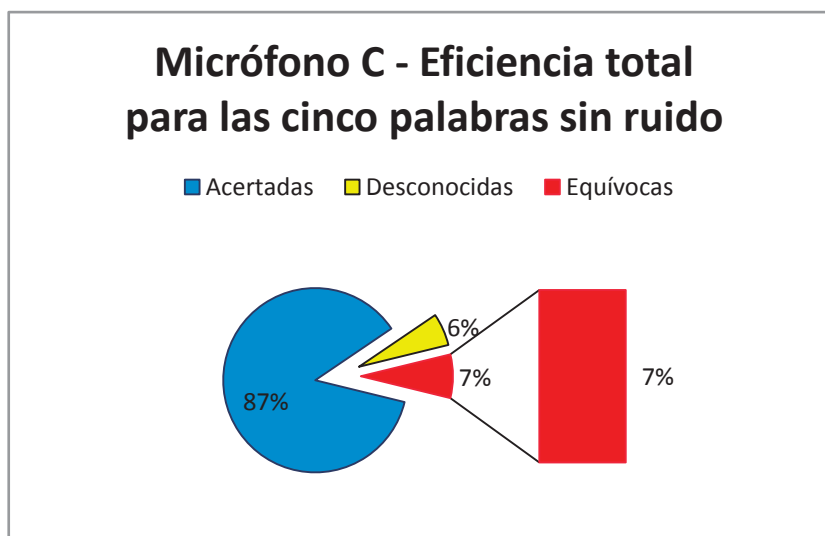
La biblioteca que obtuvo un mejor desempeño para este micrófono es la *WordLib_4* para una sensibilidad del *MIC 30*.

Gráfica 4.9 Micrófono C - Eficiencia de cada palabra sin ruido



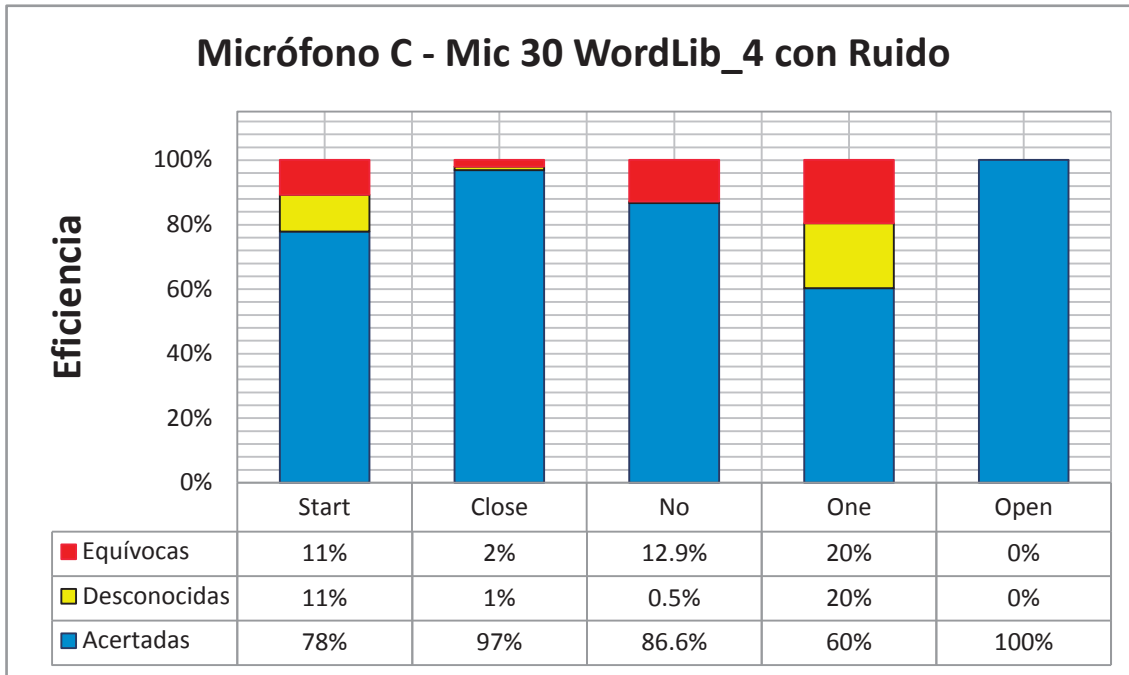
La gráfica 4.10 muestra que el dispositivo obtuvo un 87% de eficiencia. Comparado con los otros micrófonos éste fue el que obtuvo la menor eficiencia.

Grafica 4.10 Micrófono C - Eficiencia WordLib_4 para MIC 30 sin ruido

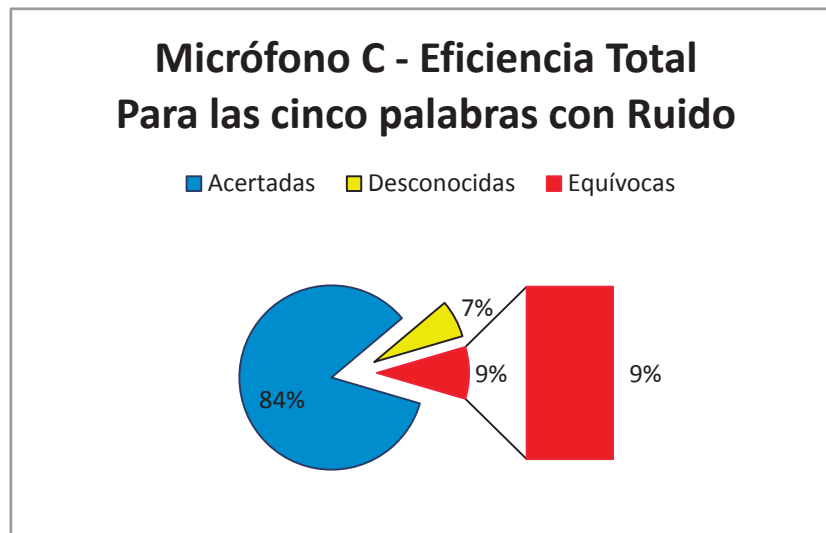


En un ambiente con ruido el desempeño de este disminuyó 3%, como lo muestran las gráficas 4.11 y 4.12.

