



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Desarrollo de un sistema
traductor de señas
dactilológicas a voz**

TESIS

Que para obtener el título de

Ingeniera en Computación

Ingeniero Eléctrico Electrónico

P R E S E N T A N

Rocío Ortiz Domínguez

Carlos Alberto Amaro Barbosa

DIRECTOR DE TESIS

M. I. Marcial Contreras Barrera



Ciudad Universitaria, Cd. Mx., 2017



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

“Si no tuviéramos voz ni lengua y quisiéramos enseñarnos las cosas los unos a los otros, ¿no intentaríamos, al igual que los sordomudos, indicarlos con las manos, la cabeza y el resto del cuerpo?”

(Platón, "Crátilo", XXXIV, cit. en Schermer, 1990:1)

ÍNDICE	3
DEDICATORIA.....	7
AGRADECIMIENTOS	8
AGRADECIMIENTOS	9
CAPITULO 1	10
ENFOQUE GENERAL.....	10
1.1 INTRODUCCIÓN	10
1.2 OBJETIVO DE LA TESIS.....	11
1.3 PLANTEAMIENTO DEL PROBLEMA	11
1.4 PROPUESTA Y RECURSOS	12
1.5 ALCANCES.....	12
CAPITULO 2	14
COMUNICACIÓN Y LENGUAJE DE SEÑAS	14
2.1 EL ARTE DE LA COMUNICACIÓN Y LA DISCAPACIDAD AUDITIVA	14
2.2 GRADOS DE SORDERA Y APROPIACIÓN DE LA LENGUA ESCRITA	17
2.3 SISTEMAS DE COMUNICACIÓN GESTUAL.....	18
2.4 EL LENGUAJE DE SEÑAS, LA DACTILOLOGÍA Y POSICION DE LA MANO .	19
2.5 LA MANO HUMANA.....	23
CAPITULO 3	26
ANÁLISIS DEL SISTEMA	26
3.1 SISTEMAS DE MEDIDA	26
3.2 TRANSDUCTOR Y SENSOR	27
3.2.1 Clasificación de los transductores	27
3.2.2 Potenciómetros.....	30
3.2.3 Sensor de flexión	31
3.3 SISTEMA DE ADQUISICIÓN DE DATOS.....	32
3.3.1 Características de los Convertidores analógico digital	33
3.4 SÍNTESIS DE VOZ	34
3.4.1 Mecanismo de la producción de la voz.....	34
3.4.2 Principales características de la voz.....	35
3.4.2.1 Información lingüística	35

3.4.2.2	Información acústica	36
3.4.3	Principios de la síntesis de voz.....	37
3.4.3.1	Conversión de texto a voz.....	38
3.4.3.2	El componente NLP	39
3.4.3.3	Analizador del texto.....	39
CAPITULO 4	43
MICROCONTROLADORES	43
4.1	INTRODUCCIÓN A LOS MICROCONTROLADORES	43
4.1.1	¿Qué es un microcontrolador?	43
4.1.2	Diferencia entre microprocesador y microcontrolador	44
4.1.3	Aplicaciones del microcontrolador	45
4.2	ARQUITECTURA DE UN MICROCONTROLADOR	46
4.2.1	Recursos comunes a todos los microcontroladores	47
4.2.2	Recursos Especiales	50
4.3	PRINCIPALES FABRICANTES	53
4.3.1	Intel.....	54
4.3.2	Freescale Semiconductor (antes Motorola)	58
4.3.3	Microchip	63
CAPITULO 5	67
SINTETIZADORES DE VOZ	67
5.1	INTRODUCCIÓN A LOS SINTETIZADORES.....	67
5.1.1	Concepto	67
5.1.2	Historia	67
5.1.3	Aplicaciones del sintetizador de voz	68
5.2	TIPOS DE SINTETIZADORES	69
5.2.1	Sintetizadores de formantes o por Regla.....	69
5.2.2	Sintetizadores mediante modelos articulatorios	69
5.2.3	Sintetizadores basados en concatenación de unidades	70
5.3	PRINCIPALES FABRICANTES	70
5.3.1	RC System	70
5.3.1.1	Sintetizador de voz V8600A.....	70
5.3.2	Winbond Electronics Corporation	73
5.3.2.1	Sintetizador de Voz I2C para robot	74
5.3.3	ScanSoft Speech Solutions	75

5.4	AVANCES EN LOS SINTETIZADORES.....	77
CAPÍTULO 6	79
IMPLEMENTACIÓN DEL SISTEMA	79
6.1	INTRODUCCIÓN	79
6.2	DESCRIPCIÓN Y ESTRUCTURA GENERAL	79
6.3	SELECCIÓN DE MATERIALES, CONSTRUCCIÓN DEL PROYECTO.....	80
6.3.1	Construcción del Guante con Sistema Sensorial.....	80
6.3.1.1	Primer Prototipo	80
6.3.1.2	Segundo Prototipo	85
6.3.1.3	Tercer Prototipo (Prototipo Final).....	90
6.3.2	Elección del microcontrolador.....	94
6.3.3	Implementación del Sintetizador de Voz.....	95
6.3.3.1	Elección y adquisición del Sintetizador	95
6.3.3.2	Operación y comandos de programación del sintetizador RC8660	97
6.3.4	Construcción física de la tarjeta de interfaz de acoplamiento.....	101
6.3.5	Construcción física de la tarjeta indicadora visual con LEDs para las posiciones de los dedos.....	105
6.3.6	Prototipo terminado del sistema traductor de señas dactilológicas a voz. .	106
CAPÍTULO 7	109
PRUEBAS Y RESULTADOS	109
7.1	PRUEBAS GUANTE SENSORIAL.....	109
7.1.1	Pruebas del primer prototipo	109
7.1.2	Pruebas del segundo prototipo.....	110
7.1.3	Pruebas del tercer prototipo (Prototipo Final).....	112
7.2	PRUEBAS DEL GUANTE SENSORIAL CON LA ETAPA DE AMPLIFICACIÓN.....	116
7.3	PRUEBAS REALIZADAS ENTRE EL GUANTE SENSORIAL Y CONVERTIDOR A/D DEL MICROCONTROLADOR.....	118
7.4	PRUEBAS DE OPERACIÓN CON EL KIT DE EVALUACIÓN DEL SINTETIZADOR RC8660	120
7.4.1	Ejemplo de prueba del sintetizador	122
7.5	INTEGRACIÓN Y PRUEBAS DE LOS ELEMENTOS DEL SISTEMA: GUANTE CON SISTEMA SENSORIAL, INTERFAZ DE ACOPLAMIENTO, MICROCONTROLADOR Y SINTETIZADOR DE VOZ.....	123
7.5.1	Mejoramiento del algoritmo para no repetir las frases.....	130

7.6 PRUEBAS DE PROTOTIPO TERMINADO DEL SISTEMA TRADUCTOR DE SEÑAS DACTILOLÓGICAS A VOZ.....	132
CONCLUSIONES.....	134
ANEXO A.....	136
ANEXO B.....	138
ANEXO C.....	141
ANEXO D.....	144
Programa 1.....	144
Programa 2.....	145
Programa 3.....	147
Programa 4.....	152
ANEXO E.....	158
ANEXO F.....	161
BIBLIOGRAFÍA.....	165
REFERENCIAS DE INTERNET.....	169

DEDICATORIA

Dedicada a los mejores padres que pude tener María Ángela y Salvador, a mis hermanas Gely y Mari, a mi esposo Carlos y mis hijos queridos Carlitos y Angelito, a mis sobrinos Sofi y Dieguito, y a mi familia preciada.

Les dedico este trabajo sin duda también a: mi Doctor Santos Regino Uscanga Sanchez y mi director de tesis M.I. Marcial Contreras Barrera.

La vida tiene mil formas de vivirse, lo sabe la mayoría, pero cuando no tienes todos los elementos para lograrlo, seguramente esa manera de vivirla se complica. Si todos fuéramos conscientes de dicho suceso, la vida tendría sentido para muchas personas con discapacidad, construyamos con nuestro conocimiento los elementos que los demás necesitan para vivir de una manera mejor. Dedicada a las personas con discapacidad.

Rocío Ortiz Domínguez

Dedico este proyecto de tesis a mis papás, a mis hermanas, a mi esposa, a mis hijos y a mi familia.

Especialmente está dedicado a todas las personas con discapacidad auditiva, especialmente a los niños, que día a día realizan sus actividades aún con las barreas de comunicación que se encuentran al interactuar con otras personas; esperando que el desarrollo de este prototipo sea una aportación importante al desarrollo de más proyectos con fines de ayudar a las personas que más apoyo necesitan para su integración y desarrollo en la sociedad.

Carlos Alberto Amaro Barbosa

AGRADECIMIENTOS

Agradezco. A mis padres: mi mamá Ángela y mi papá Salvador quienes me inculcaron los valores humanos con su ejemplo, honestidad, perseverancia, respeto, sinceridad, compromiso, tantos son que llenaría esta hoja por todo lo que me enseñaron, ellos me dieron lo necesario para lograr cualquier cosa que me propusiera, estoy orgullosa de ser su hija. Mamá y papá soy lo que soy gracias a ustedes. A mis hermanas preciadadas Gely y Mari, el mejor regalo que me dieron mis padres, les agradezco su amor, consejos y por estar conmigo en todo momento de mi vida. A mis sobrinos hermosos: Sofi, Dieguito por la felicidad que me hacen sentir cuando estoy con ellos. A mi cuñado Armando por cuidar de ellos y compartir su vida con mi hermana. A mi Abue Mari como no recordarla, a quien le agradezco porque me inculco el trabajo arduo por sobre todas las cosas y sin duda la valentía, a mis abuelitos Román, Silvestre y Tomasita.

A mis amigos de la Universidad: a mi amigocho Edgar, gracias por tu amistad y tu apoyo sin condición, a mi amigo Ángel con quien vivimos las vicisitudes de la vida durante este proyecto, gracias por tu tiempo trabajado en la tesis. A mis amigas Alice y Adri con quienes aprendí a vivir sonriendo en la FI. A mi amigo Jacobo por su amistad y cuidar de mí. Y a mis amigos Mike y Ray por su valiosa ayuda. A ustedes mis amigos les agradezco, porque sin ustedes la Facultad de Ingeniería no hubiera sido la mejor etapa de mi vida.

A mis amigas queridas Cinthya y Nancy porque puedo compartir con ellas cualquier cosa, porque comprenden la persona que soy y me dan lo mejor de ellas siempre.

A mi amigo Sergio Noble por enseñarme el mundo laboral, por enseñarme a tener éxito con constancia, tolerancia y perseverancia, por su apoyo incondicional siempre.

A mi querido amigo y Doctor Santos Regino Uscanga Sánchez, quien me enseñó que el conocimiento empleado adecuadamente salva vidas humanas. Gracias Doctor por salvar mi vida cuando Dios me permitió seguir aquí con esta vida maravillosa. Gracias a usted sigo viva, y lograré con este proyecto una meta muy importante en mi vida.

Agradezco. A mis profesores durante toda mi vida como estudiante, gracias a mi alma mater: a la “Universidad Nacional Autónoma de México”, gracias por sus conocimientos compartidos. A mi Maestro Marcial Contreras Barrera, por su asesoría durante todos estos años, el proyecto nació gracias a su ejemplo y gran apoyo. Gracias Maestro. A los profesores integrantes del jurado, por su tiempo en las revisiones necesarias para la conclusión de este trabajo.

Finalmente, agradezco. A mis adorados hijos: Carlitos y Angelito con quienes he conocido el amor puro, la inocencia de dos niños hermosos, gracias por quererme como ustedes lo saben hacer. A mi esposo, compañero y amigo Carlos Alberto por amarme, por aprender a trabajar juntos y lograr esta meta tan ansiada por los dos, gracias mi amor por ser mi inspiración y cuidar de nosotros día con día.

Rocío Ortiz Domínguez

AGRADECIMIENTOS

Agradezco. A mis papás: Inocencia y Ángel, por el gran esfuerzo dedicado a mi educación, por todos los valores que me inculcaron, por las enseñanzas y consejos que me han dado, y por ser únicos.

A mis hermanas: Angélica, Ale y Paty por su apoyo incondicional en todo lo que hago y por quererme tanto.

A mis abuelitas y abuelitos: Juanita, Juan, María y Tomás por enseñarme el valor de la familia y la fortaleza de salir adelante siempre.

A mi esposa Rocío por compartir su vida conmigo, sus sueños, su confianza y su amor. A mis hijos Carlos y Ángel por apoyarme siempre en todo y enseñarme la tantas cosas de la vida.

A todos mis tíos y tías, especialmente a mi tía Gloria por su apoyo y buenos consejos durante mi educación escolar y siempre; y a mis tíos: Juan y Gonzalo por su amistad, confianza y apoyo en todo momento. A todos mis primos y sobrinos; especialmente a: Leo, Karlita, Michelle, Erika, Lili, Clau, Dany, Hugo, Erick; y a mi cuñado David por compartir tantas cosas conmigo.

A mi amigo Udeis Alejandro por compartir sus ideas más grandes conmigo; y a mis amigos: Aarón, Zillah, Pedro, Lalo y Edgar Alejandro; por todos estos años de verdadera amistad incondicional.

A mi amigo Ángel Alfaro Ramírez por su amistad, su apoyo y dedicación para la realización y conclusión de esta tesis. A mis amigos: Edgar y Ixsacbé por ser mis amigos incondicionales siempre; a todos mis amigos del CCH y de la Facultad de Ingeniería, por su amistad y consejos.

A la Universidad Nacional Autónoma de México por permitirme estudiar y ser parte de ella.

A todos mis maestros desde la educación preescolar hasta la universidad, por todas sus enseñanzas, especialmente a la maestra Yolanda y al maestro Armando por su gran capacidad para enseñar y aconsejar.

A mi amigo Santos Uscanga por sus enseñanzas de vida y siempre dar lo mejor, y a mi amigo Sergio Noble por sus consejos, su apoyo y su confianza.

A mi director de tesis M. I. Marcial Contreras Barrera por su dedicación, paciencia y confianza hacia nosotros. A los profesores integrantes del jurado, por el tiempo invertido para la revisión y aportaciones para el mejoramiento de esta tesis.

Carlos Alberto Amaro Barbosa

1.1 INTRODUCCIÓN

La evolución de los sistemas de comunicación como forma de expresión en el hombre, depende de varios factores tales como un medio de transmisión, elementos de transducción y traducción, del aparato motor y de los sentidos capaces de producir y recibir mensajes. Los tres canales de comunicación más utilizados son: el visual, auditivo y gestual, sin embargo no todos pueden emplear estos debido principalmente a problemas de discapacidad física. En el caso de la pérdida o disminución de la capacidad auditiva conocida como sordera, las investigaciones han mostrado que ésta no afecta la capacidad intelectual del individuo ni su habilidad para aprender. No obstante, una persona con sordera pierde la estimulación lingüística que poseen aquellos sin discapacidad y adquiere retraso en el aprendizaje.

Hasta el Renacimiento se consideró que las personas con discapacidad auditiva eran incapaces de aprender a hablar y de ser educados. A partir del siglo XVI, algunos filósofos y educadores empezaron a reconsiderar esta situación. Desde esa época persiste el conflicto sobre si las personas con sordera deben ser educadas mediante procedimientos orales (lectura de labios), o manuales (lenguaje manual y de señas). Y gracias a esto actualmente es reconocido que este tipo de personas requieren de métodos opcionales de comunicación para recibir una educación adecuada.

Las personas con pérdida de la capacidad auditiva utilizan medios orales o manuales, o una combinación de ambos para la comunicación. La comunicación oral incluye lenguaje, lectura hablada y el uso de la capacidad del oído residual. La comunicación manual tiene que ver con las señas y el alfabeto manual. La comunicación total, como método de instrucción, es una combinación del método oral más las señas y alfabeto manual.

En la actualidad, se han desarrollado dispositivos para personas con alguna discapacidad que son empleados como medios de apoyo para lograr la comunicación, entre estos se encuentran los amplificadores auditivos, generadores de voz, etc. que permiten la recepción y/o transmisión de mensajes.

1.2 OBJETIVO DE LA TESIS

Diseño e implementación de un prototipo constituido por un guante con sensores basado en un sistema analógico-digital para la traducción del lenguaje manual a través de la síntesis de voz, con el fin de ayudar a que personas sordomudas se comuniquen con quienes si poseen capacidad auditiva y no conocen un lenguaje de señas. El sistema debe ser capaz de detectar determinados movimientos de la mano para generar voz a través de un sintetizador, permitiendo de esta forma la comunicación por medio del canal auditivo.

1.3 PLANTEAMIENTO DEL PROBLEMA

En la sociedad actual las personas sordomudas deben afrontar no sólo los problemas derivados directamente de su propia discapacidad, sino también aquellos que la sociedad les impone en forma de barreras de comunicación. La sociedad en general no les ofrece los medios alternativos para que accedan a la información, lo que origina una serie de problemas, como es el caso en el que un sordomudo no puede comunicarse con alguien sin discapacidad auditiva, porque éste último no conoce su lenguaje de señas. Por tal razón una persona sordomuda generalmente se aísla con otras con la misma discapacidad, formando así comunidades que difícilmente alcanzan una integración adecuada en la sociedad.

En la actualidad existen sistemas que les permiten, sino adaptarse por completo, realizar actividades que les facilitan tener una vida más adaptada a la sociedad como es el caso del lenguaje de señas, sistemas auditivos y más recientemente, los sistemas que traducen representaciones manuales a voz. En el primer caso, el individuo logra comunicarse con los demás pero existe el inconveniente de que si una de las personas no sabe el lenguaje, no se puede establecer la comunicación; la segunda clase de sistemas es muy utilizado, pero sólo funciona en aquellos que tienen sordera auditiva de menor grado. La tercera clase es la más efectiva, debido a que la persona sordomuda puede comunicarse con la mayor parte de las personas de la sociedad. Sin embargo, actualmente este último tipo de sistemas tienen un gran inconveniente al funcionar por medio de una computadora personal, ya que no se ha fabricado un sistema portátil, y por consiguiente una persona sordomuda no puede utilizarlo debido a la dificultad que implica trasladar una computadora.

Por lo tanto, debido a que el sistema traductor es muy eficiente y no se cuenta con un sistema portátil, nuestra labor es minimizar la arquitectura con base en los conocimientos y recursos con los que contamos, sin perder de vista el compromiso social que esto implica.

1.4 PROPUESTA Y RECURSOS

Para reducir y resolver el problema de comunicación entre las personas sordomudas y aquellas que poseen capacidad auditiva hemos pensado en el diseño e implementación de un sistema constituido por un guante con sensores que logre codificar el movimiento de la mano, es decir, que logre interpretar el lenguaje de señas, y de ésta manera reproducir los símbolos por medio de un sintetizador de voz.

La implementación de este sistema entre otros beneficios reduciría el número de servicios de intérprete, para que las personas con discapacidad oral puedan comunicarse con mayor facilidad.

Para comenzar con el diseño y la implementación recopilaremos información acerca de la dactilología, del lenguaje de señas y de las características de los movimientos de la mano para poder establecer las representaciones específicas del lenguaje manual que se utilizarán como entrada al sistema y así determinar la ubicación de los sensores en el guante.

Posteriormente, seleccionaremos los dispositivos de acondicionamiento de las señales analógicas generadas por el movimiento detectado por los sensores, para que sean procesadas por el módulo de adquisición de datos. Como resultado de éste proceso se obtendrán datos para ser introducidos a un sintetizador de voz capaz de producir el sonido que representa el texto de entrada.

Los sensores que proponemos utilizar en el guante son los denominados sensores de flexión; en el módulo de adquisición de datos utilizaremos una tarjeta entrenadora para el microcontrolador MC68HC11 de Freescale Semiconductor; y el sintetizador de voz lo seleccionaremos considerando que debe de utilizar el concepto de síntesis: "Texto a Voz" (Text to Speech).

1.5 ALCANCES

Con el desarrollo y aplicación práctica del guante se pretende que interactúen el lenguaje manual y la voz humana, con el fin de ayudar a que las personas sordomudas cuenten con una forma eficaz de comunicación.

Cabe mencionar que el dispositivo estará enfocado a la traducción e interpretación de un conjunto específico de posiciones fijas de los dedos de la mano por lo que solo se generarán algunas palabras, sin embargo, representará un prototipo encaminado al desarrollo de aplicaciones más complejas y con un vocabulario completo.

Lo importante es que se establecerá una forma práctica para lograr tal propósito, una idea que se concretará al lograr que se construya en serie un dispositivo analógico-digital totalmente portátil, económico y funcional. Y aunque la construcción de la aplicación se vea limitada por los recursos que estén a nuestro alcance, no perderemos de vista que el desarrollo estará enfocado en un diseño totalmente portátil.

Finalmente, con este proyecto de tesis deseamos fomentar un mayor interés en el desarrollo de aplicaciones que mejoren las condiciones de vida de personas no solo con discapacidades auditivas y orales, sino para todas aquellas que por alguna razón tienen alguna discapacidad que les impide vivir normalmente. El resultado más importante de éste trabajo será la satisfacción de poder ayudar a otras personas, a través de la implementación de nuevas tecnologías para una vida con calidad humana.

CAPITULO 2

COMUNICACIÓN Y LENGUAJE DE SEÑAS

2.1 EL ARTE DE LA COMUNICACIÓN Y LA DISCAPACIDAD AUDITIVA

Un gran paso hacia la evolución del hombre se consiguió sin duda alguna a través del arte de la comunicación. Ésta no fue fácil de establecer durante mucho tiempo, pero en cuanto el hombre tuvo la capacidad de comunicarse y pudo dejar huella de su voluntad de expresión, el progreso obtuvo una velocidad increíble.

Los hombres primitivos solamente se comunicaban con señas o señales a través de la mímica, tratando de establecer su primera simbología de comunicación. Siendo ésta de alguna manera el principio de la lengua o lenguaje.

Con el tiempo el hombre empezó a sustituir la expresión muscular de su cara, de sus manos y de su cuerpo, por la palabra que la simbolizaba y con ello se estableció un mecanismo en la comunicación que progresaría muy rápido, hasta llegar a la creación del lenguaje; el lenguaje simbólico que habría de escribirse al principio de manera primitiva, como la escritura cuneiforme, los jeroglíficos, y más tarde la escritura hebrea, fenicia y grecolatina, que habría de llegar hasta nuestros días. El lenguaje se enriqueció con el paso de cada generación y con ello la posibilidad de establecer una comunicación de dos vías (emisor-receptor).

La comunicación se puede realizar mediante cualquiera de nuestros sentidos (*oído, tacto, gusto, olfato, vista*). En la vida cotidiana la comunicación se lleva a cabo con un simple apretón de manos, mediante una mirada o de muchas otras maneras. Sin embargo, todas ellas carecen de la complejidad necesaria que entraña el lenguaje oral, el cual representa el sistema más completo para definir y expresar una relación sin crear confusiones. [2.8]

El emisor crea, elabora, produce y emite las señales con la pretensión de provocar la comprensión en el receptor. En la transmisión de los estímulos intervienen las condiciones del “canal de la comunicación” que los puede transmitir con más o menos alteraciones. El receptor recibe y percibe estos estímulos según las condiciones de sus capacidades, los elabora, es decir, los descodifica y los manda hacia las zonas del sistema nervioso central (SNC) donde son reconocidos y comprendidos. [2.8]

Una de las condiciones básicas para realizar la comunicación es el conocimiento mutuo, tanto por parte del emisor como del receptor, del sistema abstracto de señas

utilizado; este sistema tan abstracto es el lenguaje y su materialización concreta en la mayoría de las ocasiones, es el habla. El emisor y el receptor deben conocer el mismo lenguaje, entre más similar sea, más exitosa será la comunicación. [2.8]

Así pues, los componentes fundamentales de la comunicación se pueden presentar destacando cuatro factores:

- Emisor
- Canal de la comunicación
- Ruido en el canal de la comunicación
- Receptor [2.8]

Gracias a la comunicación, la sociedad humana tiene como uno de sus propósitos centrales la tarea de integrar a todos sus componentes, de tal manera que compartan un objetivo común. La integración debe enfrentar el enorme desafío de considerar y respetar la diversidad de los seres humanos que forman parte de la sociedad: diversidad de ideas, de vocaciones y de capacidades concretas. En ésta última se incluyen aquellos grupos humanos que presentan síntomas parciales o totales de discapacidad, como es el caso de personas con discapacidad auditiva, también conocida como sordera.