Grafica 4.11 Micrófono C - Eficiencia de cada palabra con ruido



Gráfica 4.12 Micrófono C - Eficiencia WordLib_4 para MIC 30 con ruido



4.5 ANÁLISIS DE RESULTADOS

De estos datos se puede concluir la tabla 4.6, que muestra la eficiencia para cada micrófono analizado.

Tabla 4.6 Resultados obtenidos del muestreo

<i>Micrófono</i>	<i>Biblioteca</i>	<i>Sensibilidad</i>	<i>Eficiencia</i>	
			<i>Sin ruido</i>	<i>Con ruido</i>
A	WordLib_6	MIC 30	92%	90%
B	WordLib_6	MIC 30	97%	93%
C	WordLib_4	MIC 30	87%	84%

Por último en la tabla 4.7 se muestran los parámetros buscados para un óptimo reconocimiento de voz.

Tabla 4.7 Mejores parámetros para el reconocimiento de voz

<i>Parámetro</i>	<i>Optimo</i>		
Sensibilidad	MIC 30		
Eficiencia del dispositivo	93%		
Micrófono	B		
Ruido - Predefinido	Noise Standart		
	Nivel SNR 15 dB		
	White 50%	Car 0%	Office 50%

CONCLUSIONES

Al inicio de la investigación para el desarrollo del presente proyecto no tenía conocimiento real de lo que implicaba el reconocimiento de voz, temas como filtros para la atenuación de ruidos, espectros de voz, algoritmos utilizados para el reconocimiento de voz, programación en leguajes, como C o ensamblador, eran poco conocidos para mí. Sin embargo, durante el desarrollo del proyecto estos temas se hicieron de uso común, lográndose así manipular los aspectos generales e importantes para un mejor desarrollo del proyecto.

El presente trabajo planteó una nueva alternativa en la forma de control de las prótesis funcionales utilizando el reconocimiento de voz que no se había contemplado para este tipo de problemas con prótesis. Esto tal vez, debido a la dificultad que presenta utilizar la voz como señal de entrada, ya que cada voz humana es única, y palabras idénticas pueden tener diferentes significados. Sin embargo, en los últimos años la tecnología del reconocimiento de voz ha tenido avances importantes en el procesamiento de señal, algoritmos y arquitecturas.

Aunque existe un gran número de productos orientados al reconocimiento de voz de diferentes proveedores, algunos productos ofrecen un reconocimiento de voz no mayor al 95%. En realidad algunos no sobrepasan el 85% en ambientes con ruido, además son circuitos complejos y programados listos para el reconocimiento de voz, sin la opción de poder reprogramar algunas variables que puedan mejorar el rendimiento del reconocimiento de voz.

La empresa *Microchip* ofrece una opción a estos problemas con la familia DSC que dentro de su gama de aplicaciones está el de reconocimiento de voz, ofreciendo una forma mucho más sencilla de programar, ya que la mayoría de las instrucciones son muy similares a los microcontroladores PIC's.

Además *Microchip* ofrece las variables con las cuales se pueda manipular y mejorar la eficiencia, comparándole con lo que podría obtenerse con otros productos como un DSP que son mucho más complejos de programar y el costo de éstos es mucho más elevado que el de un DSC.

Si bien, en la actualidad existen diversos productos enfocados al reconocimiento de voz, la información sobre ésta es muy poca y las empresas que han desarrollado dispositivos con reconocimiento de voz aceptables no la comparten o la comparten a medias.

Uno de los principales problemas que se tuvieron en el desarrollo del proyecto, fue encontrar información para desarrollar el algoritmo basado en el modelo oculto de Markov, algoritmo que compara los patrones de voz entrantes con los espectros de voz guardados previamente en la biblioteca. Es por eso que para el desarrollo de esta parte del software se utilizó la herramienta *Word Library Builder*.

Esta herramienta crea el algoritmo de comparación de espectros de voz. Eliminando así, uno de los problemas que se tuvieron para el desarrollo del proyecto. Esto permitió integrar los demás programas desarrollados para el reconocimiento de voz.

Una vez iniciada la investigación para el reconocimiento de voz se plantearon algunas interrogantes importantes que influirían en la decisión de si era factible o no el reconocimiento de voz como sistema de control para la prótesis de mano. Las interrogantes eran:

1. ¿Qué tiempo que tarda el dispositivo en reconocer el comando de voz y transformarlo en movimientos de la mano artificial?
2. ¿Qué tendrá el dispositivo y que tipos de errores se tendrán durante el reconocimiento de voz?
3. ¿Afectará el tipo de micrófono seleccionado a la eficiencia del dispositivo?

El tiempo que se propuso como máximo para reconocer el comando de voz y que el dispositivo accione la mano artificial y así obtener un desempeño satisfactorio fue de 1 segundo. Ya que se integró el reconocimiento de voz y el dispositivo que acciona la mano artificial, se comprobó que el sistema responde en un tiempo aproximadamente de medio segundo.

Algunas de las variantes que afectaron el desempeño de todo el sistema fue que se optó por utilizar dos dispositivos uno encargado del reconocimiento de voz y otro para accionar la prótesis. Otra variante fue que la prótesis utilizada para la integración del sistema no es el diseño final si no es un prototipo, al cual le hacen falta algunos ajustes para que su desempeño sea mejor.

Después de la coordinación entre el software, el circuito electrónico y la mano artificial, se realizó un muestreo para conocer la eficiencia del sistema y los diferentes errores que éste presentaba. Las eficiencias que se obtuvieron fueron desde 93% para un micrófono B, 90% para el micrófono A y 84% para el micrófono C en un ambiente con ruido para un total de 5000 palabras, tomando un muestreo simple aleatorio para una población finita.

Del muestreo también se obtuvieron dos tipos de errores (utilizando el micrófono que mejor desempeño obtuvo) el principal y que ocurre con más frecuencia es el de desconocer la palabra entrante, con un 4% de error. El segundo es el de confundir la palabra entrante con otra palabra de las que se encuentra en la biblioteca, mostrando un error del 3%.

Durante la toma de muestras se utilizaron tres micrófonos de diferentes precios, con y sin supresión de ruidos. Las eficiencias que se obtuvieron para los micrófonos fueron muy diferentes, aumentando hasta en 9 puntos porcentuales para micrófonos con supresión de ruidos. Mostrando que la elección del micrófono afecta en gran medida el desempeño del reconocimiento de voz.

Con los resultados obtenidos se concluye que la hipótesis planteada en un principio es verdadera. Se puede construir un software con su respectivo hardware que sea capaz de transformar las señales sonoras (palabras) en movimientos de una mano artificial. Sin embargo, ahora se plantean nuevas interrogantes, como:

- ¿Cuál será el comportamiento que tendrá el sistema en la vida cotidiana de la persona que la utilizará?
- ¿Cuántas baterías son necesarias para que trabaje un mínimo de 12 horas con una eficiencia aceptable?
- ¿Qué tanto afectará al desempeño del dispositivo el que la mano artificial no fuera el diseño final si no que apenas fuera un prototipo al cual le hacen falta algunos ajustes para mejorar su desempeño?
- ¿Disminuiría el tiempo de reacción de la mano si el DSC se encargara de todo y no fueran dos los dispositivos encargados del sistema?

Por último, durante el desarrollo del proyecto se abrió una nueva línea de investigación, el algoritmo basado en el modelo oculto de Markov utilizado para la comparación de espectros de voz, que si bien se abarcó en este proyecto, la información recabada no fue suficiente para poder desarrollar el algoritmo que permitirá realizar reconocimiento de voz para comandos en español, mejorar la eficiencia del sistema.

BIBLIOGRAFÍA

- [1] INEGI. <http://www.inegi.gob.mx/est/default.aspx?c=4338&e=16>
- [2] Principales causas de amputación. <http://ccc.inaoep.mx/~emorales/propfin.pdf>
- [3] Revista gaceta, número 3, 696, 16 de febrero de 2004. Dirección electrónica (Email): dginfo@condor.dgsca.unam.mx. <http://www.unam.mx/gaceta>
- [4] Antecedentes, años 80's y 90's.
http://www.secyt.frba.utn.edu.ar/gia/IA1_IntroReconocimientoVoz.pdf
- [5] Historia del reconocimiento de voz, año 50's, 60's y 70's
Página: <http://ict.udlap.mx/people/ingrid/Clases/IS412/index.html>
- [6] Historia años 50's y 70's.
http://tamarisco.datsi.fi.upm.es/ASIGNATURAS/FRAV/apuntes/historia_rv
- [7] Historia año 2000. http://www.ejournal.unam.mx/busca_palabra.html
- [8] FreeSpeech 2000.
<http://www.munimadrid.es/portal/site/munimadrid/menuitem.f4bb5b953cd0b0aa7d245f019fc08a0c/?vgnextoid=84d1b6b6ee36a010VgnVCM10000026205a0aRCRD&vgnnextchannel=254a171c30036010VgnVCM100000dc0ca8c0RCRD&idCapitulo=4213776>
- [9] L & H Voice Xpress Professional, Version 4.
http://accesibilidad.blogspot.com/2005_05_01_archive.html
- [10] Dragon Naturally Speaking Preferred 9.
www.pas.deusto.es/recursos/Reconocimiento%20de%20Voz%20-%20Grupo%20PAS.ppt
- [11] Kit manos libres. <http://www.ordenadores-y-portatiles.com/reconocimiento-de-voz.html>
- [12] Reconocimiento de voz en teléfonos celulares.
<http://www.celularis.com/tecnologia/reconocimiento-de-voz-en-moviles.php>
- [13] Reconocimiento de voz en teléfonos celulares.
<http://www.movilae.com/2007/09/20/la-tecnologia-de-reconocimiento-de-voz-%C2%BFel-futuro>

[14] Tarjeta de crédito con reconocimiento de voz.

<http://isogadgets.com/category/tarjetas>

[15] Software de reconocimiento de voz en los aviones. <http://www.ounae.com/2007/10/18/software-de-reconocimiento-de-voz-en-los-aviones>

[16] TV con reconocimiento de aplausos y gestos. <http://www.blogdelgizmo.com/2007/10/10/tv-con-reconocimiento-de-palmas-y-gestos/>

[17] Robot Hitachi. <http://gizmologia.com/2008/07/nuevo-robot-con-avanzado-reconocimiento-de-voz-de-hitachi/>

[18] Etapas del reconocimiento de voz.

<http://ict.udlap.mx/people/ingrid/Clases/IS412/index.html>

[19] Tipos de reconocimiento de voz. <http://www.jegsworks.com/Lessons-sp/lesson3/lesson3-5.htm>

[20] José María Angulo Usategui, Aritza Etxebarria, Ignacio A. Martínez e Iván Trueba Parra. DsPIC diseño práctico de aplicaciones. Mc Graw Hill 1ra edición.

[21] Selección del dsPIC.

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2127¶m=en023431

[22] Metodología para el reconocimiento de voz. Nigel Gardner. A Beginners Guide to the Microchip PIC. Farnell Components

[23] Herramienta dsPIC30F Speech Recognition Word Library Builder.