La *discapacidad auditiva* se puede definir como la dificultad o imposibilidad de utilizar el sentido del oído, que impide la realización de actividades el margen de la “normalidad”. [2.5]

En el año 2006, en la publicación de la Convención sobre los Derechos de las Personas con Discapacidad de la ONU, se establecía que la discapacidad es un concepto en constante evolución y, además, “resulta de la interacción entre las personas con impedimentos y las barreras actitudinales y del entorno que evitan su participación plena y efectiva en la sociedad, en pie de igualdad con los demás”. [2.5]

Datos del censo de población y Vivienda 2010 revelan que poco más de 5 millones de personas padecían de discapacidad auditiva en nuestro país, en su mayoría mujeres. El grupo de edad que presenta un mayor reporte de casos es el comprendido entre los 70 y los 84 años, el de los adultos mayores (INEGI, 2010). Estas cifras representan la medición de indicadores que nos hablan de esta condición como un factor que limita la vida de las personas sordas y, por lo tanto, las discapacita e impide la correcta interacción con la sociedad oyente; sin embargo, no mencionan si la sordera es prelocutiva o poslocutiva (estos conceptos se explicarán más adelante). También hay otros indicadores importantes como los de formación académica: la mayoría de los sordos en proceso de formación académica se encuentran inscritos en el nivel básico, es decir, el 55.1% del total de sordos en México, que el 5.4% cursan el nivel medio superior y apenas un 4.1% el nivel superior (de acuerdo con cifras del INEGI, 2010). [2.5]

La Comunidad sorda se integra por un grupo heterogéneo de personas que tienen la lengua de señas como su principal medio de comunicación, y con ella una cultura particular que se distingue de las comunidades oyentes donde los Sordos habitan.

Uno de los principales problemas a los que se enfrenta una persona con sordera o deficiencia auditiva es la *adquisición del lenguaje*. El lenguaje y la audición están estrechamente relacionados y recíprocamente influenciados y condicionados. [2.8]

Las condiciones necesarias para la adquisición del lenguaje oral son las siguientes:

- Tener *capacidades psicofísicas* que permiten la creación de la competencia lingüística.
- *Oír* el habla humana.
- *Escuchar* un tiempo suficientemente largo.
- Utilizar el habla como *primer sistema* de comunicación.
- Tener la *necesidad* y la *oportunidad* para la expresión verbal. [2.8]

La primera condición para la adquisición del lenguaje oral es poder oírlo, como modelo del sistema de comunicación de una determinada comunidad lingüística. La adquisición del lenguaje depende de una serie de condiciones personales y ambientales. [2.8]

Una de las funciones principales de la audición es la recepción del lenguaje hablado, por medio de ésta, una persona adquiere el lenguaje al recibir los sonidos articulados, lo que le permite llevar a cabo la comunicación humana, sistema fundamental para integrarse al medio social. Aquel que no puede lograr la comunicación, no consigue espontáneamente dicha integración, y entra a pertenecer al mundo de la sordera. Un mundo en el que la pérdida auditiva no implica simple y exclusivamente la necesidad de compensar esta deficiencia con las aportaciones de otras vías sensoriales, sustituir por ejemplo, la comunicación oral por la visual; sino que puede tener también importantes repercusiones en la adquisición del lenguaje, en la relación con el entorno de la persona afectada y, por tanto, en su organización psíquica. [2.4]

Cuando la persona sorda puede codificar la información de origen sonoro visualmente, como el lenguaje oral emitido por una persona de la que ve su rostro, ello le plantea la necesidad de prestar atención visual a la cara del hablante, y a la vez a los estímulos de imagen propiamente visuales que constituyen el contexto del discurso. [2.4]

También, en determinados casos, principalmente en los de pérdida auditiva profunda, la persona sorda puede tener *problemas para hacerse comprender por lo oyentes*, especialmente por aquellos que no están habituados a reconocer su emisión oral, que puede caracterizarse por ciertas deformaciones fonéticas, un ritmo anómalo y el no respetar los aspectos suprasegmentales del lenguaje oral, rasgos, todos ellos, que pueden dificultar la inteligibilidad. [2.4]

Son muchas las situaciones de la vida cotidiana en las que las *informaciones son exclusivamente auditivas*: mensajes por megafonía en lugares públicos, comunicaciones telefónicas, timbres de la puerta, etc. En todas ellas la persona sorda precisa compensaciones visuales y, si éstas no se producen, pueden ocurrir alteraciones en su vida social con los oyentes. Según la sensibilidad y el desarrollo tecnológico de cada formación social, hallamos un nivel determinado de *eliminación de las barreras de comunicación*. En la mayoría de los casos, la eliminación de dichas barreras no significa

una gran inversión económica. Se trata principalmente de tener presente las necesidades de un grupo social minoritario y utilizar la información auditiva con la visual de una manera adecuada. Por ejemplo, en los espacios de convivencia de personas de distintas lenguas y nacionalidades (como es el caso de los aeropuertos) la información visual prevalece sobre la auditiva.

2.2 GRADOS DE SORDERA Y APROPIACIÓN DE LA LENGUA ESCRITA

El grado de pérdida auditiva afecta fundamentalmente los mecanismos de recepción y producción del lenguaje oral y, en el caso de sordera anterior a éstos, afecta la adquisición del lenguaje y los procesos que implica. [2.4]

El diagnóstico audiológico es imprescindible para conocer con exactitud el grado de pérdida auditiva que presenta una persona. Una vez explorada la sensibilidad auditiva para cada una de las frecuencias que componen el audiograma de cada oído, el grado de pérdida auditiva se calcula a partir de la media aritmética de la cantidad de decibelios (dB) necesarios para la audición de los sonidos de cada una de las siguientes frecuencias conversacionales: 500, 1000 y 2000 Hz. [2.4]

Es importante subrayar que el grado de pérdida auditiva constituye un factor importante en el desarrollo del lenguaje; determina los *mecanismos de recepción y producción*, pero no la capacidad lingüística que depende de la interacción de muchos otros factores. [2.4]

La deficiencia auditiva se puede clasificar en función de muy diversos criterios, describiremos las cuatro más comunes. [2.8]

Teniendo en cuenta la *localización de la lesión* podemos hablar de:

- Las sorderas de transmisión o conductivas: en las que está afectado la parte mecánica del oído (oído externo y oído medio), lo que impide que el sonido llegue a estimular correctamente al órgano de Corti, que es el órgano receptor de la audición.
- Las sorderas de percepción o neurosensoriales: son aquellas en las que la lesión se localiza en el órgano de Corti y/o en la vía auditiva central.
- Se habla de sordera mixta cuando existe una asociación de sordera de transmisión y de percepción. [2.8]

Por otro lado, teniendo en cuenta la intensidad de la pérdida auditiva, y siguiendo las normas de la oficina Internacional de audiofonología, BIAP (*Bureau International d' Audiophonologie*), se establecen cuatro grupos:

- Sorderas leves (21-41 dB HL).
- Sorderas medias (41-70 dB HL).
- Sorderas severas (71-90 dB HL).
- Sorderas profundas (>90 dB HL). [2.8]

“En cuanto al *momento de la aparición* del déficit auditivo y teniendo en cuenta la adquisición del lenguaje, se distinguen:

- Sorderas prelocutivas o prelinguales, que aparecen antes de la aparición del lenguaje (0–2 años).
- Sordera perilocutivas, que aparecen entre el segundo y el cuarto año de vida.
- Sorderas postlocutivas o postlinguales, que se instauran después de que las adquisiciones lingüísticas estén consolidadas.

Las sorderas prelocutivas y perilocutivas cuando son bilaterales y de intensidad severa o profunda interfieren o impiden el desarrollo del lenguaje oral.”
[2.8]

Por su origen, la sordera puede ser congénita o adquirida; en el ámbito de las hipoacusias neurosensoriales, de acuerdo al momento de su aparición se clasifican en:

- Prenatales (genéticamente determinadas o adquiridas).
- Perinatales.
- Postnatales. [2.8]

La consecuencia de la sordera sobre el desarrollo del lenguaje en el niño va a depender de la importancia de la pérdida auditiva y de la edad de aparición de la misma.
[2.8]

La lengua escrita debería constituir una vía de comunicación privilegiada para la persona sorda, a fin de compensar las deficiencias que padece en otras fuentes informativas, tales como la radio, el cine y la televisión, e incluso en la simplificación que se produce con frecuencia en las conversaciones con el entorno de los oyentes. Aunque no exista una comunicación directa entre la sordera y dificultades en la lengua escrita, lo cierto es que es necesario un determinado dominio del sistema lingüístico oral para acceder al escrito, y ello sería una primera explicación de las dificultades que experimenta una gran parte de personas con sordera en la lectura y escritura. [2.4]

El lenguaje de señas, junto con el oral, puede resultar de gran utilidad en algunos casos y en determinados momentos evolutivos; la apropiación de la lengua escrita es el principal objetivo para el acceso a la cultura de la persona afectada por la sordera, y esto conlleva, de una u otra forma, al acceso del lenguaje oral. [2.4]

2.3 SISTEMAS DE COMUNICACIÓN GESTUAL

En su relación con los oyentes, el sordo profundo tiene tendencia a utilizar gestos para expresar sus intenciones. Solicita, directamente o no, el mismo medio de comunicación de parte de su entorno. Se establece de este modo un sistema simbólico elemental que mejora la relación social cotidiana. En general son gestos espontáneos que sirven de base a este modo de comunicación, a los que vienen a añadirse mímicas, gestos explicativos y descriptivos.

Para todos los sordos el gesto no es universal, aunque los signos descriptivos que han evolucionado son inteligibles a la mayoría de ellos. En los institutos de sordos aparecen numerosos *dialectos* a veces lo bastante distintos para hacer difícil la comprensión de la ideas.

Desde hace doscientos años, a esta comunicación se le llamaba el lenguaje de los signos. Convertido, con su paso a Estados Unidos, en *Sign Language*, el lenguaje de señas, traducción frecuentemente marcada por la influencia de psicólogos y lingüistas.

Las primeras investigaciones realizadas en las comunidades de sordos en Estados Unidos establecieron los *parámetros formativos* en la organización del lenguaje de señas utilizado por dichas comunidades. Existen los siguientes parámetros: el lugar donde se ejecuta la seña, la configuración de la mano y el movimiento que se realiza. [2.4]

En un estudio sistemático desde el punto de vista lingüístico del lenguaje de señas español, realizado en España en 1992, se señalan 6 parámetros formativos de la seña gestual: los *queíremas* o configuraciones de la mano (abierta, cerrada, índice y pulgar levantados, etc.), de los cuales se distinguen 29; los *toponemas* o lugares en que se realiza la seña (ante la frente, los labios, etc.), de los que se distinguen 25; los *kinemas* o movimientos de la mano (se distinguen 18); los *kineprosemas* o direcciones de los movimientos de la mano (se distinguen 6); los *queirotropemas* u orientaciones de la mano (se distinguen 9), y finalmente, los *prosonemas* o expresiones de la cara. [2.4]

Atendiendo a las relaciones semánticas básicas, las señas gestuales se clasifican en las siguientes categorías: *señas motivadas*, *intermedias* y *arbitrarias*. [2.4]

Las *señas motivadas* expresan una relación muy estrecha con la realidad, ya sea porque reproducen la forma, el movimiento o alguna relación espacial de los referentes representados (signos iconocinéticos), o, sea porque los indican directamente (gestos deícticos). En las señas intermedias se sitúan aquellas que se basan en la dactilología o los algoritmos aritméticos. Finalmente, las señas arbitrarias son aquellas que no permiten ver su relación con el referente, aunque muchos de ellos sean el resultado de un proceso de abstracción icónica. [2.4]

El lenguaje de señas es muy útil y altamente efectivo como herramienta para facilitar los casos de comunicación de personas con dificultades auditivas. En la mayoría de sus formas, este lenguaje tiene muchas semejanzas con su contraparte hablada, lo más llamativo es que también es un lenguaje vivo.

2.4 EL LENGUAJE DE SEÑAS, LA DACTILOLOGÍA Y POSICION DE LA MANO

Existe muy poca información sobre el origen de las lenguas de señas. Algunos autores colocan este origen en una época previa al nacimiento de las lenguas orales. Concretamente, defienden teorías que colocan el origen de la capacidad lingüística humana en una lengua de señas que, posteriormente, fue reemplazada por la lengua oral. [2.7]

“Si no tuviéramos voz ni lengua y nos quisiéramos manifestar recíprocamente las cosas, ¿acaso no intentaríamos, como ahora los sordos, manifestarlas con las manos, la cabeza y el resto del cuerpo?”, es una de las más antiguas referencias que Platón, en su diálogo Crátilo, pone en boca de Sócrates –su maestro- una reflexión sobre el lenguaje. [2.7]

Hasta el siglo IV no volvemos a encontrar comentario sobre la comunicación gestual en los escritos de San Agustín, donde se maravilla de la comunicación que puede establecerse entre los sordos a través de los gestos y eleva éstos a la categoría de señas, al mismo nivel que las palabras. [2.7]

Durante el Renacimiento, el medio que usan las personas sordas para comunicarse vuelve a ser celebrado en el *Tratado de Pintura* de Leonardo da Vinci. A partir de éste movimiento ha evolucionado la lengua de señas y se ha reconsiderado la educación de los sordos con las aportaciones de filósofos y educadores, por mencionar algunos: Leonardo da Vinci, Fray Pedro Ponce de León, Juan Pablo Bonet, Jacobo Rodrigues Pereira, Abad de L'Epée, Roch Ambroise Sicard, Auguste Bébien, J.M.G. Itard, Louis Laurent Marie Clerc, Thomas Hopkins Gallaudet, Edward Miner Gallaudet, Lorenzo Hervás y Panduro, Roberto Francisco Prádez, Francisco Fernández Villabrille, que fundó la *Revista de sordomudos*, publicó el primer diccionario de lengua de señas española LSE y *Diccionario de mímica y Dactilología*; y William Strokoe que publicó *Sign Language Structure y A Dictionary of American Sign Language*. [2.7]

La Federación Mundial de Sordos (por sus siglas en inglés WFD) estima que la información sobre las lenguas de señas y sus usuarios no es precisa, pero representa alrededor de 70 millones de personas, de esta población se agrupa a personas con algún déficit auditivo, y sólo un número determinado de ellas tendrían la lengua de señas como su primera lengua, o como la lengua en la que basara su intercambio comunicativo. [2.5]

Como hemos podido observar, en el mundo existen varias lenguas de señas que son utilizadas. En México existen dos: la Lengua de Señas Mexicana (LSM) y la Lengua de Señas Maya (LSMy). [2.5]

Para reforzar la comprensión de la lengua, desde los mismos tiempos en que se utilizaba la de las señas, se viene empleando un deletreo de las palabras por la posición de los dedos y de las manos, *la dactilología*. Se utiliza también en la comunicación de la lengua de señas cuando la palabra no existe, ni de manera aproximativa, como es el caso de los nombres propios o vocabulario especializado, tal como el que se encuentra en las ciencias. Éste modo de expresión es, por tanto, un sistema complementario basado en la lengua escrita en toda su complejidad ortográfica.

La dactilología (deletrear con los dedos) es también conocida como un alfabeto visual o manual. Es el alfabeto que, en otros tiempos, daba el “pequeño Larousse” como modo de comunicación de los sordomudos, un alfabeto gestual. Cada una de las letras del alfabeto tiene una configuración o representación manual distinta. Muchas de estas

configuraciones o representaciones manuales se crearon para parecerse a las formas escritas de las letras del alfabeto. La mayoría de las personas utilizan la dactilología mucho más lentamente que la velocidad con la que pueden hablar, por lo tanto, disminuye la velocidad del habla y de la comunicación en una manera anormal.

En el Sistema Dactilológico como forma o lenguaje de comunicación, uno de los principales problemas que se detectan está en las diversas formas en que las personas sordomudas (en distintos puntos del mundo, pero incluso entre los residentes en una misma localidad), configuran las letras del alfabeto.

Debido a esta gran dispersión geográfica de las personas sordomudas por todo el mundo, añadido al hecho de tratarse de un grupo no muy numeroso en comparación con el resto de la población mundial no sordomuda, siempre se ha considerado de gran importancia para su unidad que todos ellos tuvieran un sistema de comunicación común y unificado, independientemente de otros sistemas de comunicación que puedan conocer y utilizar. Sin embargo, la dactilología continúa variando de un país a otro y de una región a otra. Aun así, en cada región o país se han creado comisiones formadas por personas sordomudas que trabajan en la unificación de un alfabeto dactilológico.

Los principales aspectos que se deben tener en cuenta a la hora de seleccionar una de las variantes como configuración oficial son: la claridad para la captación por el tacto y la comodidad del interlocutor que lo utilice, evitando posturas forzadas de la mano al deletrear, entre otros. Por otra parte, el alfabeto dactilológico aunque suele sufrir constantes cambios en muy pocas letras, se mantiene muy similar al empleado por las personas sordas.

Así por ejemplo, el alfabeto dactilológico mexicano al igual que el español es un sistema unimanual, es decir, que se realiza configurando cada una de las letras del alfabeto con una sola mano, mientras que en algunos países se utilizan ambas manos para una sola letra (*bimanual*).

El sistema consiste en deletrear cada letra del mensaje sobre la palma de la mano de la persona sordomuda que recibe, de la manera más cómoda y rápida posible, asegurando siempre su captación táctil por la persona sordomuda.

La colocación más cómoda sería con ambos interlocutores sentados frente a frente. La persona que emite el mensaje toma con su mano libre la mano de la persona sordomuda que está recibiendo el mensaje (suavemente y sin apretar, pero procurando que esté firme y no se mueva) mientras con la otra mano se configuran las letras.

La palma de la mano de la persona sordomuda debe estar hacia arriba, para deletrear sobre ella el mensaje. Las letras deben ejecutarse en el centro de la palma, letra tras letra y haciendo una breve pausa entre cada palabra. Las letras deben ser correctamente formadas, pretendiendo una caligrafía perfecta. Como es de gran importancia ver el movimiento de los labios, la mano no debe de ser sostenida en una posición tal que bloquee la vista de la boca; tampoco debe ser sostenida tan lejos de la boca que los labios no puedan ser leídos simultáneamente al lenguaje manual. [2.6]

En la *Figura 2.1* se muestra la descripción gráfica de cada una de las letras del Alfabeto dactilológico empleado por personas sordomudas como estándar en gran parte de los países de habla hispana.



Figura 2.1 Alfabeto dactilológico [W2.3]

Gracias a los avances a favor de las personas con discapacidades auditivas, actualmente son consideradas alrededor de 103 Lenguas de Señas distintas en todo el mundo. Las personas que usan ese tipo de lenguas son muchas veces monolingües, desconociendo cualquier otro tipo de lenguaje, incluida la lengua nacional que se habla en su entorno geográfico.

El conjunto de señas de los sordos, en lo que respecta a las letras, es tal como lo vemos en la *Figura 2.2*. Estas señas han sido seleccionadas cuidadosamente acatando un estándar en cuanto a la Ciudad de México se refiere. [W2.1]

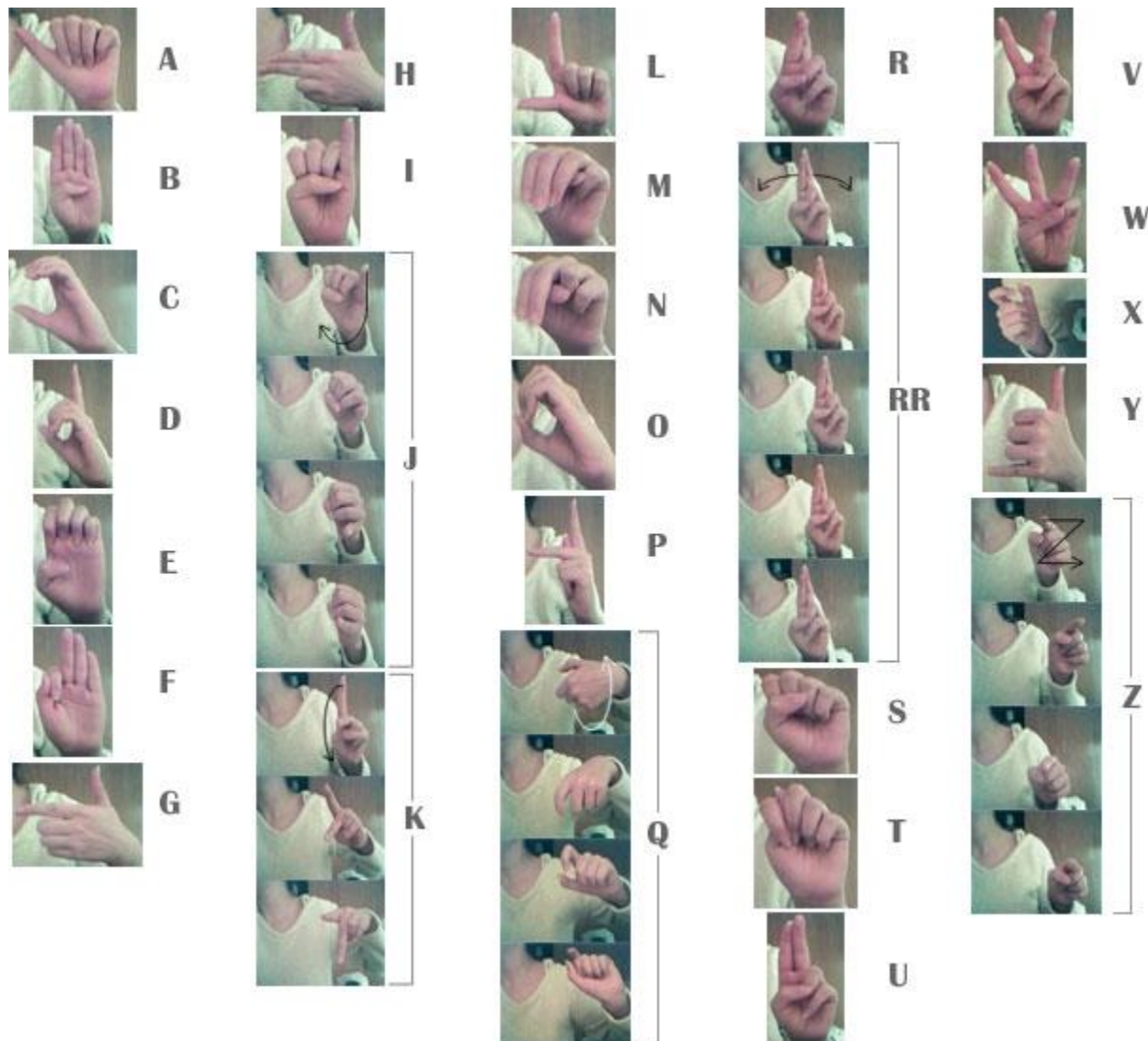


Figura 2.2 Imágenes de la mano que representan letras en el sistema de señas mexicano. [W2.4]

Cabe mencionar que junto a estos aspectos de la comunicación por gestos, el lenguaje de las señas y dactilología, han sido escritos diversos métodos que requieren la palabra en su estructura fonética y fonológica, en el curso de estos últimos decenios. No se basan en la lengua oral y escrita, sino en la expresión de la palabra.

2.5 LA MANO HUMANA

La mano del hombre cuenta con cuatro dedos similares entre sí (índice, medio, anular y meñique), siendo su longitud la única diferencia entre ellos. Los cuatro cuentan con tres falanges, llamadas respectivamente, desde la más próxima a la palma, primera falange o falange proximal; segunda falange o falange medial; y, tercera falange o falange distal.

Las articulaciones de estos dedos son conocidas como articulaciones interfalángicas distales y proximales. Las primeras permiten a la falange medial y a la falange distal tener un movimiento relativo entre ellas de flexión y de extensión, y la segunda permite tener los mismos movimientos entre la falange medial y la falange proximal. En la articulación metacarpofalángica las falanges proximales se acoplan con los huesos del metacarpo para lograr dos movimientos: la flexión y extensión, y la aducción y abducción (*Figura 2.3*).