<http://www.microchip.com/downloads>

[24] Creación de la biblioteca.

http://www.microchip.com/downloads/en/DeviceDoc/dsPIC_Speech_Recognition_User_Guide_70140a.pdf

[25] MPLAB IDE v8.10. <http://www.microchip.com>

[26] Proton IDE 1.1.2.8. <http://www.mecanique.co.uk>

- [27] WinPic800 v3.59, Sisco Benach Font. <http://perso.wanadoo.es/siscobf>
- [28] Proteus 6 Profecinal, Labcenter Electronics. <http://www.labcenter.co.uk>
- [29] Técnicas de muestreo. William G. Cochran. Editorial continental S.A. C.E.C.S.A 6ta impresión, impreso en México 1976
- [30] Prótesis. <http://es.wikipedia.org/wiki/Pr%C3%B3tesis>
- [31] Reconocimiento de voz. <http://www.ejournal.unam.mx/compuysistemas/vol09-03/CYS09307.pdf>
- [32] Microcontrolador. <http://www.monografias.com/trabajos18/descripcion-pic/descripcion-pic.shtml>
- [33] Procesador digital de señal.
http://es.wikipedia.org/wiki/Procesador_digital_de_se%C3%B1al
- [34] Arquitectura Harvard. http://es.wikipedia.org/wiki/Arquitectura_Harvard
- [35] Lenguaje de programación.
http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n
- [36] Depurador. <http://www.mastermagazine.info/termino/4582.php>
- [37] Emulador. <http://es.wikipedia.org/wiki/Emulador>
- [38] Señal analógica. http://es.wikipedia.org/wiki/Se%C3%B1al_anal%C3%B3gica
- [39] Señal digital. http://es.wikipedia.org/wiki/Se%C3%B1al_digital
- [40] ventajas de transmisión digital. <http://www.monografias.com/trabajos17/tipos-transmision-datos/tipos-transmision-datos.shtml>
- [41] Ruido blanco. http://es.wikipedia.org/wiki/Ruido_blanco
- [42] Circuito eléctrico. http://es.wikipedia.org/wiki/Circuito_el%C3%A9ctrico
- [43] Componente electrónico.
http://es.wikipedia.org/wiki/Componente_electr%C3%B3nico

ANEXO I. CONCEPTOS

Prótesis

Una prótesis es una extensión artificial que reemplaza una parte faltante del ser vivo. Por otra parte, la ortopedia aporta un suplemento al restablecimiento del cuerpo humano. Las prótesis son típicamente utilizadas para reemplazar partes perdidas por heridas (traumáticas) o que faltan de nacimiento (congénitas), supliendo partes inexistentes, por ejemplo permite a transexuales cambiar su apariencia sexual.

Algunos tipos de prótesis incluyen [30]:

- Miembros artificiales
- Prótesis auditiva, ó auxiliares auditivos
- Prótesis oculares
- Prótesis faciales
- Prótesis maxilofaciales
- Prótesis sexuales

Reconocimiento de voz

Un sistema de Reconocimiento Automático del Habla (*SRAH*) es aquel sistema automático que es capaz de interpretar una señal de voz emitida por un individuo. Dicha señal requiere pasar por un proceso de digitalización para obtener elementos de medición (muestras), los cuales permitan denotar su comportamiento e implementar procesos de tratamiento de señal enfocados al reconocimiento [31].

Microcontrolador

Se denomina microcontrolador a un dispositivo programable capaz de realizar diferentes actividades que requieran del procesamiento de datos digitales, del control y comunicación digital de diferentes dispositivos.

Los microcontroladores poseen una memoria interna que almacena dos tipos de datos: las instrucciones, que corresponden al programa que se ejecuta, y los registros, es decir, los datos que el usuario maneja, así como registros especiales para el control de las diferentes funciones del microcontrolador.

Los microcontroladores se programan en el lenguaje ensamblador y cada microcontrolador varía su conjunto de instrucciones de acuerdo a su fabricante y

modelo. De acuerdo al número de instrucciones que el microcontrolador maneja se le denomina de arquitectura RISC (*Reduced Instruction Set Computer-computadora con conjunto de instrucciones reducidas*) o CISC (*Complex Instruction Set Computing-computadora con conjunto de instrucciones complejas*) [32].

Procesador digital de señal

DSP es el acrónimo de *Digital Signal Processor*, que significa Procesador Digital de Señales.

Un DSP es un sistema basado en un procesador o microprocesador que posee un juego de instrucciones, un hardware y un software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad. Debido a esto es especialmente útil para el procesamiento y representación de señales analógicas en tiempo real. En un sistema que trabaja en tiempo real, se reciben muestras, normalmente provenientes de un convertidor analógico/digital (ADC).

Se ha dicho que puede trabajar con señales analógicas, pero es un sistema digital, por lo tanto necesitará un convertidor analógico/digital a su entrada y digital/analógico en la salida. Como todo sistema basado en procesador programable necesita una memoria donde almacenar los datos con los que trabaje y el programa que ejecute [33].

Arquitectura Harvard

Todas las computadoras constan principalmente de dos partes: el CPU (*Central Processing Unit*), que procesa la información y la memoria que guarda la información. Cuando hablamos de memoria manejamos dos parámetros, los datos o registros, y el lugar donde se encuentran almacenados (o dirección). Se pueden fabricar memorias con tiempo de respuesta mucho más rápidas, pero a costa de un precio muy alto. La solución, por tanto, es proporcionar una pequeña cantidad de memoria adicional muy rápida, conocida con el nombre de caché.

La arquitectura Harvard ofrece una solución particular a este problema. Las instrucciones y los datos se almacenan en cachés separados para mejorar el rendimiento. Por otro lado, tiene el inconveniente de tener que dividir la cantidad de caché entre los dos, por lo que funciona mejor sólo cuando la frecuencia de lectura de instrucciones y de datos es aproximadamente la misma. Esta arquitectura suele utilizarse en DSP's o procesador de señal digital, usados habitualmente en productos para procesamiento de audio y video [34].