El dedo pulgar solo cuenta con falange proximal y falange distal, es decir, que carece de falange medial, por lo que solo tiene una articulación interfalángica, y las articulaciones metacarpofalángica y la trapeciometacarpal (*Figura 2.4*). Las dos primeras permiten que este dedo tenga movimientos de flexión y extensión, mientras que la tercera permite, además, los movimientos de aducción y abducción. Este dedo está dispuesto de una forma muy distinta a la de los otros cuatro para lograr que éste se anteponga sobre la palma de la mano al movimiento de estos cuatro; movimiento que se llama de oposición.

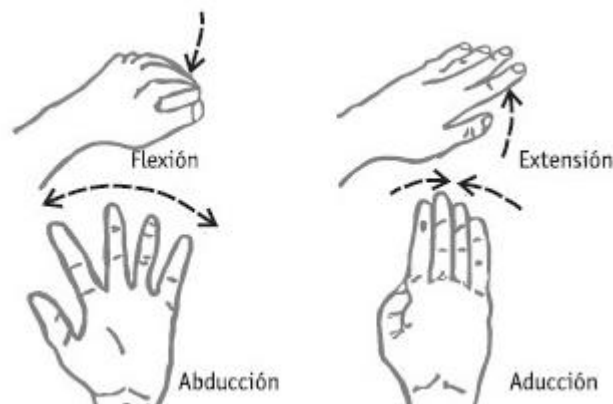


Figura 2.3 Movimientos de la mano humana [W2.5]

En términos generales, se habla de que la mano humana tiene 22 grados de libertad. Una en cada una de las articulaciones interfalángicas proximales de los cinco dedos; dos en cada una de las articulaciones metacarpofalángicas de los dedos índice, medio, anular y meñique (8 en total) y sólo una en la correspondiente del dedo pulgar; dos en la muñeca o carpo; una pequeña rotación en cada una de las articulaciones carpometacárpicas (4 en total), y dos más en la trapeciometacarpal.



Figura 2.4 Estructura ósea de la mano humana **

3.1 SISTEMAS DE MEDIDA

La mayoría de los eventos del mundo real y sus medidas son analógicas, lo que implica que las medidas pueden tomar un rango de valores casi continuo. Las cantidades físicas de interés pueden ser diversas como: temperatura, presión, intensidad luminosa, fuerza, velocidad, posición, etc. y pueden manipularse de acuerdo a los requerimientos de cada aplicación. Existen algunos sistemas para la manipulación de estas variables, entre los que destacan los sistemas de medición o también llamados sistema de medida y los sistemas de adquisición de datos (SAD).

En los sistemas de medida la variable se mide y su valor se visualiza para conocer sus características y comportamiento. Los bloques que integran éste sistema son: fuente de alimentación, detección de variable, acondicionamiento de señal y presentación y/o visualización. En la *Figura 3.1* se muestra el diagrama de bloques de un sistema básico de medida.

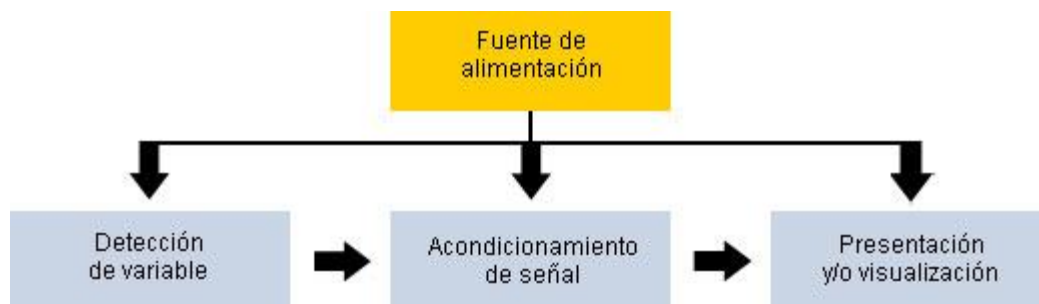


Figura 3.1 Diagrama de bloques de un sistema de medida. [3.2]

La fuente de alimentación suministra la energía eléctrica a todos los elementos del sistema que la requieran; el módulo de detección de variables sensa la variable a medir por medio de un transductor; el acondicionamiento de señal tiene la función de adecuar la señal de un transductor mediante un circuito, que dependiendo de la aplicación específica,

puede amplificar, acoplar las impedancias, aislar, filtrar, etc. la señal con el fin de obtenerla adecuadamente para el medio de presentación; por último una vez que se logra una señal con el nivel y las características apropiadas, puede ser enviada a diversos dispositivos de presentación que pueden ser desde un voltímetro hasta un osciloscopio para su observación y registro de datos. [3.2]

Por otra parte el sistema de adquisición de datos (SAD) es muy similar al sistema de medición, debido a que ambos funcionan a partir del mismo principio. La diferencia está en que en el sistema de medición sólo se puede visualizar la variable capturada y en el SAD se puede procesar información en un dispositivo digital (microcontroladores, microprocesadores, etc.). Los módulos SAD son muy importantes en Ingeniería y no debemos olvidar que ambos tienen el mismo dispositivo de entrada, el transductor.

3.2 TRANSDUCTOR Y SENSOR

Se denomina transductor, en general, a todo dispositivo que convierte una señal de una forma física en una señal correspondiente pero de otra forma física distinta. Es por lo tanto, un dispositivo que convierte un tipo de energía en otro. En la práctica, no obstante, se consideran transductores por excelencia aquellos que ofrecen una señal de salida eléctrica. Ello se debe al interés de este tipo de señales en la mayoría de procesos de medida y la facilidad que presentan para ser manipuladas. [3.3]

Un sensor, es capaz de detectar cualquier tipo de acción externa y generar una respuesta ante ésta. Comúnmente, un sensor transforma magnitudes físicas o químicas en magnitudes eléctricas.

Sensor y transductor se emplean a veces como sinónimos, sin embargo un sensor es considerado como una ampliación de los sentidos, para detectar cantidades físicas que por su naturaleza o tamaño no pueden ser percibidas directamente por nuestros sentidos. Puede decirse también que es un dispositivo que aprovecha una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo. [3.3]

La distinción entre transductor de entrada (señal física/señal eléctrica) y transductor de salida (señal eléctrica/presentación) está prácticamente en desuso. La tendencia actual, particularmente en robótica, es emplear el término sensor para designar el transductor de entrada, y el término actuador o accionamiento para designar el transductor de salida. Los primeros pretenden la obtención de información, mientras que los segundos buscan la conversión de energía. [3.3]

3.2.1 Clasificación de los transductores

El número de transductores disponibles para las distintas magnitudes físicas es tan elevado que no se puede proceder racionalmente a su estudio sin clasificarlos previamente de acuerdo con algún criterio. Los transductores se pueden clasificar de acuerdo a su principio de operación, aplicación, método de conversión de energía, tipo de señal de entrada o de salida, etc. [3.3]

Según el aporte de energía, los transductores se pueden dividir en moduladores y generadores. En los transductores moduladores o activos, la energía de la señal de salida procede, en su mayor parte, de una fuente de energía auxiliar. La entrada sólo controla la salida. En los transductores generadores o pasivos, en cambio, la energía de salida es suministrada por la entrada. [3.3]

Los transductores moduladores requieren en general más hilos que los generadores, ya que la energía de alimentación suele suministrarse mediante hilos distintos a los empleados para la señal. Además, esta presencia de energía auxiliar puede crear un peligro de explosiones en algunos ambientes. Por el contrario, su sensibilidad se puede modificar a través de la señal de alimentación, lo que no permiten los transductores generadores. [3.3]

Según la señal de salida, los transductores se clasifican en analógicos o digitales. En los primeros la salida varía de forma continua. La información está en la amplitud, si bien se suele incluir en este grupo los transductores con salida en el dominio temporal. Si es en forma de frecuencia, se denominan, a veces, “casi digitales”, por la facilidad con que se puede convertir en una salida digital. [3.3]

En los transductores digitales, la salida varía en forma de saltos o pasos discretos. No requieren conversión A/D y la transmisión de su salida es más fácil. Tienen también mayor fidelidad y mayor fiabilidad, y muchas veces mayor exactitud, pero lamentablemente no hay modelos digitales para muchas de las magnitudes físicas de mayor interés. [3.3]

De acuerdo al modo de funcionamiento, los transductores pueden ser de deflexión o de comparación. En los transductores que funcionan con deflexión, la magnitud medida produce algún efecto físico, que engendra algún efecto similar, pero opuesto, en alguna parte del instrumento, y que está relacionado con alguna variable útil. [3.3]

En los transductores que funcionan por comparación, se intenta mantener nula la deflexión mediante la aplicación de un efecto bien conocido, opuesto al generado por la magnitud a medir. Hay un detector del desequilibrio y un medio para restablecerlo. [3.3]

A continuación se muestra una tabla con la clasificación de los transductores, así como algunos ejemplos para cada caso. *Ver Tabla 3.1.*

Criterio	Clases	Ejemplos
Aporte de energía	Moduladores Generadores	Termistor Termopar
Señal de salida	Analógicos Digitales	Potenciómetro Codificador de posición
Modo de operación	De deflexión De comparación	Acelerómetro de deflexión Servo acelerómetro

Tabla 3.1 Clasificación de transductores. [3.3]

Según el tipo de relación entrada-salida, los transductores pueden ser de orden cero, de primer orden, de segundo orden o de orden superior. El orden está relacionado con el número de elementos independientes que almacenan energía incluidos en el transductor, repercutiendo en su exactitud y velocidad de respuesta. Esta clasificación es de gran importancia cuando el transductor forma parte de un sistema de control de lazo cerrado. [3.3]

Desde el punto de vista de ingeniería electrónica, es más atractiva la clasificación de los transductores de acuerdo con el parámetro variable: resistencia, capacidad, inductancia, añadiendo luego los transductores generadores de tensión, carga o corriente, y otros tipos no incluidos en los anteriores grupos. Si bien este tipo de clasificación es poco frecuente permite reducir el número de grupos a unos pocos y se presta bien al estudio de los acondicionadores de señal asociados. La clasificación correspondiente se muestra a continuación, mostrando el parámetro variable y ejemplos de cada uno de ellos [3.3]:

Resistencia [3.2]:

- Dispositivo potenciométrico
- Galga extensiométrica resistiva
- Medidor de alambre caliente o medidor Pirani
- Termómetro de resistencia
- Termistor
- DTR (Detector de Temperatura Resistivo) [3.3]
- Higrómetro de resistencia
- Celda fotoconductiva

Capacidad:

- Medidor de presión de capacidad variable
- Micrófono de capacitor
- Medidor dieléctrico [3.2]

Inductancia:

- De circuito magnético
- Detector de reluctancia
- Transformador Diferencial Lineal Variable (LVDT)
- Medidor de corriente parásita
- Medidor de magnetostricción [3.2]

Voltaje y Corriente:

- Cámara de ionización
- Celda fotoemisiva
- Tubo fotomultiplicador
- Termopar y termopila

- Generador de bobina móvil
- Detector piezoeléctrico
- Celda fotovoltaica [3.2]

De la clasificación anterior, los transductores basados en la variación de la resistencia eléctrica de un dispositivo son probablemente los más abundantes, debido a que son muchas las magnitudes físicas que afectan al valor de la resistencia eléctrica de un material. En consecuencia, ofrecen una solución válida para numerosos problemas de medida. A continuación veremos algunos de ellos en detalle. [3.3]

3.2.2 Potenciómetros

El potenciómetro (resistor variable) es frecuentemente acoplado mecánicamente para mediciones de desplazamiento. Puede conducir cualquiera de las señales de CA o CD y usualmente no requiere de un amplificador. Este dispositivo no es costoso, pero no puede utilizarse en aplicaciones de alta velocidad, limitando su exactitud, capacidad de repetición y tiempo de vida, debido al desgaste mecánico del material resistivo activo. [3.6]

Los potenciómetros, tienen su principio en la propiedad resistiva de los materiales, que es proporcional a la superficie de éste. [3.2]

Los potenciómetros cuentan con una escobilla o contacto deslizante que se mueve sobre un elemento resistivo, que está unido directamente al eje del sensor. Existen dos tipos de potenciómetros: con desplazamiento lineal y de desplazamiento angular, mostrados en las Figuras 3.2 (a) y 3.2 (b). [3.2]

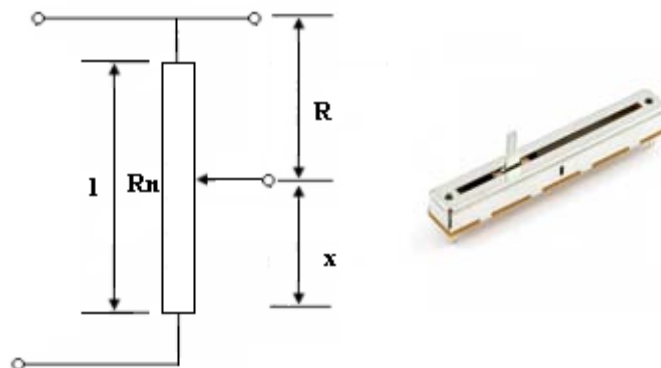


Figura 3.2 (a) Potenciómetro Lineal [3.3]

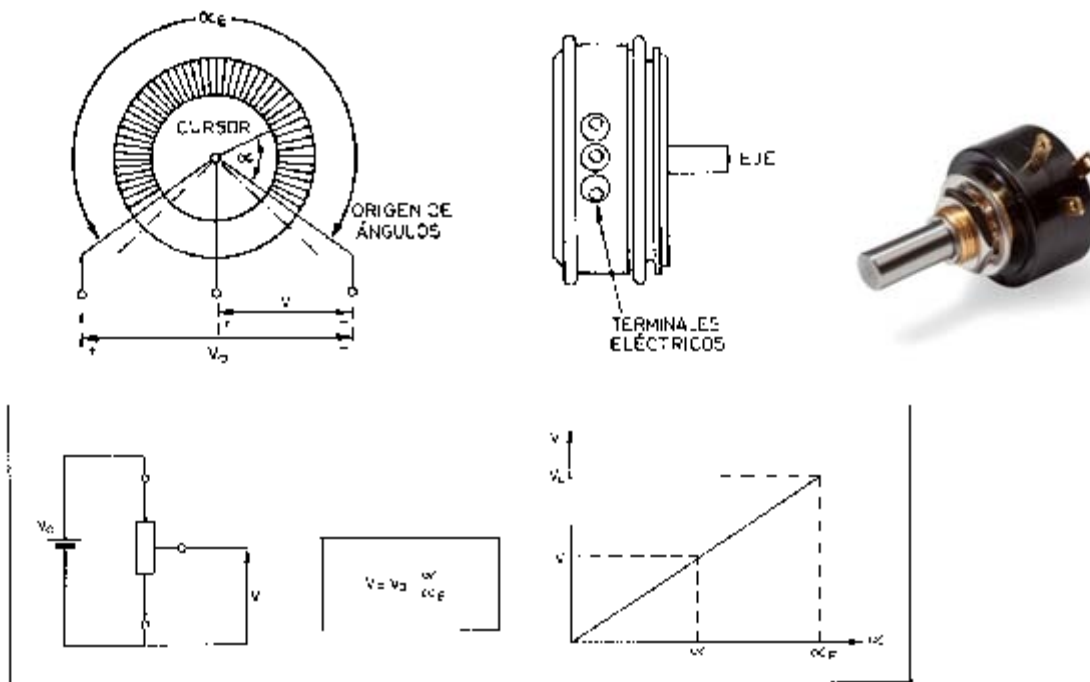


Figura 3.2 (b) Potenciómetro angular [3.3]

3.2.3 Sensor de flexión

El sensor de flexión es un dispositivo constituido por una cubierta de sustrato similar a un plástico que cambia su conductividad eléctrica cuando es flexionado. Sistemas electrónicos pueden ser conectados al sensor y medir con precisión la cantidad de movimientos que ocurren. Un solo movimiento de aproximadamente 1 pulgada (2.54 cm) puede arrojar alrededor de 200,000 datos.

Un ejemplo de alguna aplicación es cuando un sensor es colocado en una puerta. Cuando la puerta es abierta se puede medir a que distancia se encuentra y que tan rápido fue su movimiento. Un sensor de este tipo es muy pequeño, ligero, de fácil empaquetado y muy exacto. Ver Figura 3.3 (a) y 3.3 (b).

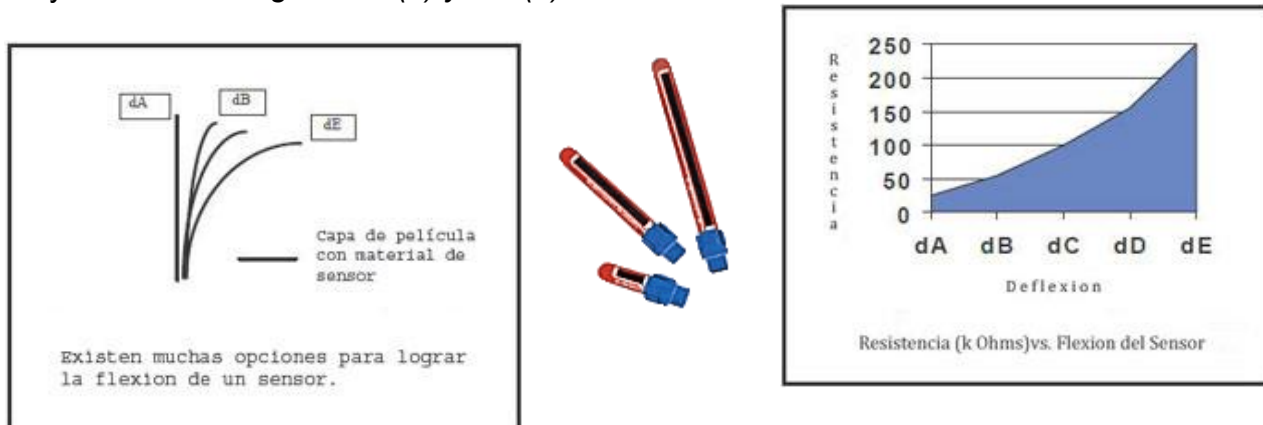


Figura 3.3 (a) Flexión del sensor

Figura 3.3 (b) Gráfica del comportamiento del sensor

3.3 SISTEMA DE ADQUISICIÓN DE DATOS

La adquisición de datos en el sentido general, es el proceso de recolección de información del mundo real. Para muchos ingenieros y científicos éstos datos son sobre todo numéricos y son usualmente recolectados, almacenados y analizados con una computadora. El uso de una computadora automatiza el proceso de adquisición de datos, habilitando la recolección de más datos en menos tiempo y con pocos errores. [3.6]

El sistema de adquisición de datos (SAD) recolecta información del entorno para visualizarla, analizarla y/o procesarla en una computadora. Estos sistemas se desarrollaron debido a que muchas aplicaciones de ingeniería requieren de procesamiento de datos provenientes de los eventos del mundo real. En la *Figura 3.4* se muestra el diagrama de bloques de un sistema de adquisición de datos.



Figura 3.4 Diagrama de bloques de un SAD

El multiplexor es un circuito integrado que selecciona alguna de las señales analógicas cuando se tienen varias señales de entrada. Las líneas de control digital del dispositivo determinan cual entrada es dirigida hacia la salida.

La etapa de acondicionamiento realiza la adecuación de la señal proveniente del transductor que se ha seleccionado con el multiplexor. Casi todas las señales de los transductores deben ser acondicionadas por circuitos analógicos antes de que puedan ser digitalizadas y utilizadas por una computadora. Éste acondicionamiento a menudo incluye amplificación y filtrado de la señal. La amplificación es necesaria para que la amplitud de la señal pueda ajustarse a una porción razonable del rango dinámico del Convertidor Analógico/Digital (CAD). La filtración se debe realizar generalmente en las señales analógicas por varias razones. Algunas veces el ruido o componentes no deseadas de la señal pueden ser eliminados filtrando porciones reales de los espectros de la señal.

Para procesar señales analógicas por medios digitales es necesario convertirlas a formato digital, esto es, transformarlas en una secuencia de números de precisión finita. Este procedimiento se llama conversión analógico-digital y los dispositivos correspondientes conversores analógico-digital (CADs). [3.4]

El CAD convierte las señales analógicas provenientes del bloque de Acondicionamiento en señales digitales para que puedan ser procesadas por la computadora. [3.4]

Conceptualmente, podemos ver la conversión A/D como un proceso en tres pasos:

1. *Muestreo*. Ésta es la conversión de una señal en tiempo continuo a una señal en tiempo discreto obtenida tomando muestras de la señal en tiempo continuo en instantes de tiempo discreto. Así, si $x_a(t)$ es la entrada del muestreador, la salida es $x_a(nT) \equiv x(n)$, donde T se denomina *intervalo de muestreo*. [3.4]
2. *Cuantificación*. Ésta es la conversión de una señal en tiempo discreto con valores continuos a una señal en tiempo discreto con valores discretos (señal digital). El valor de cada muestra de la señal se representa mediante un valor seleccionado de un conjunto finito de valores posibles. La diferencia entre la muestra sin cuantificar $x(n)$ y la salida cuantificada $x_q(n)$ se denomina error de cuantificación. [3.4]
3. *Codificación*. En el proceso de codificación, cada valor discreto $x_q(n)$ se representa mediante secuencia binaria de b -bits. [3.4]

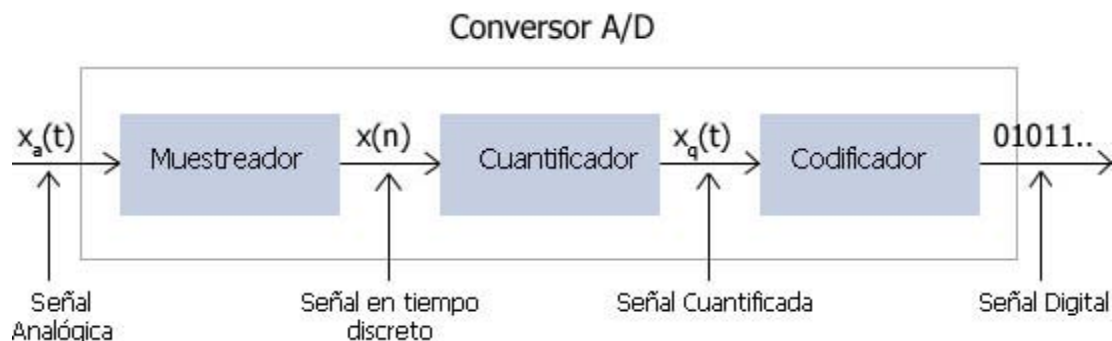


Figura 3.3 Partes básicas de un convertor analógico-digital (A/D). [3.4]

3.3.1 Características de los Convertidores analógico digital

Los parámetros más importantes del convertidor analógico digital son la resolución y la frecuencia de muestreo.

- *Resolución del convertidor analógico digital*. La resolución en los CADs es el más pequeño cambio que puede detectar en una medida. Éste valor es realmente un porcentaje de la lectura de escala completa, pero es comúnmente especificado como el número de bits de salida. Un CAD de n bits tiene 2^n posibles valores de salida y una resolución de una parte en 2^n .
- *Frecuencia de muestreo de CAD*. El muestreo o frecuencia de conversión es la especificación del CAD más examinada. Es el número de lecturas completadas cada segundo. Este parámetro es extremadamente importante cuando las señales que son medidas cambian rápidamente. Es obvio que si la frecuencia de la señal es más alta que la frecuencia de muestreo, las variaciones rápidas de la señal se pierden cuando ocurren entre las muestras consecutivas del CAD.