Lenguaje de Programación

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones, y es utilizado para controlar el comportamiento físico y lógico de una máquina.

La necesidad de recordar secuencias de programación para las acciones usuales llevó a denominarlas con nombres fáciles de memorizar y asociar: ADD (*sumar*), SUB (*restar*), MUL (*multiplicar*), CALL (*ejecutar subrutina*), etc. A esta secuencia de proposiciones se le denominó "instrucciones", y a este conjunto de instrucciones se le llamó lenguaje ensamblador [35].

Depurador

En inglés *debugger*. En el trabajo con herramientas de programación, consiste en la revisión de la aplicación generada con el fin de eliminar los posibles errores que puedan existir en ésta. También persigue la optimización del programa para que su funcionalidad y velocidad sean las máximas. Esta operación pueden realizarla programas especializados, lo que facilita el trabajo del programador y acorta el tiempo empleado en la fase de depuración. Para evitar los errores en el funcionamiento de los programas, suelen lanzarse varias versiones «*beta*» de cada programa [36].

Emulador

En informática, un emulador es un software que permite ejecutar programas de computadora en una plataforma (arquitectura hardware o sistema operativo) diferente de la cual fueron escritos originalmente. A diferencia de un simulador, que sólo trata de reproducir el comportamiento del programa, un emulador trata de modelar de forma precisa el dispositivo que se está emulando [37].

Señal analógica

Una señal analógica es aquella función matemática continua en la que es variable su amplitud y periodo (representando un dato de información) en función del tiempo. Algunas magnitudes físicas comúnmente portadoras de una señal de este tipo son eléctricas como la intensidad, la tensión y la potencia, pero también pueden ser hidráulicas como la presión, térmicas como la temperatura, mecánicas, etc. La magnitud también puede ser cualquier objeto medible como los beneficios o pérdidas de un negocio.

La gran desventaja respecto a las señales digitales, es que en las señales analógicas, cualquier variación en la información es de difícil recuperación, y esta pérdida afecta en gran medida al correcto funcionamiento y rendimiento del dispositivo analógico [38].

Señal digital

Se dice que una señal es digital cuando las magnitudes de la misma se representan mediante valores discretos en lugar de variables continuas. Por ejemplo, el interruptor de la luz sólo puede tomar dos valores o estados: abierto o cerrado, o la misma lámpara: encendida o apagada.

Los sistemas digitales, como por ejemplo el ordenador, usan lógica de dos estados representados por dos niveles de tensión eléctrica, uno alto y otro bajo. Por abstracción, dichos estados se sustituyen por ceros y unos, lo que facilita la aplicación de la lógica y la aritmética binaria [39].

En la transmisión digital existen varias ventajas que la hacen que tenga mayor aceptación cuando se compara con la analógica. Estas son [40]:

- La ventaja principal de la transmisión digital es la inmunidad al ruido. Las señales analógicas son más susceptibles que los pulsos digitales a la amplitud no deseada, frecuencia y variaciones de fases.
- Se prefieren a los pulsos digitales por su mejor procesamiento. Los pulsos digitales pueden guardarse fácilmente, mientras que las señales analógicas no son fáciles de guardar.
- Las señales digitales son más sencillas de medir y evaluar.

Ruido Blanco

El ruido blanco es una señal aleatoria que se caracteriza porque sus valores de señal en dos instantes de tiempo diferentes no guardan correlación estadística. Como consecuencia de ello, su densidad espectral de potencia (*PSD, Power Spectral Density*) es una constante, su gráfica es plana. Esto significa que la señal contiene todas las frecuencias y todas ellas tienen la misma potencia.

Se pueden hallar ejemplos domésticos del ruido blanco en el funcionamiento de los secadores de pelo, aspiradoras, y el ruido e imagen estática de los televisores [41].

Circuito eléctrico

Se denomina circuito eléctrico a una serie de elementos o componentes eléctricos o electrónicos, tales como resistencias, inductancias, condensadores, fuentes, y/o dispositivos electrónicos semiconductores, conectados eléctricamente entre sí con el propósito de generar, transportar o modificar señales electrónicas o eléctricas [42].

Componente electrónico

Se denomina componente electrónico a aquel dispositivo que forma parte de un circuito eléctrico. Se suele encapsular, generalmente en un material cerámico, metálico o plástico, y terminar en dos o más terminales o patillas metálicas. Se diseñan para ser conectados entre ellos, normalmente mediante soldadura, a un circuito impreso, para formar el mencionado circuito.

Hay que diferenciar entre componentes y elementos. Los componentes son dispositivos físicos, mientras que los elementos son modelos o abstracciones idealizadas que constituyen la base para el estudio teórico de los mencionados componentes. Así, los componentes aparecen en un listado de dispositivos que forman un circuito, mientras que los elementos aparecen en los desarrollos matemáticos de la teoría de circuitos [43].

ANEXO II. PROGRAMA PRINCIPAL DE LA BIBLIOTECA WordLibSR.h

```
// Word Library Builder Version 1.00
//
// Word Library Vocabulary Header file
//
// Creation date:           09/01/08 20:57:28
//
// Word Library file:      C:\Carlos\WordLib_6\WordLibSR.swl
// Language:               English
// Comments:               Created by Word Library Builder
//
// Word Library:          WordLibSR
// Noise profile:         standard
//   SNR level:           15 dB
//   white:                50%
//   car:                  0%
//   office:               50%
// Keyword:                start
// number of words:       5

#define    SRL_NUM_WORDS    5        // WL size

enum
{
    SRL_NONE                ,        // not ready yet
    SRL_START               ,        // word 1 (keyword)
    SRL_CLOSE               ,        // word 2
    SRL_NO                   ,        // word 3
    SRL_ONE                  ,        // word 4
    SRL_OPEN                 ,        // word 5
    SRL_UNKNOWN              ,        // unknown word
};
```