3.4 SÍNTESIS DE VOZ

3.4.1 Mecanismo de la producción de la voz

El proceso de producción de la voz involucra tres subprocesos: generación de la fuente, articulación y radiación. El órgano vocal humano está constituido por los pulmones, la tráquea, laringe, faringe y cavidades nasales y orales. Juntos forman un tubo conectado como se indica en la *Figura 3.4*. La parte superior que comienza en la laringe es conocido como tracto vocal, que adquiere diferentes formas al mover la mandíbula, la lengua, los labios y otras partes internas. La cavidad nasal está separada de la faringe y la cavidad oral al levantar el velum o el paladar suave. [3.5]

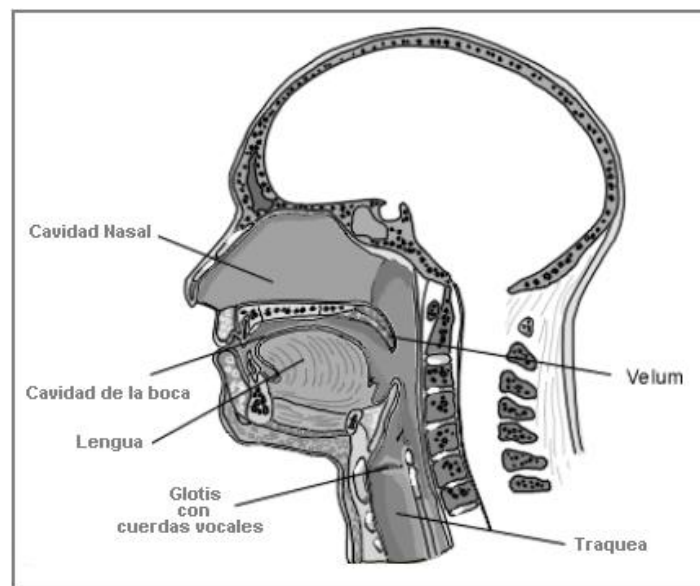


Figura 3.4 Sistema de producción de la voz humana. [3.5]

La voz es producida cuando el aire es forzado desde los pulmones hacia las cuerdas vocales. Cuando los músculos abdominales forzan al diafragma hacia arriba, el aire es empujado hacia adentro y fuera de los pulmones, con un flujo de aire pasando a través de la tráquea y la glotis en la laringe. La glotis, o el espacio de izquierda a derecha de las cuerdas vocales, que se abre usualmente durante la respiración, se hace más estrecha cuando el hablante intenta producir un sonido. El flujo de aire a través de la glotis es interrumpido periódicamente al abrir y cerrar el espacio en concordancia con la interacción entre el flujo de aire y las cuerdas vocales. Este flujo intermitente, llamado el recurso glótico o la fuente de la voz, puede ser simulado por medio de formas de onda triangulares asimétricas. [3.5]

El mecanismo de la vibración de la voz es complicado. En un principio, sin embargo, el efecto de Bernoulli asociado con el flujo de aire y la estabilidad producida por la elasticidad de los músculos mostraron la función de las cuerdas vocales. Cuando las cuerdas vocales son filtradas fuertemente y la presión del aire levantado de los pulmones

es alta, el periodo de apertura y cierre (el periodo de vibración de las cuerdas vocales) se hace corto y el tono de la fuente del sonido es alto. A la inversa, la baja presión de aire produce sonidos con un tono bajo. Este periodo de vibración de las cuerdas vocales es conocido como periodo fundamental, y su recíproco es llamado frecuencia fundamental. El acento y las entonaciones son resultado de variaciones temporales del periodo fundamental. Además, la fuente del sonido, que consiste de la frecuencia fundamental y sus componentes armónicas, es modificada por el tracto vocal para producir calidades de tono en producciones vocales. [3.5]

Es interesante mencionar, que análisis estadísticos de la variación temporal de la frecuencia fundamental durante una conversación de voz de cada hablante indican que la media y la desviación estándar para las voces femeninas son aproximadamente dos veces éstas que para las voces masculinas. [3.5]

3.4.2 Principales características de la voz

La forma de onda de la voz por si sola contiene información lingüística, características vocales del hablante y emociones de éste. Ha sido confirmado que las estructuras lingüísticas y acústicas de la voz están relacionadas de manera compleja a nuestra habilidad intelectual, y están, por otra parte, estrechamente entrelazadas con nuestro desarrollo cultural y social.

3.4.2.1 Información lingüística

La información lingüística indica el significado de lo que desea expresar el hablante y la información emocional. Sin necesidad de decirlo, el primer tipo de información es la más importante.

La producción de la voz humana comienza con el concepto inicial de una idea que el hablante desea llevar al oyente. Subsecuentemente, el hablante convierte la idea en una estructura lingüística al seleccionar las palabras adecuadas o frases que representen ésta, y las ordena de acuerdo a reglas de gramática sencillas o muy rígidas dependiendo de la relación entre hablante y oyente.

Siguiendo este proceso, el cerebro humano produce comandos del nervio motor que mueve varios músculos de los órganos vocales. Este proceso es esencialmente divisible en dos subprocesos: el proceso fisiológico que involucra nervios y músculos, y el proceso físico (acústico) a través del cual la forma de onda de la voz es producida y propagada. Las características de la voz como fenómeno físico son continuas, a pesar de que el lenguaje llevado por el discurso está compuesto esencialmente de unidades discretas de código.

Una sentencia es construida utilizando unidades básicas de palabra, cada palabra está compuesta de sílabas, y cada sílaba está constituida por fonemas, que pueden ser clasificados vocales y consonantes. A pesar de que una sílaba por sí misma no está bien definida, una sílaba está formada generalmente por la concatenación de una vocal y una o

varias consonantes. El número de vocales y consonantes varían dependiendo del método de clasificación y del idioma en cuestión. Puesto que hay una combinación de reglas para construir fonemas en sílabas, el número de las sílabas en cada idioma comprende solo una fracción de todas las posibles combinaciones de los fonemas.

En contraste con el fonema, que es la unidad más pequeña desde el punto de vista lingüístico o fonético, la unidad física de la voz actual es referida como phone. El fonema y el phone son indicados con símbolos de fonema y fonética, respectivamente.

3.4.2.2 Información acústica

Es posible obtener información de tipo acústico en un espectrograma, en donde podemos observar que la forma de onda de la voz y su espectro varían como un proceso no estacionario en periodos de $\frac{1}{2}$ segundo o mayores, sin embargo, en periodos apropiados de 20-40ms se pueden observar características constantes.

Las componentes de la frecuencia dominante que caracterizan a los fonemas corresponden a las componentes de la frecuencia resonante de las vocales, generalmente tienen tres formantes, nombrados primer, segundo y tercer formante, comenzando con la componente de frecuencia más baja. Estos son usualmente escritos como F_1 , F_2 y F_3 . Aun para el mismo fonema, las frecuencias de los formantes varían, dependiendo del hablante. Además, las frecuencias varían, dependiendo de los fonemas adyacentes en pronunciaciones continuas.

A pesar de la dificultad, la medida de los anchos de banda de los formantes ha sido utilizada por muchos investigadores. El rango de los valores extraídos va de los 30 hasta los 120Hz (media de 50Hz) para F_1 , de 30 a 200Hz (media de 60Hz) para F_2 y para F_3 de 40 a 300Hz (media de 115Hz). La variación en el ancho de banda tiene una pequeña influencia en la calidad de la voz a escuchar.

Las características básicas acústicas de los formantes de las vocales pueden ser caracterizadas por F_1 y F_2 . Las cinco vocales son típicamente distribuidas en una forma triangular, que es denominada el triángulo de las vocales. Por otra parte, el orden más alto del formante indica una variación pequeña, dependiendo de las vocales pronunciadas. Por lo tanto, el formante de orden más alto tiene un valor peculiar para cada hablante correspondiente a la longitud de su tracto vocal.

El traslape de características fonéticas de fonema a fonema, es denominado coarticulación. Cada fonema puede ser considerado como el blanco al que los órganos vocales apuntan pero nunca lo alcanzan. Tan pronto como se ha acercado lo suficiente al blanco para ser inteligible al oyente, los órganos cambian sus destinos y comienzan a dirigirse a otro blanco. Esto es hecho para minimizar el esfuerzo expendido al hablar y ayuda a tener una fluencia muy grande.

El fenómeno de la co-articulación se agrega a los problemas de síntesis y reconocimiento de la voz. El caso de la voz en el que no existe la co-articulación suena no natural a nuestros oídos, en síntesis de voz de alta calidad, se debe de incluir un grado

apropiado de co-articulación. En reconocimiento, las co-articulaciones significan que las características de fonemas aislados no se encuentran nunca en silabas conectadas; por lo tanto, cualquier sistema de reconocimiento basado en fonemas identificables debe ser necesariamente correcto para influencias del contexto.

Las consonantes son clasificadas por la periodicidad de la onda, espectro de la frecuencia, duración y variación temporal. Las características acústicas de la gran parte de las consonantes varían como resultado de la co-articulación con vocales, debido a que originalmente las consonantes no tienen estabilidad o periodos de estado constante.

La co-articulación algunas veces afecta fonemas localizados más allá de fonemas adyacentes. Además, desde que varios órganos articulatorios participan en la producción de voz actual, y de que cada órgano tiene su propio tiempo constante de movimiento, el fenómeno acústico resulta de estos movimientos sumamente complicados. Por esa razón, es muy difícil obtener una correspondencia de uno a uno entre los símbolos de fonema y las características acústicas.

3.4.3 Principios de la síntesis de voz

La síntesis de voz es un proceso que artificialmente produce voz para varias aplicaciones, disminuyendo la dependencia de utilizar una persona que produzca voz. Los métodos de síntesis de voz habilitan a una máquina a pasar instrucciones o información al usuario a través de voz. El progreso en la tecnología de computadoras de baja escala de integración y técnicas de predicción lineal (LPC) han ayudado colectivamente al avance de las investigaciones de síntesis. La síntesis de voz esta estrechamente relacionada en investigaciones derivadas de las unidades básicas de información llevadas en la onda de la voz y en el mecanismo de producción de la voz.

El sistema de síntesis se compone de dos módulos claramente diferenciados, que requieren para su realización una metodología y conocimientos de base radicalmente distintos: el proceso lingüístico prosódico y el proceso acústico. Entre estos dos procesos existen puntos de conexión y el punto de encuentro entre ellos será la representación fonética y prosódica del texto. *Ver Figura 3.5.*

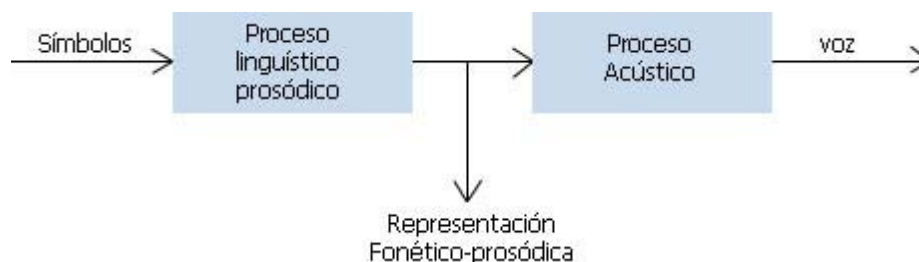


Figura 3.5 Diagrama de bloques de un sistema de síntesis de voz.

Existen métodos de síntesis de voz que pueden dividirse en tres tipos:

1. La síntesis basada en el código de la forma de onda, que se encarga principalmente de almacenar la onda de la voz producida por el hombre, para que posteriormente esta sea utilizada para producir el mensaje deseado.
2. Síntesis con el método análisis/síntesis, en el que las ondas grabadas de la voz humana son transformadas y almacenada en secuencias de parámetros por el método análisis/síntesis, con un sintetizador de voz son dirigidas con parámetros de concatenación para producir mensajes.
3. Y en síntesis por regla, la voz es producida con base en las reglas fonéticas y lingüísticas de las secuencias de las letras o secuencias de los símbolos de fonema y características prosódicas.

Por lo tanto, dentro de la tecnología de síntesis de voz se puede decir que existen dos aproximaciones diferenciadas: la síntesis a partir del concepto y la síntesis a partir del texto. En el primer caso la voz es generada a partir de una representación simbólica del mensaje que queremos emitir y, por lo tanto, requiere un proceso de generación del lenguaje natural. En el segundo caso, no disponemos del significado que queremos representar mediante la señal de la voz, sino que disponemos de un texto que debemos leer. Nos concentraremos en este último tipo de sistemas, que comúnmente los conocemos como conversión de texto a voz.

3.4.3.1 Conversión de texto a voz

El concepto de síntesis TTS (Text to Speech) de alta calidad apareció a mediados de los 80's, como resultado de desarrollos importantes en síntesis de voz y técnicas de procesamiento del lenguaje natural, la mayoría por la aparición de nuevas tecnologías (Procesadores de Señales Digitales y de Lógica de inferencia).

La *Figura 3.6* introduce el diagrama funcional de un sintetizador TTS muy general. Tal como un humano leyendo, este comprende el módulo de Procesamiento del Lenguaje Natural (NPL), capaz de producir la transcripción fonética del texto leído, junto con la entonación y ritmo deseados (a veces recibe el término de prosodia), y el módulo de Procesamiento de la Señal Digital (DSP), que transforma la información que este recibe en voz. Pero los formalismos y los algoritmos aplicados a veces administran, gracias al uso juicioso de conocimiento matemático y lingüístico, a ciertos circuitos pequeños en los pasos del proceso.

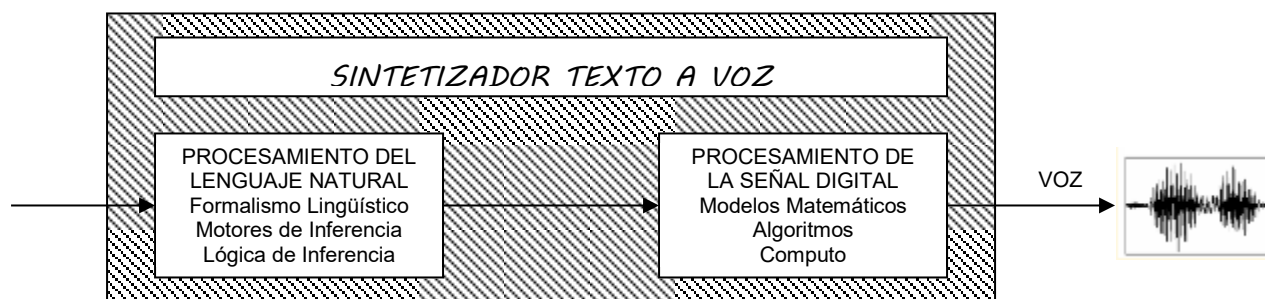


Figura 3.6 Diagrama simple y general de un sistema TTS.

3.4.3.2 El componente NLP

El módulo NLP para propósitos TTS (mostrado en la *Figura 3.7*) está constituido por un analizador morfo-sintáctico necesario en algunos procesos de sistemas de texto a voz de alta calidad, por un módulo nombrado letra-a-sonido y por un bloque generador de prosodia. El analizador morfo-sintáctico permite reducir una sentencia dada en una secuencia de partes de voz. Brevemente narramos en los siguientes párrafos las partes de este módulo.

3.4.3.3 Analizador del texto

Los bloques que conforman la parte del analizador del texto son:

- El módulo de pre-proceso, que organiza la sentencia de entrada en una lista de palabras manipulable.
- El módulo de análisis morfológico, cuya tarea es proponer todas las partes posibles de las categorías de la voz para cada palabra tomada individualmente, con base en su pronunciación.
- El módulo de análisis contextual considera las palabras en su contexto, lo que permite reducir la lista de las partes posibles de las categorías de voz a un número muy restringido de hipótesis altamente probables, dando las posibles partes correspondientes de voz de sus palabras vecinas.
- Finalmente, el parser prosódico-sintáctico, examina el espacio de búsqueda restante y encuentra la estructura del texto que se relaciona más estrechamente con su realización prosódica.

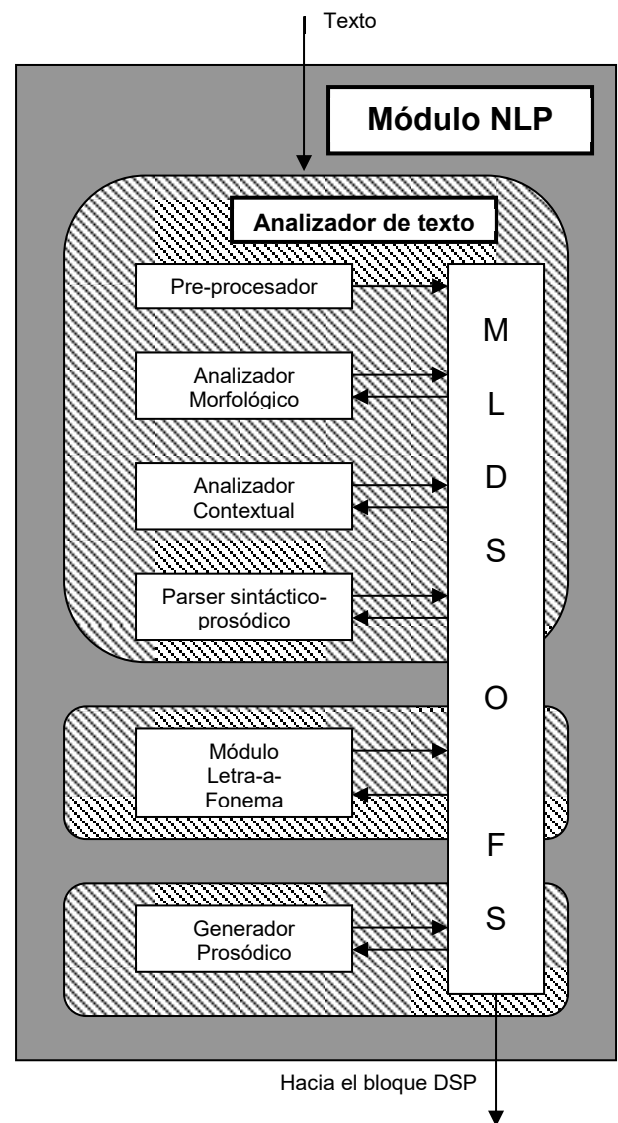


Figura 3.7 El módulo NLP de un sistema general de conversión de texto a voz.

- **El modulo letra-a-fonema (LTS)**

El LTS es responsable de la determinación automática de la transcripción de la fonética del texto de entrada. Esto pudiera parecer, a primera vista de que la tarea de este módulo es como la función de buscar en el diccionario. Sin embargo, a través de un análisis profundo, la mayoría de las palabras aparecen con muchas transcripciones fonéticas en voz genuina, y muchas de éstas no son ni mencionadas en diccionarios de pronunciación.

Es posible organizar la tarea de un módulo LTS en varias formas, a veces clasificadas estrictamente en estrategias basadas en diccionario y basadas en reglas, a pesar de esto existen muchas soluciones intermedias. Las soluciones con base en diccionario consisten en almacenar el máximo conocimiento fonológico en un léxico. En sistemas de transcripción con base en reglas es una estrategia diferente, pues se transfiere la mayoría de la capacidad de los diccionarios en una serie de reglas de letra-a-sonido; las palabras que son pronunciadas en una forma particular y que constituyen una regla por si propias son almacenadas en un diccionario de excepciones.

- **Generador prosódico**

El término de prosodia se refiere a ciertas propiedades de la señal de la voz que están relacionadas a cambios audibles en el tono, intensidad y longitud de la sílaba. Las características prosódicas tienen funciones específicas en la comunicación con voz.

Las características prosódicas crean una segmentación en la cadena de la voz en grupo de sílabas, o, dan un flanco de subida al grupo de sílabas y palabras en pedazos más grandes. Por otra parte, hay características prosódicas que indican la relación entre tales grupos, indicando que dos o más grupos de sílabas están relacionados en varias formas. Este efecto de agrupación es jerárquico, a pesar que no son necesariamente idénticos a la estructura sintáctica de la pronunciación.

Una vez que la estructura prosódica-sintáctica de una sentencia ha sido derivada, es utilizada para obtener la duración precisa de cada fonema (y de silencios), así como la entonación para aplicar en ésta.

- **Componente del DSP**

Intuitivamente, las operaciones involucradas en el módulo DSP son la analogía con el control dinámico de los músculos articulatorios y la frecuencia vibratoria de las cuerdas vocales de modo que la señal de salida se relacione con los requerimientos de entrada. Para que el DSP realice esto correctamente, debe de tomar en cuenta los contrastes articulatorios, al saber por largo tiempo que las transiciones fonéticas son más importantes que los estados constantes para el entendimiento de la voz.

Esto, alternadamente, puede ser alcanzado básicamente de dos maneras:

- Explícitamente, una serie de reglas formalmente describen la influencia de los fonemas entre unos y otros.
- Implícitamente, almacenando ejemplos de transiciones fonéticas y co-articulaciones en una base de datos de segmentos de voz, y utilizando estos justo como son, como unidades acústicas últimas.

Dos clases principales de los sistemas TTS han aparecido de esta alternativa, que rápidamente se han tornado en filosofías de síntesis dando las divergencias de estas en sus significados y objetivos: síntesis por regla y síntesis por concatenación.

Por razones prácticas e históricas los sintetizadores por regla aparecen como sintetizadores formantes. Estos describen a la voz como la evolución dinámica de hasta 60 parámetros, la mayoría relacionados con formantes y frecuencias anti-formantes y anchos de banda junto con la forma de onda. Claramente, el gran número de parámetros complican la etapa de análisis y tiene a producir errores de análisis. Sin embargo, estos sintetizadores poseen un enfoque potencialmente poderoso, por ejemplo, para estudiar las características de la voz dependientes del hablante. Siguiendo la misma idea, los sintetizadores por regla parecen ser una forma natural de dirigir los aspectos articulatorios de los cambios en los estilos del hablante.

Y como oponiéndose a los sintetizadores por regla, el sintetizador por concatenación (*Ver Figura 3.8*) posee un limitado conocimiento de los datos que maneja: la mayor parte de este es embebido en los segmentos a ser encadenados.

En este sintetizador las operaciones que podrían ser utilizadas en el contexto de un sintetizador de música han sido agrupadas en un bloque de procesamiento de sonido, opuesto al bloque de procesamiento de la voz cuyo diseño requiere al menos una cierta comprensión de la fonética.

Finalmente la síntesis de voz resulta de lo siguiente: primero una secuencia de segmentos es deducida de la entrada de fonemas en el sintetizador, en un bloque denominado generación de la lista de segmentos, que da la interfaz entre los módulos NLP y DSP. Una vez que los elementos prosódicos han sido conformados correctamente a segmentos individuales, el módulo de la relación de prosodia hace una petición a la base de datos de los segmentos de síntesis para obtener los parámetros actuales, adecuadamente sin codificar, de los sonidos elementales a ser utilizados, y adapta estos uno por uno a la prosodia requerida. El bloque de concatenación de segmento es entonces el encargado de relacionar dinámicamente los segmentos uno con otro, suavizando discontinuidades. Aquí una vez más, un modelo adecuado de la voz es altamente provechoso, dotándolo de esquemas simples de interpolación, hace que sus parámetros aproximadamente correspondan con las transiciones acústicas suaves entre sonidos. El resultado de los parámetros es finalmente presentado en la entrada del bloque de síntesis, las contrapartes exactas de análisis. Esta tarea finalmente produce la voz.

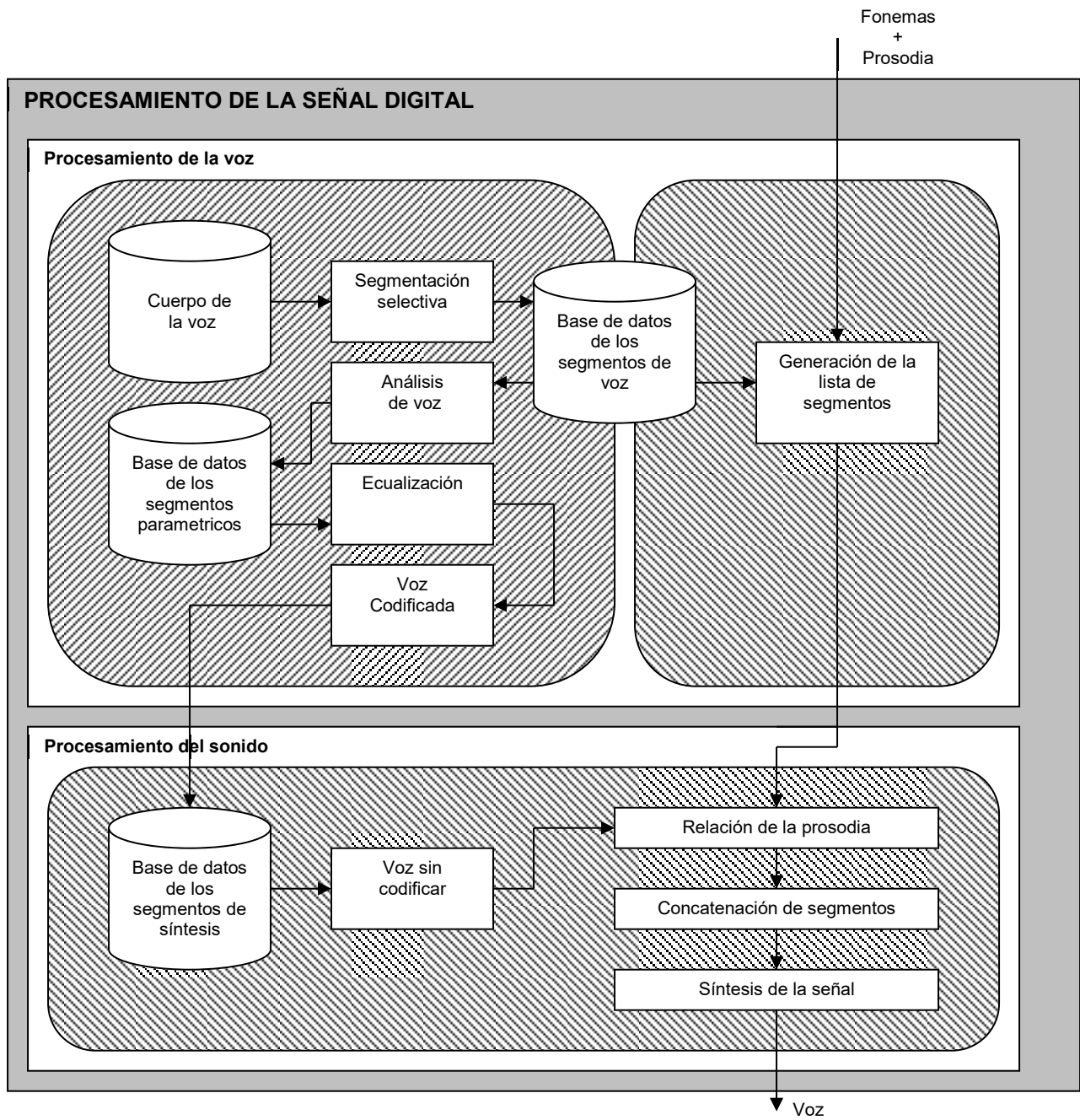


Figura 3.8 Diagrama general de un sintetizador por concatenación.

CAPITULO 4

MICROCONTROLADORES

4.1 INTRODUCCIÓN A LOS MICROCONTROLADORES

A mediados de los años 70's los ejecutivos de la industria del automóvil comenzaron a analizar los beneficios que obtendrían al incluir computadoras en los automóviles. Ellos deseaban utilizar las capacidades de las computadoras para hacer los autos más confortables, seguros y atractivos. Sin embargo, tales computadoras no requerían la velocidad ni las dimensiones de las que ya existían para propósito general. Las computadoras tendrían que ser muy compactas para que se pudieran introducir en los automóviles y el costo era importante para no elevar los precios en la industria automovilística. Tales exigencias llevaron al surgimiento de los primeros microcontroladores, diseñados para necesidades específicas de una aplicación, contenían todos los recursos internos de una computadora en un sólo chip. La capacidad de cada recurso dependía de la aplicación del usuario.

El desarrollo de los microcontroladores comenzó a partir del llamado controlador, que es un dispositivo que se emplea para el gobierno de uno o varios procesos. Aunque, el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían con elementos de lógica discreta; posteriormente se emplearon los microprocesadores, que se rodeaban de circuitos integrados sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir dentro de un chip. [4.2]

4.1.1 ¿Qué es un microcontrolador?

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador (basado en un microprocesador), es decir, está constituido por este circuito integrado y por una serie de recursos internos que se intercomunican; en otras palabras, el microcontrolador es una evolución del microprocesador, al agregarle a éste último los componentes que antes era necesario conectarle externamente con la implementación de otros circuitos. [4.2]

Un microcontrolador está diseñado para realizar la tarea de muchos circuitos lógicos pero utilizando un sólo elemento y es necesario programarlo para que realice su labor. Se dice que un microcontrolador es “la solución en un chip” porque su tamaño reducido minimiza el número de componentes y su costo. Debido a su tamaño es posible montarlo en el propio dispositivo al que gobierna. En resumen este circuito integrado incluye todos los componentes básicos que una computadora contiene en su interior. [4.2]

4.1.2 Diferencia entre microprocesador y microcontrolador

La diferencia entre un microprocesador y un microcontrolador la determinan las características y los elementos que integran cada circuito integrado. Una de las características de los microcontroladores es que generalmente tienen una arquitectura tipo Harvard, es decir, con un bus dual (un bus dedicado para la memoria de programa y otro para la memoria de datos); mientras que en los microprocesadores es común encontrar la arquitectura Von Neumann (un sólo bus para la memoria de datos y programa).

Los elementos básicos que integran a un microprocesador son los siguientes: unidad aritmético-lógica (ALU), unidad de memoria, registros especiales, unidad de control y buses; mientras que los elementos que integran a un microcontrolador son: unidad central de proceso (CPU), memoria ROM, memoria RAM, líneas de entrada y/o salida, buses y algunos módulos para el control de periféricos (temporizadores, puerto serie, puerto paralelo, convertidores analógico-digital y/o convertidores digital-analógico, etc.). Cada uno de estos elementos que integran el microcontrolador, así como la capacidad de cada uno de ellos, dependen del fabricante y de la versión del circuito integrado.

Cada uno de estos circuitos tiene sus propias cualidades y es importante conocerlas para saber dónde aplicarlos con mayor eficiencia. Un microprocesador tiene ventajas importantes sobre un microcontrolador: mayor velocidad, longitud de palabra y rapidez en el procesamiento de datos, aunque actualmente con el avance tecnológico de los microcontroladores estas diferencias se hacen cada vez más pequeñas; pero también tiene desventajas por el número de circuitos integrados que deben acompañarlo en aplicaciones de sistemas de control, es decir, debe interconectarse con otros circuitos para que haga su tarea, esto implica un mayor espacio en el circuito impreso y mayor costo. En cambio, un microcontrolador tiene las siguientes ventajas sobre el microprocesador: un tamaño reducido porque dentro de un mismo circuito integrado encontramos todos los elementos que el microprocesador necesitaría para realizar un trabajo de control y el “conjunto de instrucciones” reducido en el microcontrolador ayuda a una mayor eficiencia en el manejo de memoria. Otra ventaja importante del microcontrolador es la programación, debido a que sólo reprogramando el circuito integrado tendríamos un nuevo sistema de control. Por el contrario, para realizar una función específica con un sistema basado en un microprocesador, el sistema tendría que alambrarse de acuerdo a las necesidades de la aplicación, pero si estas cambian habría que rediseñar nuevamente el circuito

Aunque son diferentes, ambos circuitos integrados son de gran importancia en sus variadas aplicaciones. Mientras que un microprocesador es más eficiente en labores de procesamiento de información, el microcontrolador por su parte, es más eficiente en labores de control de procesos y en aplicaciones de sistemas embebidos.

4.1.3 Aplicaciones del microcontrolador

Las aplicaciones de los microcontroladores son muchísimas y están limitadas por la imaginación del usuario. Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño, costo, mejorar su fiabilidad y disminuir el consumo.

Una aplicación típica emplea varios microcontroladores para controlar pequeñas partes de algún sistema. Estos microcontroladores se comunican entre ellos y con un microprocesador central, probablemente más potente, comparten la información y coordinan sus acciones, este ejemplo lo encontramos en el funcionamiento de cualquier computadora.

En la *Tabla 4.1* se muestran algunos ejemplos donde se emplean generalmente los microcontroladores, así como el área a la que pertenecen.

Área	Ejemplo
Periféricos y dispositivos auxiliares de las computadoras	Mouse, teclado, módem, impresora, escáner, plotter, etc.
Electrodomésticos y línea blanca	Lavadoras, secadoras, horno de microondas, refrigeradores, freidoras, tostadores, sistemas de calefacción, etc.
Aparatos portátiles y de bolsillo	Cámaras fotográficas, tarjetas, monederos electrónicos, teléfonos, palm, etc.
Máquinas expendedoras y juguetería	Dispensarios de gasolina y diesel, robots de juguete, etc.
Instrumentación	Osciloscopio, analizador de estados lógicos, analizador de espectros, multímetros digitales, etc.
Industria automotriz	Sistemas de control en los automóviles: encendido, climatización, frenos ABS, suspensión, seguridad, etc.
Control, Industrial y Robótica	Máquinas de ensamblaje, robots industriales, fresadoras, etc.
Electromedicina	Equipo de administración o monitoreo de funciones del cuerpo: niveles de azúcar en la sangre, presión de la sangre, ritmo cardiaco, control de miembros artificiales, etc.
Sistemas de navegación espacial	Robot explorador de superficie, radares, monitoreo de sistemas dentro de un avión, etc.
Sistemas de seguridad y alarma	Equipos de seguridad para: casa, negocio, bancos, etc.
Termorregulación	Calderas de calefacción, aire acondicionado, hornos, mantas eléctricas, etc.
Aparatos Electrónicos	Televisor, DVD's, videocaseteras, teléfonos, videocámaras, juegos de video, decodificadores satelitales, minicomponentes, etc.

Tabla 4.1 Empleo de los microcontroladores. [4.2]

En la actualidad, el uso de los microcontroladores es cada vez popular por su facilidad de implementación y costo.

4.2 ARQUITECTURA DE UN MICROCONTROLADOR

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de von Neumann, en el momento presente, la necesidad de conseguir elevados rendimientos en el procesamiento de las instrucciones ha desembocado al empleo generalizado de procesadores de arquitectura Harvard.

La arquitectura de von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control). [4.2]. (Ver Figura 4.1)

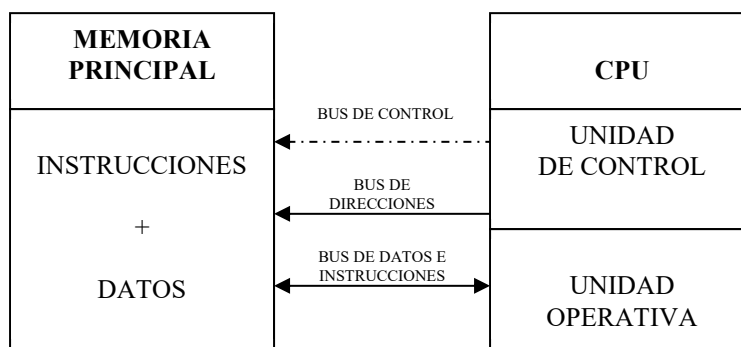


Figura 4.1 En la arquitectura de von Neumann la memoria contiene indistintamente datos e instrucciones. [4.2]

La arquitectura Harvard dispone de dos memorias independientes: una, que contiene sólo instrucciones y otra sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias. [4.2] (Ver Figura 4.2)

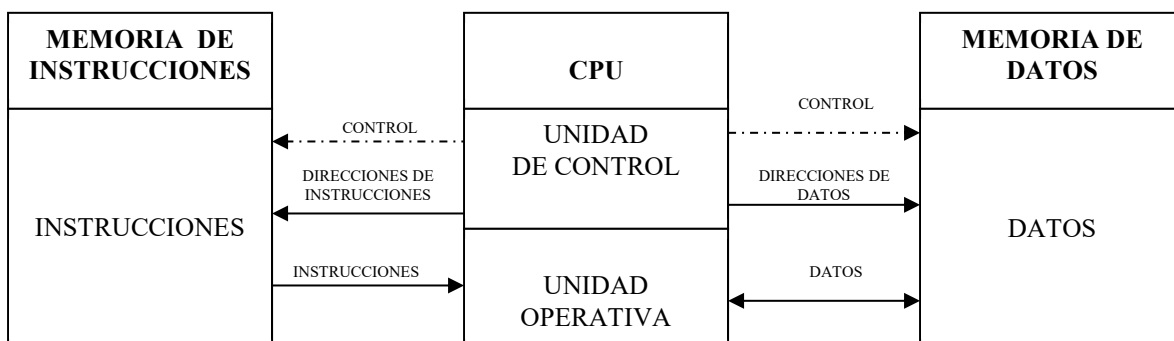


Figura 4.2 Arquitectura Harvard. [4.2]

4.2.1 Recursos comunes a todos los microcontroladores

Al estar todos los microcontroladores integrados en un chip, su estructura fundamental y sus características básicas son muy parecidas. Todos deben disponer de los siguientes módulos de hardware: Procesador (CPU), memoria, líneas de Entrada/Salida y buses. [4.2]

LA UNIDAD CENTRAL DE PROCESO O PROCESADOR (CPU)

Este elemento se encarga de direccionar la memoria de programa, recibir el código OP (Código de Operación) de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado. Realiza operaciones aritméticas y lógicas, controla el acceso a dispositivos de Entrada/Salida y procesa servicios de interrupciones. El CPU contiene una unidad aritmética y lógica (ALU), una unidad de control, registros internos (para almacenamiento temporal), temporizadores, conexiones internas y externas (buses).

Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales:

- *CISC*

Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (Computadoras de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones, máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución. La ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros. [4.2]

- *RISC*

Tanto la industria de los computadores comerciales como las de los microcontroladores están decantándose hacia la filosofía RISC (Computadoras de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo. La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador. [4.2]

- *SISC*

En los microcontroladores destinado a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es “específico”; o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se conoce con el nombre de SISC (Computadoras de Juego de Instrucciones Específico). [4.2]

MEMORIA

La memoria es una parte importante de un sistema de microcomputadora, es el área de almacenamiento para instrucciones y datos utilizados por el CPU. Dependiendo de la aplicación, las memorias se clasifican en dos grupos: las de tipo no volátil, tipo ROM (memoria de solo lectura) que almacena el código del programa y las tipo RAM (memoria de lectura/escritura), volátil, que se destina a almacenar las variables y los cambios de información que se produzcan en el transcurso del programa. Es importante mencionar que sólo existe un programa activo, por lo que no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM. [4.2]

Las memorias ROM una vez que se programan almacenan la información aunque el sistema no sea energizado. Esto permite que el microcontrolador ejecute el programa almacenado en memoria inmediatamente después de ser energizado. [4.2]

Según el tipo de memoria ROM que dispongan los microcontroladores, la aplicación y utilización de los mismos es diferente. A continuación, se describen cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado. [4.2]

1ª. ROM con máscara

Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado costo del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades. [4.2]

2ª. OTP

OTP (One Time Programmable), es una memoria no volátil de sólo lectura “programable una sola vez” por el usuario. Se graba por el usuario mediante un sencillo grabador controlado por un programa desde una PC. [4.2]

3ª. EPROM

Los microcontroladores que disponen de memoria EPROM (Erasable Programmable Read Only Memory) pueden borrarse y grabarse muchas veces. Se graba de la misma forma que una OTP, con un dispositivo que es gobernado mediante un computador personal, que recibe el nombre de grabador. Si, posteriormente, se desea borrar el contenido, disponen de una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico y son más caros que los microcontroladores con memoria OTP que están hechos con material plástico. [4.2]

4ª. EEPROM (Electrical Erasable Programmable Read Only Memory)

Se trata de memorias de sólo lectura, programables y borrables eléctricamente. Es muy cómoda y rápida la operación de grabado y la de borrado. No disponen de ventana de cristal en la superficie. Los microcontroladores dotados de esta memoria, una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera sin ser retirados de dicho circuito. Para ello se usan “grabadores en circuito” que confieren gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo. [4.2]

El número de veces que puede grabarse y borrarse una memoria EEPROM es finito, por lo que no es recomendable una reprogramación continua. [4.2]

5ª. FLASH

Funciona como ROM y RAM pero tiene menor consumo de energía y mayor capacidad de almacenamiento. A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM. [4.2]

La memoria de datos es una memoria que puede ser escrita y leída según sea requerido por el programa. Tiene las funciones de almacenamiento de datos (pila) y como almacenamiento de variables. Este tipo de memoria es usualmente llamada memoria RAM (Memoria de Acceso Aleatorio). Cada localidad de memoria tiene una dirección única con la cual el CPU encuentra la información necesaria. [4.2]

Los microcontroladores actuales contienen ambas memorias (Datos y Programa) incluidas dentro del circuito integrado. Por otro lado, resulta necesario contar con otras unidades que hacen posible el funcionamiento mínimo de un microcontrolador que son las líneas de Entrada/Salida, los buses y el circuito del Reloj. [4.2]

LINEAS DE ENTRADA Y/O SALIDA

Los microcontroladores requieren de una interfase para comunicarse con la circuitería externa, denominada comúnmente como puerto. Existen puertos de entrada y/o salida los cuales permiten que las señales (o datos) sean leídos del exterior o mandados al exterior del Microcontrolador.

A excepción de dos pines para la alimentación, dos para el cristal de cuarzo y uno más para provocar el Reset, los pines restantes del microcontrolador están designados para los puertos que sirven para soportar su comunicación con los periféricos externos que controla, éstas se agrupan en conjuntos. Hay modelos con líneas que soportan la comunicación en serie; otros disponen de conjuntos de líneas que implementan puertas de comunicación para diversos protocolos, como el I²C, el USB, etc. Dependiendo de la aplicación, los pines son conectados a periféricos exteriores como teclados, interruptores, sensores, relevadores, motores, etc.

BUSES

Los buses son conexiones físicas entre el CPU, la memoria y las líneas de Entrada/Salida. Hay tres tipos de buses: el bus de direcciones, el bus de datos y el bus de control.

El bus de direcciones es utilizado para localizar direcciones específicas en donde los datos e instrucciones residen en memoria con el propósito de acceder a estos. El CPU pone una dirección en este bus cuando necesita traer información particular de la memoria o cuando requiere almacenar información en una localidad específica de memoria. El bus de datos es utilizado para llevar instrucciones y datos actuales hacia o desde localidades de memoria, CPU, y dispositivos de E/S. El último bus, el bus de control, es necesario para enviar y recibir comandos de control tales como señales de lectura o escritura entre el CPU, una memoria y dispositivos de E/S.

RELOJ PRINCIPAL

El Reloj provee una referencia en el tiempo para la ejecución del programa. Esta señal determina en qué momento los datos deben ser escritos o leídos de la memoria. Así mismo, provee la sincronía con los dispositivos conectados al microcontrolador (Periféricos). Este dispositivo es un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema. [4.2]

Generalmente, el circuito de reloj esta incorporado en el microcontrolador y solo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien o resonador cerámico o una red R-C. [4.2]

Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva a un incremento del consumo de energía. [4.2]

4.2.2 Recursos Especiales

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizara el costo, el hardware y el software. [4.2]

Los principales recursos específicos que incorporan los microcontroladores son:

TEMPORIZADORES O “TIMERS”

Se emplean para controlar periodos de tiempo y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores). [4.2]

Para la medida de los tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando a ritmo de los impulsos del reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso. [4.2]

Cuando se desea contar acontecimientos que se materializan por cambios de nivel o flancos en algunos de las pines del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos. [4.2]

PERRO GUARDIÁN O “WATCHDOG”

El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema. Está destinado a provocar una reinicialización cuando un programa queda bloqueado. [4.2]

Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al Perro guardián y, al completar su temporización, ladrará y ladrará hasta provocar el reset. [4.2]

PROTECCIÓN ANTE EL FALLO DE ALIMENTACIÓN O “BROWNOUT”

Se trata de un circuito que provoca un reset al microcontrolador cuando el voltaje de alimentación (V_{DD}) es inferior a un voltaje mínimo (“brownout”). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor. [4.2]

ESTADO DE REPOSO O DE BAJO CONSUMO

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento.

Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que los

pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se “congelan” sus circuitos asociados, quedando sumido el microcontrolador en un estado de reposo. Al avisarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo. [4.2]

CONVERTIDOR ANALÓGICO/DIGITAL

El convertidor A/D procesa señales analógicas para obtener señales digitales. Suele disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las pines del circuito integrado. [4.2]

CONVERTIDOR DIGITAL/ANALÓGICO

Transforma los datos digitales obtenidos del procesamiento del procesador en su correspondiente señal analógica que saca al exterior por uno de los pines de la cápsula. [4.2]

COMPARADOR ANALÓGICO

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por uno de los pines de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra. [4.2]

También hay modelos con un módulo de tensión de referencia, que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores. [4.2]

MODULADOR DE ANCHURA DE IMPULSOS O PWM

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de los pines del encapsulado. [4.2]

PUERTOS DE ENTRADA Y SALIDA DIGITALES.

Todos los microcontroladores destinan algunos de sus pines a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando puertos. [4.2]

Las líneas digitales de los puertos pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración. [4.2]

PUERTOS DE COMUNICACIÓN.

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos, algunos modelos disponen de recursos que permiten este tipo de comunicaciones, entre los que destacan: UART (adaptador de comunicación serie asíncrona), USART (adaptador de comunicación serie síncrona y asíncrona), Puerta paralela esclava para conectarse a los buses de otros microprocesadores, USB (Universal Serie Bus, moderno bus serie para las PC), BUS I²C (Interfaz Serie de dos hilos desarrollado por Philips) y el adaptador CAN (Controller Area Network) para permitir la adaptación con redes de conmutación multiplexada desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. [4.2]

4.3 PRINCIPALES FABRICANTES

Se considera Intel como el “padre” de los microcontroladores. Este fabricante ha ido por delante de los demás presentando los nuevos productos. El primer microcontrolador de 8 bits se considera al 8048 y lo fabricó Intel en la década de los 70's. La siguiente generación, más evolucionada, fue la 8051 a la que ha seguido la de 16 bits, 80C196, que alcanza frecuencias de funcionamiento de 50 Mhz. [4.2]

Con buen olfato comercial Intel ha diseñado versiones de microcontroladores basados en los microprocesadores 8088 y 8086. Se trata de los modelos 80188 y 80186. La misma política ha seguido con el 80386 creando el 80386EX. Estos productos permiten trabajar con las herramientas clásicas desarrolladas por el entorno del PC. [4.2]

Otra de las principales empresas del mundo de los dispositivos micro programables es Freescale Semiconductor, que dispone del potente microcontrolador 68HC11 de 8 bits para datos y 16 para direcciones. [4.2]

Los microcontroladores PIC de la empresa Microchip han sido conocidos durante los últimos 30 años. Su popularidad avanza día a día siendo incluidos en la mayoría de los proyectos debido a su bajo costo, consumo reducido, tamaño pequeño, programación fácil y a su gran cantidad de herramientas de soporte económicas. [4.2]

También Zilog ha irrumpido con fuerza en el mercado de los microcontroladores. Ha lanzado algunos modelos con memoria OTP de la familia Z86XX, muy completos y fáciles de manejar. [4.2]

Otras empresas como SGS-Thomson, Hitachi, Texas, Toshiba, Nacional Semiconductor, etc., abarcan pequeñas partes del mercado. [4.2]

Todos los microcontroladores que se fabrican en el presente son buenos y el mejor no siempre es el mismo. Cambian el modelo y el fabricante según la aplicación y las circunstancias que lo envuelven. Un buen diseñador profesional debe conocer la gama de las ofertas para elegir en cada situación el microcontrolador que le conviene. [4.2]

En la *Tabla 4.2* se muestran los fabricantes más destacados con sus modelos más sobresalientes:

Fabricante	Modelos de Microcontroladores
INTEL	8048, 8051, 80C196, 80C186, 80188, 80386EX
Freescale Semiconductor (<i>antes MOTOROLA</i>)	6805, 68HC11, 68HC12, 68HC16, 683XX
HITACHI	HD64180
PHILLIPS	GAMA COMPLETA DE CLONICOS 8051
SGS-THOMSON	ST-62XX
MICROCHIP	PIC's
NATIONAL SEMICONDUCTOR	COP400, COP800
ZILOG	Z8, Z86XX Z8051 series
TEXAS INSTRUMENTS	max.120 MHz Piccolo™ C28x™ series
TOSHIBA	TX00, TX03 and TX09

Tabla 4.2 Fabricantes de Microcontroladores más destacados. [4.2]

En los apartados siguientes recopilamos información de tres fabricantes que ocupan un porcentaje muy importante en el mercado de los microcontroladores, junto con la familia que mas ha sobresalido de cada uno de ellos.