ANEXO III. PROGRAMA PARA EL RECONOCIMIENTO DE VOZ

- Programa **Config_Dip_SR.c**

```

//*****
//* ***** *
//**                                     **
//**          Configuración del dispositivo para el          **
//**          programa de reconocimiento de voz              **
//**                                     **
//* ***** *
//*****

#include "p30f6014A.h"

//.....
//..          Ajustes de interrupción          ..
//.....

_FOSC(CSW_FSCM_OFF & ECIO_PLL8); // Habilita la función ECIO con x8 PLL
_FWDT(WDT_OFF); // Inhabilita WDT (Watchdog Timer)
_FBORPOR(PBOR_OFF & MCLR_EN); // Inhabilita PBOR, Habilita MCLR
_FGS(CODE_PROT_OFF); // Inhabilita el código de protección
    
```

- Programa **Config_Param_SR.c**

```

//*****
//* ***** *
//**                                     **
//**          dsPIC30F Reconocimiento de Voz              **
//**                                     **
//* ***** *
//*****

#include "Arch_Arranq_Lib.h"
#include "Variables_SR.h"
#include "..\WordLib_6\WordLib_SR.h"

//.....
//.....
//..          Ajustes de parámetros          ..
//.....
//.....
    
```

```

//.....
//..          Inicialización_SR          ..
//.....

#define fclock          6144L // Frecuencia del oscilador externo KHz
#define pll_setting     8 // Modo del multiplicador PLL
#define timeout         5 // Período de interrupción en segundos,
                        // Máximo 21 Segundos para una frecuencia
                        // De procesador de 12.288

#define mic_gain        30 // Nivel de sensibilidad del micrófono en
                        // El Si3000 codec (dB), puede tomar los
                        // Sigüientes Valores 0,10,20,30

#define SR_mode         0 // Modo de operación del SR:
                        // 0 self-test Habilitado, Habilita la
                        // Activación de la palabra clave (keyword)
                        // 1 self-test Inhabilitado, Habilita
                        // la activación de la palabra clave
                        // 2 self-test Habilitado, Inhabilita la
                        // Activación de la palabra clave (keyword)
                        // 3 self-test Inhabilitado, Inhabilita
                        // Activación de la palabra clave (keyword)

int main(void)
{
    int word; // Palabra Reconocida
    int SKW_flag; // Muestra el indicador de la palabra clave
    Inicializa_PORTA(); // Inicializa el PORTA
    Inicializa_PORTD(); // Inicializa el PORTD

//.....
//..          Inicialización global del SR y Auto-Test          ..
//.....

if (SR_Initialization(fclock, pll_setting, timeout, mic_gain, SR_mode))
{
    LEDGOFF_AutoTest_Falla(); // Apaga el LED Verde
    return;
}
if (SR_mode == 0 || SR_mode == 2) // Si el self-test está habilitado
    LEDG_AutoTest_OK(); // Enciende el LED Verde

if (SR_mode < 2) // Si la palabra clave está habilitada
    LEDR_Diga_KeyWord(); // Enciende el LED Rojo

//.....
//.....
//..          Ciclo principal del funcionamiento del SR          ..
//.....
//.....

SKW_flag = 1; // Muestra aviso de KW
while (1)
{
    word = SR_Recognizer(); // Reconocimiento de la palabra SR

```

```
//.....
//..          Aviso de la palabra clave ..
//..          Si los 5 seg.(timeout) pasaron ..
//.....

if (SKW_flag      == 1 &&          // Si (Muestra el indicador KW y
SR_STATUSbits.state == 1 &&      // Espera el estado de la
                                // Palabra clave Y
SR_STATUSbits.pause == 1 &&      // Pausa entre palabras y
SR_mode < 2)                    // Habilita la palabra clave)
    {
        LEDR_Diga_KeyWord();      // Enciende el LED Rojo
        SKW_flag = 0;            // No muestra Kw
    }
switch (word)
    {

        case SRL_NONE:           // no requiere reacción
            break;               // Reconocimiento en Proceso

        case 1 ... SRL_UNKNOWN:
            if (word == 1) SR_mode > 1;
            EnviaCodigo(word, SR_MaxMagnitude,
                SR_STATUSbits.clipping);
            SKW_flag = 1;         // Muestra Kw
            break;

        default :
            SKW_flag = 1;        // Muestra Kw
            break;
    };
}
```

- Programa **Variables_SR.h**

```
/**
**
** Declaración de variables para el dsPIC30F reconocimiento de voz **
**
**
**
**
*/

void LEDR_Diga_KeyWord(void);
void LEDG_AutoTest_OK(void);
void LEDGOFF_AutoTest_Falla(void);
void EnviaCodigo(int word, int MaxMagn, int SR_Status);
void Inicializa_PORTA(void);
void Inicializa_PORTD(void);
```

- Programa Prog_Rec.s

```

;*****
;* *****
;**
;**          Programa general para usar la biblioteca del          **
;**          reconocimiento de voz del dsPIC30F                    **
;**                                                                 **
;* *****
;*****

.include "p30f6014A.inc"

;.....
;.....
;..          Declaraciones globales          ..
;.....
;.....

.global _LEDOFF_AutoTest_Falla
.global _LEDG_AutoTest_OK
.global _LEDR_Diga_KeyWord
.global _EnviaCodigo
.global _Inicializa_PORTA
.global _Inicializa_PORTD

.text

;*****
;* *****
;**
;**          Subrutina '_EnviaCodigo'          **
;**          -----          **
;**          Encuentra en la memoria del programa la palabra que reconoció y **
;**          La envía al PORTD. Todas las palabras se alinean para programar **
;**          El límite de la palabra. El 1er byte de cada palabra siempre **
;**          Es "0x20", esto significa el principio de una palabra.          **
;**                                                                 **
;**          Argumentos:          **
;**          W0 - número de la palabra reconocida en el vocabulario          **
;**                                                                 **
;**          Recursos usados:          **
;**                                                                 **
;**          - Registros del sistema: PORTA          **
;**          - Registros de trabajo   : W0, W8          **
;**                                                                 **
;* *****
;*****