4.3.1 Intel

Fundada en 1968 para construir memorias con semiconductores, Intel introdujo el primer microcontrolador en 1971. Este microcontrolador era de 8 bits y pertenecía a la familia 8048. Posteriormente, a comienzos de los años 80, Intel propuso, como sucesor lógico del microcontrolador 8048, un nuevo modelo bautizado con el nombre 8051. La familia 8051 de microcontroladores son controladores de 8 bits capaces de direccionar hasta 64 Kbytes de memoria de programa y una memoria separada de datos de 64 Kbytes. [4.4]

Esta primera versión, realizada en tecnología HMOS, relativamente consumidora de energía, tenía ya tres variantes: el 8051 propiamente dicho, dotado de una ROM interna programada en fábrica, el 8751, equipado con una EPROM, y el 8031, desprovisto de cualquier tipo de memoria permanente tenía 128 bytes de RAM interna (el 8032 tenía RAM interna de 256 bytes y un temporizador adicional). Además poseía dos temporizadores/contadores, un puerto serie, cuatro puertos de entrada/salida paralelas de propósito general (P0, P1, P2 y P3) y una lógica de control de interrupción con cinco fuentes de interrupciones. Al lado de la RAM interna, el 8031 contaba con varios registros de funciones especiales (SFR) que son para el control y registros de datos. Los SFR también incluyen el registro acumulador, el registro B, y el registro o palabra de estado de programa (PSW), que contiene los flags de la CPU.

Esta primera serie se completó rápidamente con la versión 8052. A este último microcontrolador, muy cercano al 8051, le añadieron un periférico (un tercer temporizador) y 128 bytes de RAM interna. También existe una versión del 8052 que contiene un intérprete de lenguaje BASIC en su memoria ROM. [4.4]

Nació así la familia MCS-51, que se completó rápidamente con otros miembros. La multiplicidad de modelos se justifica por un concepto al que ya se había prestado atención antes de la aparición del 8051: la integración de un entorno mínimo alrededor de una unidad central de proceso limita la universalidad del componente; hay que procurar reutilizar la misma unidad central modificando solamente el entorno que se integra. [4.4]

A esta ventaja se debe añadir el hecho de que todos estos microcontroladores han sido concebidos alrededor de un mismo núcleo hardware y software de modo que la versión realizada en el estudio y empleo de uno de los microcontroladores se puede aprovechar después con los miembros de la familia. La fuerza de una familia de microcontroladores reside pues en su capacidad de diversificación. [4.4]

Esta perspectiva incitó a un buen número de fabricantes (SIEMENS, MHS, OKI, SIGNETICS, PHILIPS, etc.) a firmar acuerdos de producción como segundas fuentes de estos microcontroladores. La adquisición de una licencia de producción del núcleo básico del 8051 también autoriza a estas compañías a diseñar nuevas versiones originales. Diseñar un nuevo microcontrolador perteneciente a la familia 8051 consiste en completar la máscara de fabricación del 8051 con nuevas funciones periféricas. [4.4]

La unidad de la familia está garantizada por el hecho de que todas las nuevas versiones se construyen alrededor de un mismo núcleo de software y hardware. Solamente existe diversidad en la elección de funciones periféricas integradas alrededor de este núcleo. [4.4]

Actualmente, Intel suministra a las industrias en cómputo y comunicaciones chips, tarjetas, sistemas, y módulos de software que son partes de una computadora, servidores y productos de redes y comunicaciones. Estos productos son utilizados por miembros de la industria para crear sistemas más avanzados.

FAMILIA 8051

El 8051 es una familia muy popular, de la industria estándar de 8 bits, manufacturada por varias compañías con características diferentes. El dispositivo básico, considerado el primer miembro de la familia, es el microcontrolador 8051 con 40 pines. El 80C51 es la versión de baja potencia CMOS. El 8751 contiene una memoria de programa EPROM, ayudando a que éste pueda ser programado por el usuario. El 89C51 contiene una memoria FLASH programable y borrable sin luz ultravioleta. El 8052 es un miembro de la familia mejorado que contiene más memoria RAM y más contadores y temporizadores. Existen muchas versiones de la familia de 40 pines que incluyen convertidores analógico-digital, y también moduladores del ancho de pulso. En esta familia también encontramos microcontroladores de 20 pines, cuyo código es compatible con los

dispositivos de 40 pines. Los chips de 20 pines, han sido manufacturados para aplicaciones menos complejas, en donde los requerimientos de líneas de entrada y/o salida no son muy altos y el nivel de potencia que se requiere es bajo, por ejemplo, en aplicaciones portátiles.

El AT89C1051 y el AT89C2051 (manufacturado por ATMEL) son microcontroladores con un código de programa totalmente compatible con la familia 8051 y trabajan con baja potencia.

El dispositivo 8051 básico tiene las siguientes características:

- 4 Kbytes de memoria de programa
- Memoria RAM (256 x 8)
- Empacado estándar de 40 pines
- 2 temporizadores y contadores de 16 bits
- Seis fuentes de interrupción
- Puertos serie programable (UART)
- Interfaz de memoria externa
- 32 líneas de entrada y/o salida programables

Las versiones de la memoria EPROM en la familia son utilizadas para diversas aplicaciones, y la memoria de programa es borrada con rayos de luz ultravioleta. La configuración estándar de los pines se muestra en la *Figura 4.3*.

30	ALE	P0.0/AD0	39
31	E \bar{A}	P0.1/AD1	38
29	PSEN	P0.2/AD2	37
9	RST	P0.3/AD3	36
		P0.4/AD4	35
18	XTAL2	P0.5/AD5	34
19	XTAL1	P0.6/AD6	33
		P0.7/AD7	32
1	P1.0	P2.0/AD8	21
2	P1.1	P2.1/AD9	22
3	P1.2	P2.2/AD10	23
4	P1.3	P2.3/AD11	24
5	P1.4	P2.4/AD12	25
6	P1.5	P2.5/AD13	26
7	P1.6	P2.6/AD14	27
8	P1.7	P2.7/AD15	28
		P3.0/RXD	10
		P3.1/TXD	11
		P3.2/INT0	12
		P3.3/INT1	13
		P3.4/T0	14
		P3.5/T1	15
		P3.6/WR	16
		P3.7/RD	17

Figura 4.3 Configuración de pines del microcontrolador 8051 estándar

TEMPORIZADORES Y CONTADORES.

EL 8051 contiene dos temporizadores/contadores conocidos como temporizador/contador 0 y temporizador/contador 1 (los miembros más grandes de la familia 8051 contiene más temporizadores/contadores). Estas características pueden ser operadas en varios modos dependiendo de la configuración de dos registros TCON y TMOD. Estos registros deben de ser programados antes de utilizar el temporizador o el contador.

CONTROL DE INTERRUPCIONES.

El 8051 estándar está provisto de seis fuentes de interrupción:

- Dos interrupciones externas (INT0 e INT1)
- Dos interrupciones del temporizador (temporizador 0 y 1)
- Una interrupción de recepción del puerto serie
- Una interrupción de transmisión del puerto serie

Cada interrupción es asignada en localidades fijas de memoria y una interrupción causa que el CPU brinque a esta localidad, en donde se ejecuta la rutina de servicio a la interrupción.

Cada fuente de interrupción puede ser habilitada o deshabilitada individualmente colocando un 1 o un 0 en el bit correspondiente.

CONFIGURACIÓN MÍNIMA DEL MICROCONTROLADOR.

En la configuración mínima (*Figura 4.4*) de sistemas basados en el microcontrolador 8051, solamente se requiere de los siguientes componentes para que funcionen:

X1	Cristal de cuarzo.
C1, C2	Capacitores de 33pF.

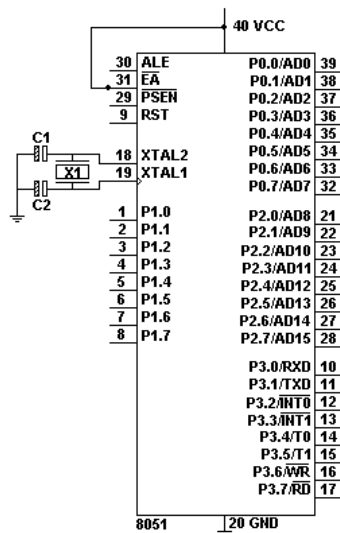


Figura 4.4 Configuración mínima del 8051.

4.3.2 Freescale Semiconductor (antes Motorola)

Así como Intel, Freescale es el mayor fabricante de microprocesadores para computación, lidera desde hace años el mercado mundial de microcontroladores para aplicaciones específicas con sus familias 68HC05, 68HC08 y 68HC11 de 8 bits, 68HC12 y 68HC16 de 16, y 68300 y MCore de 32.

Su primera familia, la 6805 (de gama baja y media) ha sido la más vendida en la historia de los microcontroladores; la 68HC08 se ha introducido para usuarios de la 05 que necesitan más prestaciones. Más adelante, a mediados de 1980, Motorola desarrolló y comercializó los microcontroladores de la familia 68HC11, una de las más populares en la gama alta, que por su relación prestaciones/precio y gran facilidad de uso resulta ideal para el aficionado a la electrónica. Sin embargo, a pesar de que el 68HC11 es utilizado continuamente, un incremento en las aplicaciones embebidas de cómputo ha estado buscando microcontroladores más veloces y con más capacidades. Freescale Semiconductor ha respondido con las familias de 16 bits de velocidad variable, de mayor capacidad de memoria, funciones matemáticas y líneas de E/S. Uno de estos es la familia 68HC12, introducida por primera vez en 1997, provista de más velocidad, lo cual es posible expandiendo el ancho del bus de datos de 8 a 16 bits e implementando un mecanismo del manejo de instrucciones en una cola. El 68HC12 expandió su capacidad matemática agregando operaciones matemáticas complejas y se hace distinguir entre otros microcontroladores por sus funciones de controlador integrado **fuzzy**. Por su parte, el 68HC16 ofrece operaciones DSP (Digital Signal Processing).

Todos los miembros de las familias 68XX parten de la arquitectura del clásico microprocesador M6800: en algunos casos el núcleo del microcontrolador (o CPU) es un 6800 reforzado (68HC11) y en otros simplificado (6805). Sus juegos de instrucciones se basan en el del 6800, y los modos de direccionamiento son prácticamente los mismos, con lo que la programación de todas las familias resulta muy similar (y sencilla).

Los microcontroladores de Freescale Semiconductor han sido utilizados en varias aplicaciones a lo largo de los años. Los más grandes consumidores de estos controladores son los de la industria automovilística.

FAMILIA 68HC11

La familia 68HC11 está compuesta por microcontroladores (μC) de 8 bits de gama media alta (equiparable al conocido 8051 en prestaciones y precio). El 68HC11 se introdujo en el año 1985 como actualización de un μC más antiguo, el 6801, siendo su orientación inicial las aplicaciones de la industria automovilística. No obstante, el HC11 se viene empleando en todo tipo de aplicaciones de consumo, telecomunicaciones e industriales (cámaras fotográficas y vídeo, teléfonos, instrumentación biomédica, etc.), siendo uno de los microcontroladores más populares en todo el mundo.

Se comercializan ejemplares del 68HC11 fabricados en base a la tecnología HCMOS que operan a 3 y 5V, y a frecuencias de reloj de 2 a 5 MHz. Las frecuencias relativamente pequeñas a las que trabaja no deben llevar a engaño, pues aunque algunos microcontroladores de otros fabricantes operan a frecuencias más altas, los de Freescale Semiconductor realizan un ciclo de bus (acceso a memoria) en un único ciclo de reloj, mientras que otros (como los Intel) precisan tres, cuatro o más, de modo que los microcontroladores de Freescale Semiconductor operando a frecuencias inferiores pueden obtener un rendimiento superior. Por otro lado, debido a que el diseño del HC11 es estático no necesita refrescar sus registros internos, por lo tanto, no presenta un límite inferior en su frecuencia de reloj (de modo que en las aplicaciones de bajo consumo puede reducirse todo lo que sea preciso).

Cuenta internamente con 8KB de ROM, 512 bytes de EEPROM y 256 bytes de RAM. Su bus de direcciones es de 16 líneas, por lo tanto puede direccionar hasta 64 KB (aunque algunos de sus miembros pueden direccionar más de un megabyte), y aunque el ancho del bus de datos es de 8 bits, trabaja fácilmente con datos de 16 bits. Todas las versiones del 68HC11 pueden presentar sus buses al exterior para trabajar con memoria externa. El 68HC11 posee dos registros acumuladores de 8 bits (*Figura. 4.5*), el A y el B; juntos constituyen además un acumulador D de 16 bits (A y B pueden utilizarse libremente como dos registros independientes o como uno único D). También dispone de dos registros índice de 16 bits, el X y el Y, un registro contador de programa PC y un puntero de pila, SP, también de 16 bits. El registro de estado, denominado CCR (*condition code register*), incluye 8 señalizadores o *flags*: C (acarreo), V (desbordamiento), Z (resultado cero), N (resultado negativo), H (acarreo a la mitad), I (máscara de las interrupciones IRQ), X (máscara de las interrupciones XRQ) y S (máscara de la instrucción STOP empleada para el estado de bajo consumo).

El 68HC11 cuenta con siete modos de direccionamiento (los mismos que el 6800): directo, extendido, inmediato, indexado, inherente, acumulador y relativo. Su juego de instrucciones posee más de 300 códigos de operación (incluyendo los del 6800 como subconjunto); algunos códigos de operación se componen de un byte y otros de dos (pre-byte y opcode), de modo que una instrucción en total ocupa entre 1 y 4 bytes.

Además, la familia 68HC11 contiene circuitería de automonitoreo para protección contra errores del sistema, watchdog, un circuito detector de código ilegal y posee dos modos de ahorro de energía (STOP, WAIT) controlados por programa.

FAMILIAS DEL 68HC11

Familia D

Ofrece una alternativa económica cuando se requieren aplicaciones de 8 bits, sin muchos periféricos y menos memoria.

Familia E

Contiene una capacidad extremadamente flexible, además combina EPROM y EEPROM en un chip.

Familia F

Resalta por contener puertos I/O extras.

Familia K

Es formada por dispositivos de alto rendimiento, gran velocidad, PWM's y plenitud de I/O.

Familia S

Ofrece modos de ahorro de energía, PLL y memoria mayor.

ARQUITECTURA INTERNA

La siguiente figura (*Figura 4.5*) muestra los bloques internos de 68HC11 para tener una vista al interior del chip.

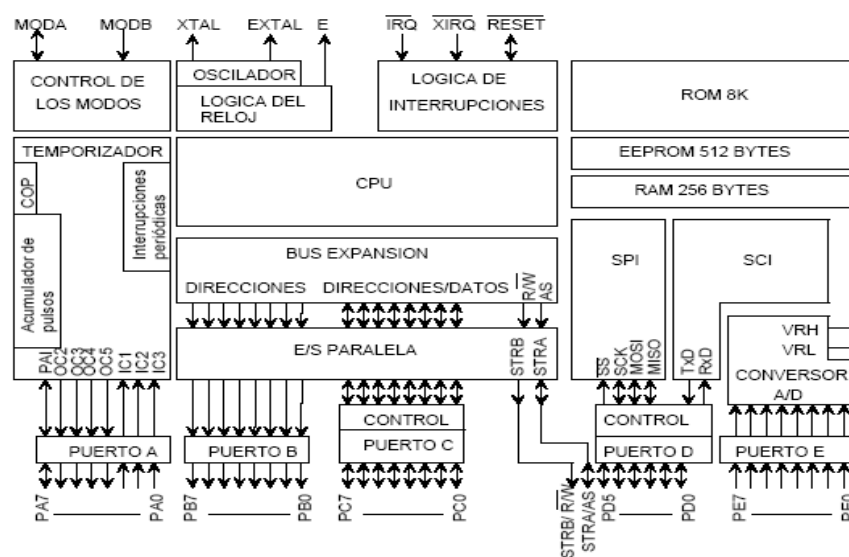


Figura 4.5 Arquitectura interna del 68HC11. [4.5]

CONFIGURACIÓN DE PINES

Para facilitar la comprensión, se clasifican todos los pines del microcontrolador en grupos de acuerdo a las funciones de los mismos, siendo estas agrupaciones las siguientes:

- Alimentación: VDD, VSS.
- Reloj: EXTAL, XTAL, E.
- Reset: RESET.
- Transmisión serie asíncrona: TxD, RxD.
- Petición de interrupciones hardware: IRQ, XIRQ, IC1–3, PAI, STRA.
- Modos de arranque: MODA, MODB.
- Comparadores: OC1–5.
- Capturadores: IC1–3, PAI.
- Transmisión serie síncrona: SCK, MISO, MOSI, SS.
- Puertos: PA0–7, PB0–7, PC0–7, PD0–3, PE0–3.
- Convertidores Analógico/Digital: AN0–7.
- Buses: AD0–7, A8–15, AS, R/W.

En la *Figura 4.6* podemos ver la configuración estándar de los pines del microcontrolador 68HC11.

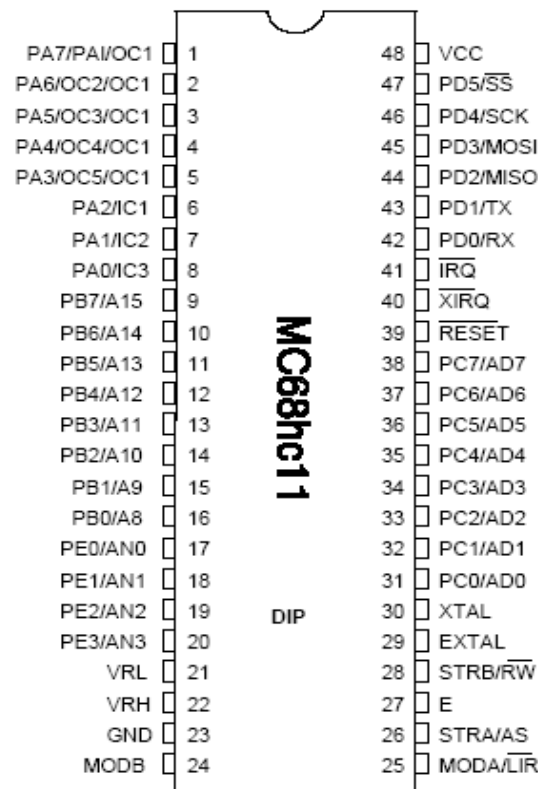


Figura 4.6 Diagrama de pines del 68HC11.

INTERRUPCIONES

EL microcontrolador 68HC11 trabaja con tres tipos de interrupciones:

Las interrupciones enmascarables

Como su nombre lo indica, se pueden enmascarar, es decir, inhibir sin más que actuar sobre el bit I del CCR.

Las interrupciones no enmascarables

Estas interrupciones no se pueden inhibir. Son las interrupciones correspondientes a un fallo en el MCU, instrucción ilegal, RESET y la interrupción externa XIRQ.

Las interrupciones software

Son las producidas por el propio programador en unos instantes totalmente conocidos. Sólo existe una interrupción software que se produce con la instrucción SWI (equivalente al TRAP del 68000). Estas interrupciones son no enmascarables porque de lo contrario la CPU se colgaría.

También es posible clasificar las interrupciones en internas y externas: *Las interrupciones internas* son las producidas por circuitos o periféricos integrados dentro del propio microcontrolador. Y *las interrupciones externas* son las producidas por circuitos o periféricos externos al MCU. Existen 2 entradas de interrupciones externas.

TEMPORIZADOR PRINCIPAL

El temporizador está formado por un registro de 16 bits que se está incrementando constantemente. Arranca con el valor 0, se va incrementando con cada pulso de reloj y cuando llega a \$FFFF se produce una interrupción de overflow, pasa a valer nuevamente 0 y continúa contando.

SISTEMAS BASADOS EN EL 68HC11: MODOS SIMPLE Y EXPANDIDO

El 68HC11 posee dos modos normales de operación, simple y expandido. En modo simple el HC11 no expone sus buses al exterior, de modo que conserva todos sus puertos y solamente opera con la memoria incluida dentro del propio chip. Por otra parte, el 68HC11 se configura en modo expandido cuando la memoria interna o los periféricos integrados resultan insuficientes para la aplicación en desarrollo, de modo que se precisa el acceso desde el exterior a los buses internos. Ver *Figura 4.7*.

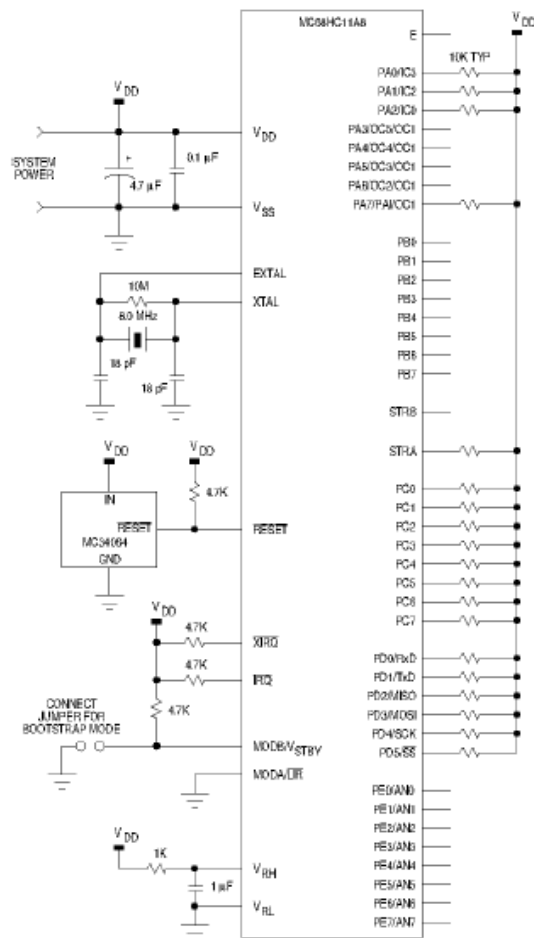


Figura 4.7. Configuración modo simple del microcontrolador 68HC11.

4.3.3 Microchip

En 1965, la empresa GI creó una división de microelectrónica GI Microelectronics Division que comenzó su andadura fabricando memorias EPROM y EEPROM, que conformaban las familias AY3-XXXX y AY5-XXXX. A principios de los años 70 diseñó el microprocesador de 16 bits CP1600, razonablemente bueno pero que no manejaba eficazmente las Entradas y Salidas. Para solventar este problema, en 1975 diseñó un chip destinado a controlar E/S: el PIC (Peripheral Interface Controller). Se trataba de un controlador rápido pero limitado y con pocas instrucciones pues iba a trabajar en combinación con el CP1600.

La arquitectura del PIC, que se comercializó en 1975, era sustancialmente la misma que la de los modelos PIC16C5X. En aquel momento se fabricaba con la tecnología NMOS y el producto sólo se ofrecía con memoria ROM y con un pequeño pero robusto micro código.

La década de los 80 no fue buena para GI, que tuvo que reestructurar sus negocios concentrando sus actividades en los semiconductores de potencia. La GI Microelectronics

Division se convirtió en una empresa subsidiaria, llamada GI Electronics Inc. Finalmente, en 1985, la empresa fue vendida a un grupo de inversores de capital de riesgo, los cuales, tras analizar la situación, rebautizaron a la empresa con el nombre de Arizona Microchip Technology y orientaron su negocio a los PIC, las memorias EPROM paralelo y las EEPROM serie. Se comenzó rediseñando los PIC, que pasaron a fabricarse con tecnología CMOS, surgiendo la familia de gama baja IC16C5X, considerada como la clásica.

Microchip cuenta con su fábrica principal en Chandler, Arizona, en donde se fabricaban los chips con los más avanzados recursos técnicos. En 1993 construyó otra fábrica de similares características en Temple, Arizona. También cuenta con centros de ensamblaje y ensayos en Taiwan y Tailandia.

Una de las razones del éxito de los PIC se basa en su utilización. Cuando se aprende a manejar uno de ellos, conociendo su arquitectura y su repertorio de instrucciones, es muy fácil emplear otro modelo.

FAMILIA PIC16F87x

Bajo el nombre de esta subfamilia de microcontroladores, actualmente encontramos cuatro modelos: EL PIC 16F873/4/6 y 7. Estos microcontroladores disponen de una memoria y programa FLASH de 4 a 8 KBytes de 14 bits, considerablemente superior frente al PIC 16F84 en el que solo disponíamos de 1 KByte de 14 bits.

En este apartado hablaremos del PIC16F877, a continuación se enlistan las características que comparte el con los dispositivos más cercanos de su familia.

Las características comparativas más relevantes son mostradas en la *Tabla 4.3*.

- CPU RISC
- 35 instrucciones de código de programa
- Todas las instrucciones se ejecutan en un ciclo de reloj, excepto los saltos
- Frecuencia de operación de 0 a 20 MHz (DC a 200 nseg de ciclo de instrucción)
- Hasta 8k x 14 bits de memoria Flash de programa
- Hasta 368 bytes de memoria de datos (RAM)
- Hasta 256 bytes de memoria de datos EEPROM
- Hasta 4 fuentes de interrupción
- Stack de hardware de 8 niveles
- Reset de encendido (POR)
- Timer de encendido (PWRT)
- Timer de arranque del oscilador (OST)
- Sistema de vigilancia Watchdog timer.
- Protección programable de código
- Modo SEP de bajo consumo de energía
- Opciones de selección del oscilador

- Programación y depuración serie “In-Circuit” (ICSP) a través de dos pines
- Lectura/escritura de la CPU a la memoria flash de programa
- Rango de voltaje de operación de 2.0 a 5.5 volts
- Alta disipación de corriente de la fuente: 25mA
- Rangos de temperatura: Comercial, Industrial y Extendido
- Bajo consumo de potencia:
 - ♦ Menos de 0.6mA a 3V, 4 Mhz
 - ♦ 20 μ A a 3V, 32 KHz
 - ♦ Menos de 1 μ A corriente de standby

Clasificados como periféricos encontramos en su arquitectura:

- Timer0: Contador/Temporizador de 8 bits con pre-escalador de 8 bits
- Timer1: Contador/Temporizador de 16 bits con pre-escalador
- Timer0: Contador/Temporizador de 8 bits con pre-escalador y post-escalador de 8 bits y registro de periodo.
- Dos módulos de Captura, Comparación y PWM
- Convertidor Analógico/Digital: de 10 bits, hasta 8 canales
- Puerto Serie Síncrono (SSP)
- Puerto Serie Universal (USART/SCI).
- Puerto Paralelo Esclavo (PSP): de 8 bits con líneas de protocolo

Características	16F873	16F874	16F876	16F877
Frecuencia Máxima	DC-20Mhz	DX-20Mhz	DX-20Mhz	DX-20Mhz
Memoria de programa FLASH Palabra de 14 bits	4KB	4KB	8KB	8KB
Posiciones RAM de datos	192	192	368	368
Posiciones EEPROM de datos	128	128	256	256
Ports E/S	A, B y C	A, B, C y D	A, B y C	A, B, C y D
Nº de Pines	28	40	28	40
Interrupciones	13	14	13	14
Timers	3	3	3	3
Módulos CCP	2	2	2	2
Comunicaciones Serie	MSSP, USART	MSSP,USART	MSSP,USART	MSSP, USART
Comunicación Paralelo	-	PSP	-	PSP
Líneas de entrada en Convertidor A/D de 10 bits	5	8	5	8
Juego de Instrucciones	35 instrucciones	35 instrucciones	35 instrucciones	35 instrucciones
Longitud de la instrucción	14 bits	14 bits	14 bits	14 bits

Tabla 4.3 Comparación de las características de los PIC's

En la *Figuras 4.8 y 4.9* tenemos el diagrama de pines del PIC16F877 y el diagrama de bloques de la organización interna, para tener una visión al interior y exterior del chip.

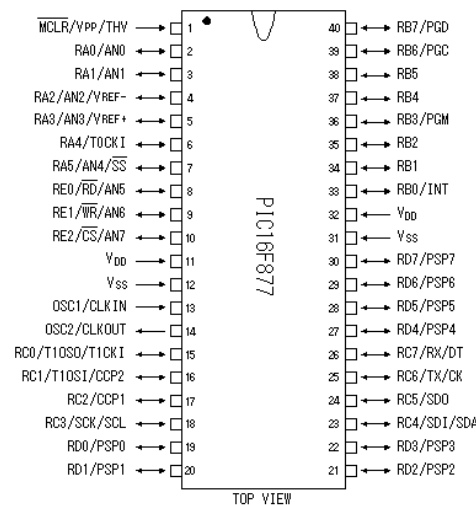


Figura 4.8 Diagrama de pines del PIC16F877. [W4.3]

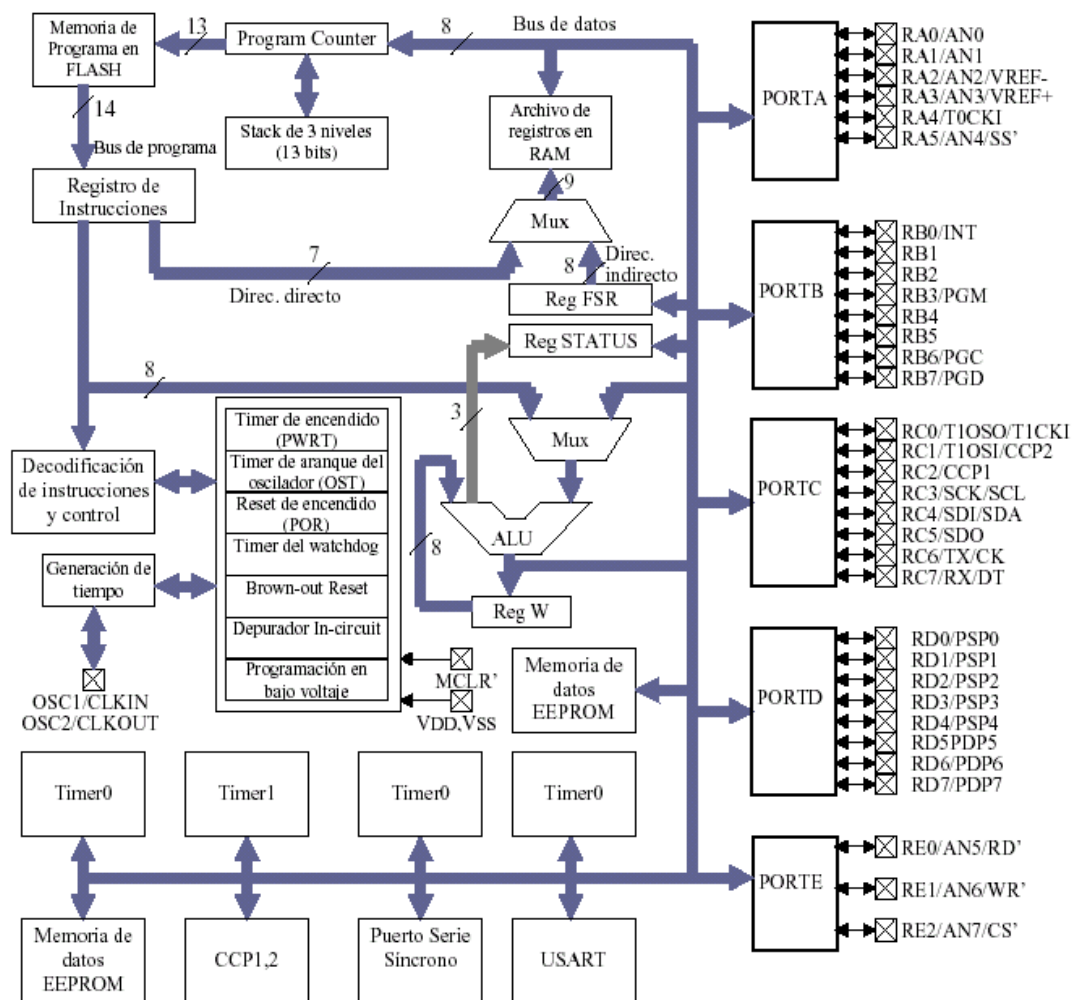


Figura 4.9 Arquitectura interna del microcontrolador PIC16F877.

CAPITULO 5

SINTETIZADORES DE VOZ

5.1 INTRODUCCIÓN A LOS SINTETIZADORES

5.1.1 Concepto

Un sintetizador es un dispositivo que genera y manipula sonidos por medios electrónicos. Con este artefacto se pueden crear nuevos sonidos así como reproducir los de los instrumentos musicales conocidos. La forma de la onda generada es alterada en su duración, altura y timbre mediante el uso de dispositivos tales como amplificadores, mezcladores, filtros, reverberadores, secuenciadores y moduladores de frecuencia.

5.1.2 Historia

El primer sintetizador apareció en 1955. Fue desarrollado por Olsen y Belar en la Radio Corporation of America (RCA) en Princeton. Estaba destinado a investigar las propiedades del sonido, no a ser un instrumento musical. Varios compositores se interesaron en ampliar las posibilidades de producción de sonidos y en controlar todos los aspectos de la música gracias a esta herramienta. Entre ellos destacan Milton Babbitt, Marion Davido, Morton Subotnik y Charles Wuorinen. [W5.1]

El sintetizador más conocido es el *Moog*, la primera máquina sencilla indicada hasta para el uso doméstico, que recibe su nombre del ingeniero estadounidense Robert Moog, quien desarrolló el primer prototipo en 1964. Apareció en la década de 1960 y se manejaba con uno o más teclados. Es posible crear un número casi infinito de sonidos y combinaciones y pueden imitarse muchos instrumentos. También pueden producirse sonidos que superan la capacidad de los instrumentos convencionales y que se combinan con los que captan los micrófonos. Además de su utilidad en la composición, los sintetizadores se utilizan para adaptar música instrumental, como en las conocidas grabaciones de Johann Sebastian Bach que Walter Carlos realizó en 1969. [W5.1]

El sintetizador *Moog* fue seguido por otros de diseño similar de empresas como Buchla y Putney. Entre 1960 y 1980 se fabricaron aparatos más reducidos que usaban microprocesadores, con lo que se hizo posible la grabación digital del sonido. Estos instrumentos han sido utilizados por músicos de formación clásica y grupos de música pop. [W5.1]

5.1.3 Aplicaciones del sintetizador de voz

Los sintetizadores como dispositivos intérpretes de señales a sonidos o mensajes de texto a voz humana, han permitido el desarrollo de diversas aplicaciones enfocadas principalmente al apoyo de personas con discapacidad tales como personas invidentes.

A continuación haremos mención de algunas aplicaciones de sintetizadores a través de algunos proyectos desarrollados:

SEÑALIZADOR BLUETOOTH

El Señalizador Bluetooth es un sistema *capaz de recibir y transformar mensajes de texto en mensajes de voz humana sintetizada* (Señalizador - RECEPTOR). Su objetivo principal es que busca aumentar la autonomía e independencia de las personas invidentes, para que cosas tan sencillas y normales como ir a comprar o quedar con los amigos para ver/escuchar un partido de fútbol no se conviertan en una “prueba de obstáculos”. [W.5.8]

El Señalizador-EMISOR puede ser instalado en paradas de autobús, en autobuses, paradas de metro, establecimientos, farmacias, etc. Este aparato va transmitiendo constantemente la información que con anterioridad le ha sido grabada (puede ser una información permanente o modificable por el usuario del sistema EMISOR). [W.5.8]

En cuanto a su funcionamiento, por ejemplo, en el caso de un autobús, al abrir las puertas o al acercarse a una parada comenzaría a emitir información auditiva en un radio de unos 5 a 10 metros (*la cual es ampliable según la tecnología de transmisión empleada, en este caso Bluetooth tiene una cobertura para alcanzar incluso hasta más de 100 metros, y también está diseñada para entornos Industriales*). En este caso el Señalizador Bluetooth tiene almacenada en su Señalizador-EMISOR la siguiente frase: << *BUS NÚMERO CINCO..., etc* >>. Así cuando un invidente se encuentre dentro del radio de cobertura de un Señalizador-RECEPTOR escuchara por el altavoz el mensaje anterior. [W.5.8]

Si fuera una parada de autobús, haría algo similar pero constantemente iría transmitiendo: << *PARADA DE AUTOBÚS DEL CENTRO..., etc* >>. O si se tratase de un establecimiento podría incluso informar del nombre del establecimiento y de las posibles ofertas, etc. Por ejemplo en una panadería: << *PANADERIA CASERA, PAN RECIEN HECHO A PARTIR DE LAS SEIS DE LA TARDE* >>. [W.5.8]

En resumen, las diversas aplicaciones desarrolladas y que han sido basadas en el empleo de sintetizadores, básicamente son sistemas y/o dispositivos que se han diseñado para determinados usos especiales y pensando en su viabilidad comercial. Para ello se han seguido las siguientes pautas:

- Sistema de bajo costo tanto de fabricación como de comercialización.
- Utilización de las últimas tecnologías de comunicación como es el caso de Bluetooth.
- Adaptado para las personas invidentes pero sin olvidar su posible utilización en otros sectores totalmente diferentes.
- Sistema de fácil utilización e instalación.
- Empleo de materiales que cuiden al máximo el medio ambiente (baterías recargables, placas solares, etc.)
- Utilización de componentes electrónicos comunes sin que ello desmerezca el producto final.

5.2 TIPOS DE SINTETIZADORES

Existe un gran número de sistemas de conversión texto-voz que incorporan una amplia variedad de aproximaciones diferentes para el proceso lingüístico, difíciles de estructurar de una manera sistemática. En consecuencia, los sistemas se suelen catalogar atendiendo a la técnica de proceso acústico que incorporen. Actualmente se utilizan métodos de síntesis que pueden clasificarse en tres grupos principales: los sintetizadores mediante modelos articulatorios, los sintetizadores de formantes y los sintetizadores basados en concatenación de unidades.

5.2.1 Sintetizadores de formantes o por Regla

En éstos la cadena fonética y la prosodia controlan las frecuencias, anchos de banda y la excitación de un conjunto de resonadores y antiresonadores. Un exponente claro de este tipo de sistemas lo constituye el sintetizador de Klatt. El sintetizador de formantes consiste en una composición de filtros que modelan las resonancias y antiresonancias de las cavidades vocal y nasal. Para este modelado se usan filtros que en la configuración más general están conectados en serie y en paralelo. Es un procedimiento de enorme flexibilidad que se pone de manifiesto en la alta calidad de la voz sintética que se puede obtener mediante ajuste manual de los parámetros del sintetizador. Sin embargo, se necesita un número enorme de reglas en la síntesis automática, lo que requiere compiladores cada vez más sofisticados capaces de integrar todo el conocimiento que se adquiere a base de experimentar con el sistema.

5.2.2 Sintetizadores mediante modelos articulatorios

En estos, se trata de simular la propagación de las ondas acústicas en el tracto vocal. Los segmentos y las variables prosódicas se traducen en parámetros de un modelo

simplificado del aparato fonador humano, que implícitamente restringen la dinámica del sistema, pudiendo producir voz de la más alta calidad. Surgieron para tratar de hacer corresponder explícitamente los sintetizadores de formantes con un modelo más explícito del tracto vocal. Su interés se centra en que las restricciones implícitas en este modelo permiten ver el habla como un continuo acústico, por lo que se solventan los problemas de concatenación de segmentos. Sin embargo, la dificultad principal de estos tipos de sistemas es que todavía no se conoce totalmente el proceso de producción del habla humana.

5.2.3 Sintetizadores basados en concatenación de unidades

En estos sintetizadores, como su propio nombre indica, se concatena un conjunto de unidades extraídas de producción humana. El representante más clásico de este tipo de sintetizadores es el conocido como PSOLA (Pitch Synchronous Overlap Add). En este tipo de sintetizadores debe estar presente un algoritmo que permita, además de la concatenación de unidades, modificar prosódicamente los segmentos a concatenar. Adicionalmente, se pueden usar técnicas de codificación de voz para reducir las necesidades de almacenamiento en la base de unidades acústicas. También existe la posibilidad de incluir en el modelo de codificación de voz las tareas de concatenación y modificación prosódica, siempre que el codificador parametrize la señal de voz con la suficiente flexibilidad para el modelado prosódico de las unidades.

5.3 PRINCIPALES FABRICANTES

5.3.1 RC System

RC System es una empresa líder de bajo costo en el mercado, en productos de alta calidad tales como intérpretes de texto y sintetizadores de voz. La empresa viene desarrollando este tipo de dispositivos desde 1983.

RC Systems ha desarrollado una amplia gama de sintetizadores de voz, máquinas de venta parlantes ATMs y dispositivos de seguridad con sistemas de telemetría vía satélite.

5.3.1.1 Sintetizador de voz V8600A

El sintetizador de voz V8600A (*Figura 5.1*) es un dispositivo versátil, un sistema modular de voz diseñado para ajustarse a los requerimientos de salida de voz.



Figura 5.1 Sintetizador de voz V8600A

El chip RC8660 instalado en el V8600A permite convertir automáticamente el texto de un lenguaje llano en una voz determinada de la alta calidad. El RC8660 incluye todo lo necesario para implementar síntesis de texto a voz con un total control dinámico de las características de la voz. En adición, varios tonos generadores han sido integrados para proveer de marcado telefónico, música y tonos de señales programables. Hasta 3.5 MB de datos en memoria flash pueden almacenar un número ilimitado de archivos de sonido (15 minutos en total) que pueden ser reproducidos al ser requeridos por el host. Además, un convertidor Analógico/Digital integrado de cuatro canales puede ser utilizado para recolectar datos analógicos de hasta cuatro diferentes fuentes. Y un puerto serie RS-232 compatible y un bus de 8 bits son incluidos para permitir que el RC8660 interactúe con cualquier CPU (*Figura 5.2*).

El V8600A mide menos de 30 centímetros cuadrados de tamaño, requiere solamente una fuente de 9 V y una bocina con alimentación de 5 V para funcionar. Este sintetizador, está diseñado para ser llevado sobre un host PCB, tal como tablero regulador de un sistema, vía dos conexiones con 12 pines de ocho bits de datos bidireccionales y pines de control de lectura/escritura permiten al V8600A ser interconectados fácilmente a cualquier línea de comunicación de microprocesador.

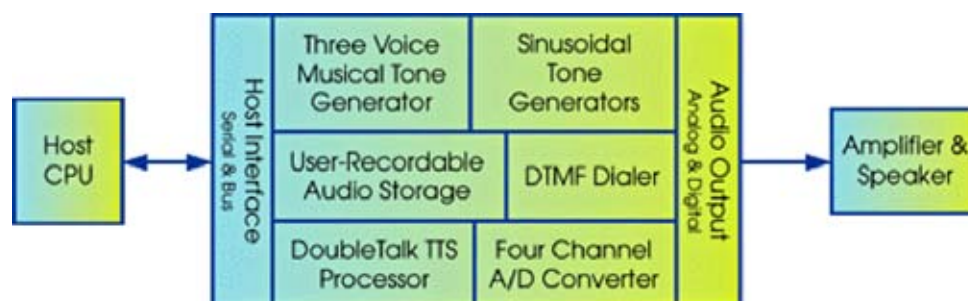


Figura 5.2 Bloques que integran la tarjeta del sintetizador V8600.

CARACTERÍSTICAS PRINCIPALES

- Características completas de un sintetizador de voz basado en los sistemas RC.
- Tecnología de texto a Voz DoubleTalk.
- Vocabulario ilimitado, convierte texto en Inglés a voz automáticamente.
- Parámetros de la voz programables durante la ejecución (tipo de voz, pitch, rango, volumen, tono, expresión, articulación, etc.)
- Capaz de altos rangos de voz y receptores superiores.
- Reproducción de hasta 15 minutos de mensajes y efectos de sonido pre-grabados.
- Almacenamiento no volátil de la configuración del usuario, del diccionario y del mensaje de bienvenida.
- Circuito 100% programable.
- Tonos generadores senoidales y musicales.
- Marcador DTMF (Touch-Tone).
- Controles de Stop, pausa y continuar.
- Salidas de audio analógicas y digitales.
- Convertidor Analógico/Digital de 8-bits de cuatro canales para monitorear baterías, adquisición de datos, etc.
- Operación a 3.3 V o 5 V.
- Suministro de corriente baja: activo a 7mA; ocupado a 0.7 mA; en espera a 2 μ A (Operación a 3.3 V)
- Puerto serie integrado; soporta buses y puertos de impresora.
- Programable y actualizable a través del Puerto serie.

FUNCIONAMIENTO DEL V8600A

El RC8660 se comunica con el procesador del host por medio del Puerto serie integrado o el bus. Cuando los mensajes de texto son enviados al RC8660, automáticamente el RC8660 convierte el texto en voz utilizando un procesador integrado de texto a voz (Text to Speech [TTS]). El procesador TTS utiliza la tecnología DoubleTalk de los Sistemas RC (RC Systems), basada en una técnica de concatenación patentada que utiliza muestras reales de la voz humana. Los parámetros de la voz, tales como la velocidad, el volumen, el tono, el pitch y la expresión, pueden ser embebidos en el flujo del texto para el control de la voz. El diccionario de excepciones definido por el usuario permite la pronunciación de cualquier secuencia a ser redefinida, e incluso provoca la reproducción de sonidos y/o tonos pre-grabados. [5.2]

Los tonos generadores integrados son activados utilizando comandos embebidos en la entrada de la cadena, permitiendo que los tonos sean generados en cualquier punto del mensaje. El RC8660 puede generar tres voces musicales, DTMF (Touch-Tone), y tonos senoidales como se requieran.

Archivos de sonido (tales como archivos wav) han sido almacenados en el RC8660 que son indexados internamente. Por ejemplo, los primeros 64 sonidos son referenciados como sonidos 0-63. Para reproducir cualquiera de estos sonidos, el procesador del host envía simplemente los valores indexados correspondientes a los sonidos a ser reproducidos. Las reproducciones requeridas son almacenadas en un chip, y pueden ser combinadas en cualquier orden – incluso con texto para el procesador TTS. Los archivos de sonido pueden ser actualizados en la tarjeta del sistema a través del puerto serie en cualquier momento.

Los cuatro canales del convertidor Analógico/Digital de 8 bits pueden ser utilizados para monitorear voltajes de baterías, temperatura, y otras cantidades analógicas. El CAD puede ser programado para convertir algún canal solamente, o escanear estos repetitivamente.

Debido a que toda entrada es almacenada en el RC8660, se pueden escribir simples mensajes y comandos para el RC8660 y olvidarte de éstos – el RC8660 hace el resto. Mensajes adicionales, comandos y reproducciones requeridas pueden ser hechos incluso cuando el RC8660 esté produciendo una salida. La salida de audio es considerada como una señal analógica; también una salida serie digital está disponible para aplicaciones que requieren procesamiento del audio stream.

El tamaño pequeño del sintetizador V8600A y su consumo de energía bajo lo hacen una solución ideal para cualquier uso, extendiéndose de la robótica a las telecomunicaciones.

5.3.2 Winbond Electronics Corporation

Winbond Electronics Corporation fue fundada en 1987 en el parque Industrial Hsinchu de Ciencias en Taiwán. Después de 16 años de crecimiento y acumulación de activos generosos en productos y tecnología, en Winbond cuentan con un planteamiento de operación sistemático y en serie para todo, al principio del diseño de un producto, durante su producción, la comercialización y en el sistemas de desarrollo. Winbond es conocida por el desarrollo de productos bajo su propia marca y se esfuerza por tener como objetivo una amplia gama de las líneas de productos para resolver las demandas del desarrollo de la industria de la información y para satisfacer las necesidades de los clientes. Winbond es la marca más grande de distribuidor de circuitos integrados en la industria del IC en Taiwán en búsqueda de la excelencia en la tecnología de proceso.

Winbond combina tecnología de innovación con una profunda experiencia en manufactura para crear productos con valor agregado para sus clientes. Comúnmente diseña y fabrica semiconductores avanzados, incorporando a éstos la conectividad inalámbrica, media digital y tecnología de almacenamiento en memoria a bajo consumo de potencia, lo que los convierte en productos altamente eficaces y de alta capacidad para los sistemas en una tarjeta.