```

```

_EnviaCodigo:

mov     w0,          w8      ; Salva el número de palabra reconocida
                                ; En W8
call    Transmite_Codigo_PortD ; Llama a subrutina para encender led's
nop
bclr   PORTA,       #RD7 ; Limpia el bit 7 del PORTA, apagando el
                                ; Led rojo Cuando se habla y se
                                ; enciende cuando se solicita la
                                ; Palabra clave
return                                ; Fin del procedimiento "_EnviaCodigo"

```

```

;*****
;* *****
; **
; **      Subrutina '_LEDR_Diga_KeyWord'      **
; **      -----                          **
; **      Enciende el led rojo cuando se solicite la palabra clave      **
; **      Y se apaga cuando está reconociendo las palabras                **
; **                                                                 **
; **      Recursos usados:          **
; **                                                                 **
; **      - Registros del sistema: PORTA      **
; **                                                                 **
; **                                                                 **
;* *****
;*****

```

```

_LEDR_Diga_KeyWord:

bset   PORTA,       #RA7 ; Pone a 1 el bit 7 del PortA, enciende el led rojo
                                ; Para avisar que se necesita la KeyWord otra vez
return                                ; Fin del procedimiento "_LEDR_Diga_KeyWord"

```

```

;*****
;* *****
; **
; **      Subrutina '_LEDG_AutoTest_OK'      **
; **      -----                          **
; **      Enciende el led verde cuando el auto-test termina            **
; **      Favorablemente.                                                **
; **                                                                 **
; **      Recursos usados:          **
; **                                                                 **
; **      - Registros del sistema: PORTA      **
; **                                                                 **
; **                                                                 **
;* *****
;*****

```

```

_LEDG_AutoTest_OK:

bset  PORTA,      #RA6      ; Pone 1 bit6 del PortA, enc. el led verde
                                ; Para avisar que el self-test Finalizo
                                ; Satisfactoriamente
return                                ; Fin del procedimiento "_LEDG_AutoTest_OK"

;*****
;* *****
; **
; **      Subrutina '_LEDGOFF_AutoTest_Falla'
; **      -----
; **      No se enciende el led verde, indicando que el auto-test fallo
; **
; **      Recursos usados:
; **      - Módulos de interfaz: Ninguno
; **
;* *****
;*****

_LEDGOFF_AutoTest_Falla:

                                ; Cuando pase esto, solo no se enciende el led verde
return                            ; Fin del procedimiento "_LEDGOFF_AutoTest_Falla"

;*****
;* *****
; **
; **      Subrutina '_Inicializa_PORTA'
; **      -----
; **      Ajusta el puerto A como salidas para encender led verde y rojo
; **      Que indicarán:
; **
; **      Autotest ok ----- LED Verde
; **      Di Palabra clave --- LED Rojo
; **
; **
; **      Recursos usados:
; **
; **      - Registros del sistema: PortA, TRISA
; **      - Registros de trabajo: W0
; **
;* *****
;*****

_Inicializa_PORTA:

clr  PORTA      ; Prepara ajuste del puerto, Despejando el PORTA
mov  #0xFF3F,   w0    ; Carga w0 con b'111111100111111' en decimal
                                ; "65343", para inicializar el puerto A
mov  w0,        TRISA ; Inicializa PORTB: RA6, RA7, como salida

return                                ; Fin del procedimiento '_Inicializa_PORTA'

```

```

;*****
;* ***** *
; **
; **      Subrutina '_Inicializa_PORTD'          **
; **      ----- **
; **      Ajusta el "puerto D" como salidas, para transmisión de datos **
; **      y led's, con el PIC16F877A.          **
; ** **
; **      Recursos usados:                      **
; **      - Registros del sistema: PortD, TRISD **
; **      - work registers: W0                  **
; ** **
;* ***** *
;*****

_Inicializa_PORTD:

clr   PORTD          ; Prepara ajuste del puerto, Despejando el PORTD
mov   #0xFFF0,      w0   ; Carga w0 con b'11111111110000' en decimal
                        ; "65520", para inicializar el puerto D
mov   w0,           TRISD ; Inicializa PORTD: RD0, RD1, RD2 y RD3 como
salidas
return                                ; Fin del procedimiento '_Inicializa_PORTD'

;*****
;* ***** *
; **
; **      Subrutina 'Transmite_Codigo_PortD'    **
; **      ----- **
; **      Transmite el código por el puerto D para mover los motores **
; **      Con el PIC16F877A.                  **
; ** **
; **      Recursos usados:                      **
; ** **
; **      - Registros del sistema: PortD, LATD  **
; **      - Registros de trabajo: W8, w0       **
; ** **
;* ***** *
;*****

Transmite_Codigo_PortD:

mov   w8,          LATD   ; Envía el número de la palabra reconocida
                        ; A los bits RD0, RD1, RD2, RD3 del PORTD
return                                ; Fin del procedimiento 'Transmite_Codigo_PortD'

.end                                ; Fin del programa

```

ANEXO IV. PROGRAMA PARA EL MOVIMIENTO DE LA PRÓTESIS UTILIZANDO EL PIC16F877A

```
*****
'* Name      : SERVO16F877A.BAS                               *
'* Author    : Oros Martinez Carlos                           *
'* Notice    : Copyright (c) 2008 Oros_Robotics UMSNH         *
'*           : All Rights Reserved                             *
'* Date      : 22/08/2008                                       *
'* Version   : 1.0                                             *
'* Notes     : Programa para Controlar toda la mano           *
'*           : Utiliza cristal de 20 MHz                       *
'* Caracteristicas del servo:                                  *
'*           : HITEC QUARTER SCALE HS-700BB                    *
'*           :                                                 *
*****

Device = 16F877A
Config hs_osc, Wrt_off, Wdt_off, Pwrte_off, Boden_off, Lvp_off, Cpd_off, CP_OFF,
Debug_off
XTAL = 20
ADIN_RES 10
ADIN_TAD 32_fosc
ADIN_STIME 60
ADCON1 = %10000000

Dim ANULAR1 As Word
Dim ANULAR2 As Word
Dim ANPROX1 As Word
Dim ANMED2 As Word
Dim MEDIO As Word
Dim MEDPROX As Word
Dim MEDMED As Word
Dim INDICE1 As Word
Dim INDICE2 As Word
Dim INDABDUC2 As Word
Dim INDPROX1 As Word
Dim INDMED2 As Word
Dim PULGAR1 As Word
Dim PULGAR2 As Word
Dim PULGAR3 As Word
Dim PULMTCAR1 As Word
Dim PULPROX2 As Word
Dim PULDIS1 As Word
Dim PULABDUC3 As Word
Input PORTC
Output PORTA
Symbol S1=PORTC.0
Symbol S2=PORTC.1
Symbol S3=PORTC.2

ANULAR1 = 1460
ANULAR2 = 700
ANPROX1 = 700
ANMED2 = 1450
```

MEDIO = 700
MEDPROX = 1507
MEDMED = 1581

INDICE1 = 717
INDICE2 = 700
INDABDUC2 = 1272
INDPROX1 = 1574
INDMED2 = 1722

PULGAR1 = 717
PULGAR2 = 725
PULGAR3 = 1353
PULMTCAR1 = 1759
PULABDUC3 = 844
PULPROX2 = 1438
PULDIS1 = 1295

PORTA=0

PRINCIPAL:

If S1=1 **Then**

If S2=1 **Then**

If S3=1 **Then**

GoTo SinSenal

EndIf

'Palabra 3 NO

 PORTA = %00000011

Servo PORTB.7, ANPROX1

Servo PORTB.6, ANMED2

Servo PORTB.5, MEDPROX

Servo PORTB.4, MEDMED

Servo PORTB.3, INDICE1

Servo PORTB.2, INDICE2

Servo PORTB.0, INDICE2

Servo PORTB.1, PULMTCAR1

Servo PORTD.7, PULPROX2

Servo PORTD.6, PULDIS1

Servo PORTD.5, PULABDUC3

DelayMS 20

Servo PORTB.7, ANPROX1

Servo PORTB.6, ANMED2

Servo PORTB.5, MEDPROX

Servo PORTB.4, MEDMED

Servo PORTB.3, INDICE1

Servo PORTB.2, INDABDUC2

Servo PORTB.0, INDICE2

Servo PORTB.1, PULMTCAR1

Servo PORTD.7, PULPROX2

Servo PORTD.6, PULDIS1

Servo PORTD.5, PULABDUC3

DelayMS 20

GoTo PRINCIPAL

EndIf

If S3=1 **Then**

'Palabra 5 Open

 PORTA = %00000101

Servo PORTB.7, ANULAR1

Servo PORTB.6, ANULAR2

Servo PORTB.5, MEDIO

```

        Servo PORTB.4, MEDIO
        Servo PORTB.3, INDICE1
        Servo PORTB.2, INDICE2
        Servo PORTB.0, INDICE2
        Servo PORTB.1, PULGAR1
        Servo PORTD.7, PULGAR2
        Servo PORTD.6, PULGAR1
        Servo PORTD.5, PULGAR3
        DelayMS 20
    GoTo PRINCIPAL
EndIf
'Palabra 1 clave Keyword
PORTA = %00000001
GoTo SinSenal
EndIf
If S2=1 Then
    If S3=1 Then
        'Palabra 6 Desconocida
        PORTA = %00000110
        GoTo SinSenal
    EndIf
    'Palabra 2 Close
    PORTA = %00000010
    Servo PORTB.7, ANPROX1
    Servo PORTB.6, ANMED2
    Servo PORTB.5, MEDPROX
    Servo PORTB.4, MEDMED
    Servo PORTB.3, INDPROX1
    Servo PORTB.2, INDABDUC2
    Servo PORTB.0, INDMED2
    Servo PORTB.1, PULMTCAR1
    Servo PORTD.7, PULPROX2
    Servo PORTD.6, PULDIS1
    Servo PORTD.5, PULABDUC3
    DelayMS 20
    GoTo PRINCIPAL
EndIf
If S3=1 Then
    'Palabra 4 One
    PORTA = %00000100
    Servo PORTB.7, ANPROX1
    Servo PORTB.6, ANMED2
    Servo PORTB.5, MEDPROX
    Servo PORTB.4, MEDMED
    Servo PORTB.3, INDICE1
    Servo PORTB.2, INDICE2
    Servo PORTB.0, INDICE2
    Servo PORTB.1, PULMTCAR1
    Servo PORTD.7, PULPROX2
    Servo PORTD.6, PULDIS1
    Servo PORTD.5, PULABDUC3
    DelayMS 20
    GoTo PRINCIPAL
EndIf
SinSenal:
PORTA = 0
Servo PORTB.7, ANULAR1
Servo PORTB.6, ANULAR2
Servo PORTB.5, MEDIO

```

```
Servo PORTB.4, MEDIO
Servo PORTB.3, INDICE1
Servo PORTB.2, INDICE2
Servo PORTB.0, INDICE2
Servo PORTB.1, PULGAR1
Servo PORTD.7, PULGAR2
Servo PORTD.6, PULGAR1
Servo PORTD.5, PULGAR3
DelayMS 20
GoTo PRINCIPAL
End
```