En el área de síntesis de voz, uno de los productos destacables en Winbond es el chip procesador de texto a voz nombrado WTS701. El WTS701 es una solución que convierte el Texto a Voz de alta calidad y totalmente integrado para aplicaciones automovilísticas, sistemas de navegación GPS, teléfonos celulares y otros productos o accesorios portátiles. El WTS701 acepta código ASCII como entrada a través de la interfaz SPI y convierte ésta en voz en una salida analógica o digital. [5.1]

El WTS701 integra un procesador de texto, un filtro y arreglos de almacenamiento de memoria de multi-nivel en un solo chip. La conversión de texto a voz es lograda con base a la síntesis de voz por concatenación al procesar el texto de entrada en una representación fonética que es mapeada a una recopilación de partes de palabras habladas naturalmente. El algoritmo de síntesis tiende a utilizar la unidad de palabra más larga posible en el contexto apropiado para maximizar la calidad del sonido natural. Las unidades de voz descomprimidas son almacenadas en un multi-nivel, en un arreglo de almacenamiento analógico no volátil para obtener un sonido de la más alta calidad a una densidad de compensación. Esta solución única en un solo chip es hecha posible a través de la tecnología de almacenamiento de multi-nivel patentada por Winbond. Las señales de voz y audio son almacenadas directamente en memoria de estado sólido en su reproducción de voz natural y de forma descomprimida, provistas de una calidad superior. [5.1]

Este chip puede ser programado a través del puerto serie (SPI), permitiendo la descarga de lenguajes diferentes y bases de datos de voces cuando las hace disponibles Winbond [5.1]. En la siguiente parte se describe una implementación de este chip que es utilizada en aplicaciones de Robótica.

5.3.2.1 Sintetizador de Voz I2C para robot

CARACTERÍSTICAS

La comunidad robótica estuvo sin un sintetizador de voz de bajo coste mucho tiempo. Con la desaparición del popular chip SP256-AL2, que era capaz de reproducir 256 fonemas diferentes, sólo quedaron en el mercado soluciones multichip hasta la aparición del procesador Winbond WTS701. Este integrado no solo incluye el chip sintetizador, sino que además incluye un procesador capaz de reproducir texto con unas características y calidad buenas, siendo el único inconveniente para el aficionado, el encapsulado en un chip de 56 patitas en formato TSOP con un espaciado entre patillas de tan solo 0,5 mm.

El SP03 (Ver Figura 5.3) es un circuito sintetizador de voz capaz de convertir en sonidos cualquier texto de forma ilimitada que se le envíe por el puerto serie RS232, o bien mediante un bus I2C. El circuito sintetizador también es capaz de almacenar hasta 30 frases con un máximo total de 1925 caracteres. Puede funcionar bajo el control de un bus I2C, un interfaz serie RS232 a 38400 baudios o bien puerto de entrada paralelo de 5 pines. El circuito de tan solo 40 x 40 mm incluye un amplificador de audio de 325 mW y un altavoz, por lo que tan solo es necesario la alimentación y las conexiones de control para su funcionamiento. Alimentación 5 V, 35 mA en reposo y 100 mA activo. Utiliza el circuito sintetizador Winbond WTS701EM/T, en versión inglesa.



Figura 5.3 Módulo sintetizador SP03. [W5.7]

El módulo sintetizador de voz SP03, incluye un amplificador de audio, un regulador de tensión de 3 voltios con conversión a 5 V, un procesador PIC encargado de facilitar la comunicación con el procesador central, un altavoz de 40 mm y el chip sintetizador WTS701. Para facilitar la conexión con exterior se incluye un bus I2C, un puerto serie RS232 y un bus paralelo capaz de reproducir hasta 30 frases diferentes. Hay disponible un programa para PC llamado SP03.EXE capaz de enviar y reproducir las 30 frases en el módulo sintetizador. También existen varios ejemplos de utilización del módulo sintetizador con varios microcontroladores.

5.3.3 ScanSoft Speech Solutions

El ScanSoft TTS-2500 es parte de la amplia gama de productos por parte de ScanSoft Speech Solutions, es una solución de alta calidad como dispositivo que permite emular virtualmente cualquier sonido de voz. [W5.9]

Con el ScanSoft TTS-2500 se han desarrollado diversas aplicaciones que van desde un sencillo generador de avisos con voz hasta un sofisticado sistema intérprete de texto con sonido natural de voz y con vocabularios ilimitados. [W5.9]

El dispositivo cuenta con múltiples lenguajes y adicionalmente algunos más se encuentran en desarrollo. En las aplicaciones desarrolladas se puede optar por reproducir tanto voces varoniles como femeninas, o bien, es posible la configuración para emplear un determinado tipo de voz modificada para propósitos específicos. [W5.9]

CARACTERÍSTICAS Y VENTAJAS [W5.9]

- Bajo consumo de energía – Ideal para dispositivos manuales y portables.
- Permite que la batería opere con un bajo nivel de voltaje.
- En modo de escucha (standby) se reduce la energía a menos de 10 μ A.
- Conductor directo de altavoz – reduce los costos del sistema.
- Múltiples algoritmos sobre el chip.
- Múltiples lenguajes – actualmente cuenta con los siguientes lenguajes: inglés americano, francés, alemán, español, chino mandarín, etc.
- Selección de voces femeninas y masculinas, y/o modificaciones de voz.

- Sistema de vocabulario ilimitado – de acuerdo con reglas lingüísticas y fonéticas profundizadas, con soporte para cualquier tipo de texto como entrada (e-mail, fechas, páginas Web, etc.).
- Avanzado modelo de sonido natural.
- Altamente exacto.
- Bajo costo de operación – esencialmente y altamente competitivo en la industria de consumo eléctrico.
- Algoritmos avanzados de habla – nunca obteniendo bajos niveles de calidad en la interpretación de datos.
- Interpretación de sonidos naturales mientras sean empleadas pequeñas cantidades de memoria.
- Calidad superior – reproducción de un sonido claro de voz humana (con buena entonación natural).

APLICACIONES

El ScanSoft TTS-2500 es capaz de permitir virtualmente la interpretación, en conjunto con cualquier dispositivo, particularmente con los dispositivos electrónicos siguientes:

- Consumo
 - Reloj parlante
 - Ayuda en la navegación
 - Aparatos electrodomésticos
 - Audio libros
 - Teléfonos celulares
- Industria
 - Sistemas de control de advertencias
- Telecomunicaciones
 - Contestadoras automáticas
 - Sistemas de correo de voz
- Educación
 - Asistentes de aprendizaje electrónicos
 - Diccionarios parlantes
 - Traductores de lenguaje
- Seguridad
 - Sistemas de monitoreo (seguridad)

ESPECIFICACIONES

El TTS-2500 representa un estándar con algoritmos implementados en hardware propiedad de ScanSoft. Este hardware en forma combinada con un procesador permite una amplia gama de usos a muy bajo costo y con el mínimo esfuerzo. [W5.9]

Frecuencia del reloj de Cristal	32.768 kHz
Suplemento de Voltaje	3 - 5 V
Impedancia a la salida del altavoz	32 Ω
Corriente esperada	< 10 μ A

DESCRIPCIÓN FUNCIONAL

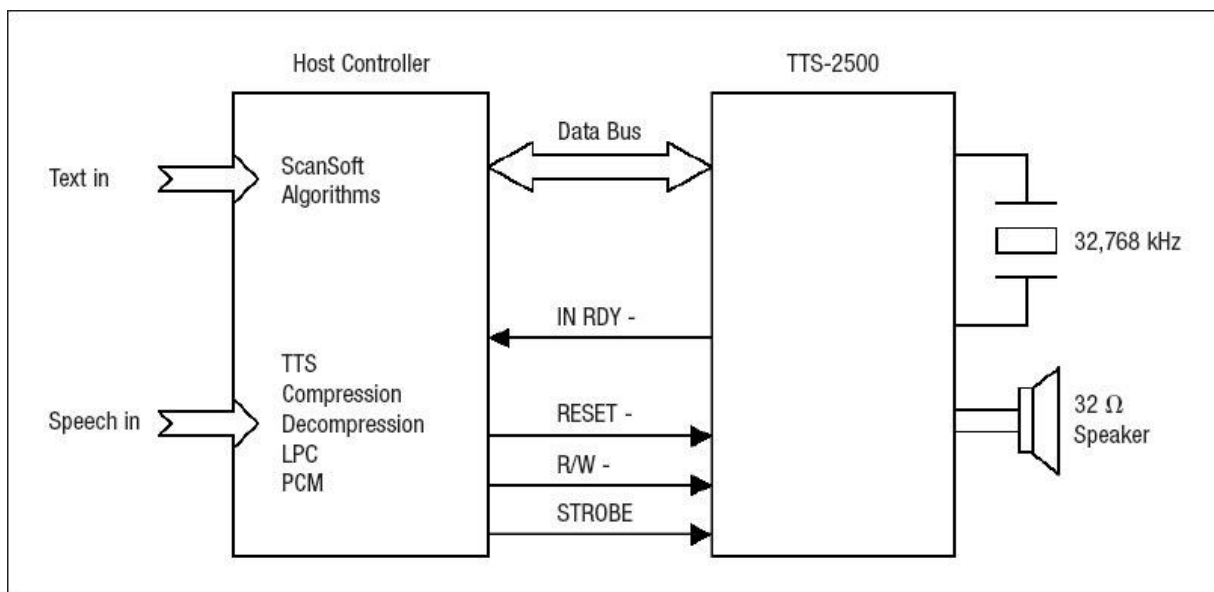


Figura 5.4 Diagrama de funcionalidad del sintetizador de voz ScanSoft TTS-2500. [W5.9]

5.4 AVANCES EN LOS SINTETIZADORES.

VOCES DE ALTA CALIDAD

La última tecnología en voces sintetizadas, es la utilizada por los laboratorios de la compañía AT&T. Esta tecnología se conoce como "Natural Voices" (Voces naturales), y la misma está basada en una combinación de voz humana con programación. Estas voces son de alta calidad y se parecen mucho a la voz humana.

VOCES DE BUENA CALIDAD

La mayoría de los programas y aplicaciones que utilizan las personas ciegas, tienen este tipo de voz de buena calidad. Esto incluye a los sintetizadores Eti Eloquence, usados por el popular programa lector de pantalla JAWS. Otro sintetizador lo es el IBM Via Voice, usado por los programas Open Book y el IBM Home Page Reader. Todos estos sintetizadores pueden ser utilizados por las aplicaciones que apoyen sintetizadores de tipo SAPI en Windows.

VOCES GRATUÍTAS TIPO SAPI

Estas voces pueden ser descargadas gratuitamente desde el sitio de los "Microsoft Agents". Son de calidad aceptable, y pueden funcionar con cualquier aplicación como sintetizador SAPI. Estas voces son de la compañía L&H TTS.

CAPÍTULO 6

IMPLEMENTACIÓN DEL SISTEMA

6.1 INTRODUCCIÓN

La mano humana es una parte muy importante de nuestro cuerpo, no solo nos es de gran utilidad en la mayoría de las actividades físicas que realizamos, también nos permite, en gran medida, el contacto y comunicación con el mundo que nos rodea. Como ya se había mencionado en el capítulo 2, la estructura de la mano es muy compleja, es por eso que para diseñar nuestro sistema debemos conocer bien sus movimientos si queremos construir un buen prototipo, que sea ergonómico, económico y sobre todo, que sirva a las personas con discapacidad auditiva.

La selección correcta de elementos o dispositivos en cualquier diseño de un sistema, es determinante para su correcto funcionamiento; por tal motivo es de gran importancia tener la suficiente información de cada componente que formará parte del sistema, teniendo en cuenta los requerimientos del mismo.

6.2 DESCRIPCIÓN Y ESTRUCTURA GENERAL

La idea general que hemos pensado para la implementación de nuestro proyecto es construir un guante con sistema sensorial, lo más ajustable posible a la mano; integrado por sensores que detecten los movimientos de los dedos de la mano para así generar señales eléctricas a partir de alguna representación manual. En seguida debemos preparar las señales generadas por los sensores del guante para procesarlas con un microcontrolador y por último enviarlas hacia otro dispositivo para convertirlas a voz, es decir, un sintetizador de voz. Esta descripción se puede visualizar mejor en un diagrama de bloques como se muestra en la *Figura 6.1*.

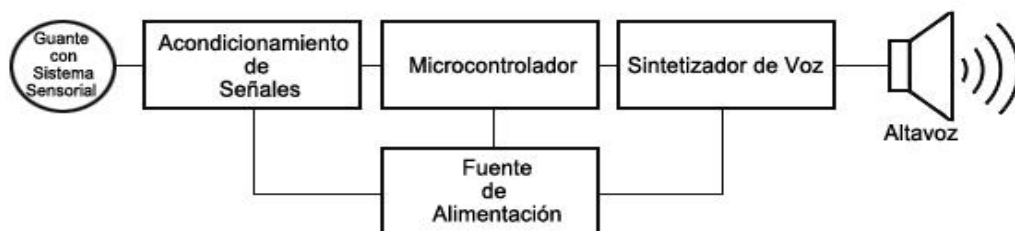


Figura 6.1 Diagrama de bloques del sistema traductor de señas dactilológicas a voz.

La **Fuente de Alimentación** proporciona la polarización adecuada a cada uno de los distintos bloques que así lo requieran. El **Guante con Sistema Sensorial** consta de un guante y de sensores colocados en la parte superior de los dedos, que, a partir de una representación manual hecha por el usuario, generará señales eléctricas. El bloque **Acondicionamiento de Señales** prepara las señales generadas en los sensores para que puedan manipularse. El **Microcontrolador** tiene la tarea de analizar las señales acondicionadas y compararlas con las palabras o frases previamente definidas por nosotros y almacenadas en memoria, donde, en caso de ser iguales, la información es enviada al siguiente bloque. El **Sintetizador de Voz** recibe la información proveniente del microcontrolador y la convierte en una señal analógica de audio que será enviada al **Altavoz** para ser escuchada.

6.3 SELECCIÓN DE MATERIALES, CONSTRUCCIÓN DEL PROYECTO

6.3.1 Construcción del Guante con Sistema Sensorial

6.3.1.1 Primer Prototipo

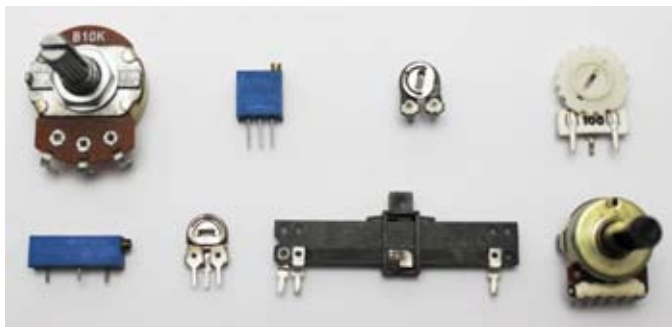
A partir del diagrama de bloques anterior, procedimos a la construcción del guante con sistema sensorial; para su construcción elegimos un guante de un material flexible y que se ajustara lo mayor posible a la mano humana. El primer guante seleccionado fue un guante de tela con puntos de PVC de uso industrial como se ilustra en la *Figura. 6.2*.



Figura 6.2 Guante de tela con puntos de PVC.

A continuación colocamos el sistema sensorial en el guante, para esto consideramos las siguientes opciones. La primera era un transductor resistivo llamado sensor de flexión; este transductor es el más óptimo para utilizar en este tipo de sistemas debido a su facilidad para ser montado y conectado, pero quedó descartada en un principio debido a su alto costo (alrededor de \$8.00 USD cada uno).

La siguiente opción era otro transductor resistivo llamado potenciómetro; analizamos sus diferentes formas (ver *Figura 6.3 (a)*), y decidimos que el más óptimo para nuestro proyecto era el potenciómetro deslizante de carbón de tipo monoaural (*Figura 6.3 (b)*) debido a su bajo costo y eficiencia; investigamos sobre los valores que podríamos utilizar en nuestra aplicación y concluimos que debe ser de un valor de resistencia alta para evitar un valor de corriente que pudiera dañar al convertidor analógico-digital del microcontrolador.



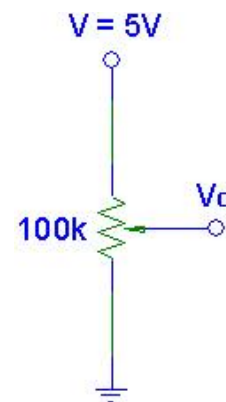
(a) Tipos de potenciómetros



(b) Potenciómetro de carbón deslizante tipo monoaural.

Figura 6.3 Algunos tipos de potenciómetros.

Para realizar pruebas y elegir el que tuviera un mejor comportamiento utilizamos diferentes valores de resistencia (20 k Ω , 50 k Ω y 100 k Ω). Tomamos uno de cada valor y utilizando un multímetro digital, realizamos pruebas para medición de voltaje, conectando el potenciómetro en una configuración divisor de voltaje, es decir, conectando una de sus terminales fijas a la terminal positiva de la fuente de voltaje de 5 V y la otra terminal a la tierra y midiendo en cada uno de ellos el voltaje en la terminal del cursor con respecto a tierra conforme desplazamos el cursor para ver la variación de voltaje en relación con el desplazamiento como se observa en la *Figura 6.4*. Encontramos que los tres tenían aproximadamente la misma variación y decidimos utilizar el potenciómetro de 100 k Ω para mayor

**Figura 6.4** configuración divisor de voltaje del potenciómetro.

El siguiente paso fue montarlos en el guante; analizamos en qué lugar podrían quedar mejor para no obstruir el movimiento de los otros potenciómetros, y decidimos colocarlo sobre los huesos metacarpianos correspondientes a cada dedo. En las *Figuras 6.5* y *6.6* podemos observar los huesos de la mano y el montaje de los potenciómetros en el guante respectivamente.



Figura 6.5 Huesos de la mano. [2.9]



Figura 6.6 Colocación de potenciómetros sobre los huesos metacarpianos de los dedos en el guante.

Fijamos en primer lugar el potenciómetro asociado al dedo medio utilizando silicón, tomando en cuenta el desplazamiento del cursor y las conexiones eléctricas que haríamos para realizar las pruebas correspondientes. Y para desplazar el cursor de una posición a otra en ambas direcciones con el movimiento natural del dedo, adaptamos un cable de acero de 2 mm de diámetro fijando un extremo a la punta del dedo y el otro al cursor del potenciómetro.

Realizamos algunos movimientos para comprobar el desplazamiento del cursor, pero el cable de acero se salía del eje de desplazamiento, entonces colocamos una guía sobre la que se desplazara el cable en una sola dirección y no a los costados; colocamos un tubo rígido transparente de 2.5 mm de diámetro que obtuvimos de algunos repuestos inservibles de plumas de tinta. Despegamos uno de los extremos del cable, cortamos trozos de 1 cm de largo de tubo y colocamos cada uno sobre las falanges del dedo medio de igual manera con silicón y nuevamente fijamos el extremo del cable que habíamos despegado. Nuevamente, hicimos pruebas visuales abriendo y cerrando el dedo, revisando que el cable tuviera movimiento libre y continuo en ambas direcciones.

De la misma manera como fijamos el dedo medio, fijamos los otros cuatro dedos restantes continuando con el dedo índice, anular, meñique y pulgar, siendo en este último el más complicado de colocar el potenciómetro por el tipo de movimiento que tiene; siempre evitando la obstrucción del movimiento en los otros dedos.

Finalmente hicimos movimientos con todos los dedos; individualmente y en conjunto, encontrando a simple vista que tendríamos un buen funcionamiento. Posteriormente colocamos cable de cobre del #32 para las conexiones eléctricas, soldando tramos de aproximadamente 40 cm en las terminales de los potenciómetros pero dejando libre los otros extremos de los cables. En la *Figura 6.7* se ilustran imágenes de nuestro primer prototipo.



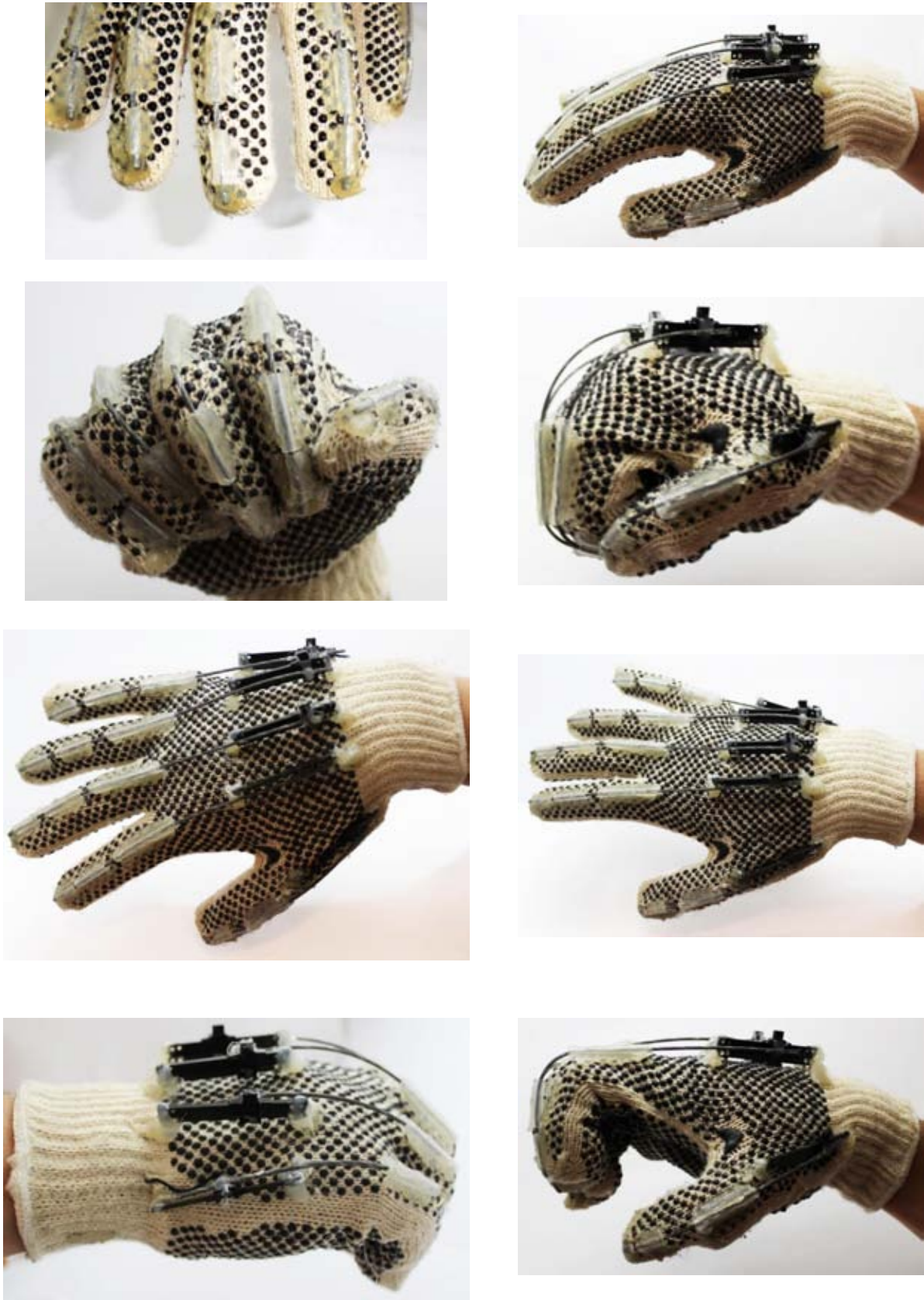


Figura 6.7 Primer prototipo del guante con sistema sensorial.

6.3.1.2 Segundo Prototipo

A partir del resultado de las pruebas (*Ver detalle de pruebas en Capítulo 7*) del prototipo anterior decidimos rediseñarlo, es decir, cambiamos el guante y los sensores, con la finalidad de evitar los problemas detectados. (En el capítulo 7 se detallan las pruebas realizadas).

El guante lo cambiamos por otro que se adaptara bien a nuestra mano. Seleccionamos un guante de piel de cabra, muy suave, flexible y ajustado para continuar con el proyecto. (*Ver Figura 6.8*).



Figura 6.8 Guante de piel de cabra.

Seleccionamos el nuevo sensor a utilizar, elegimos la opción que habíamos descartado previamente, los sensores de flexión. Buscamos los que tuvieran las características funcionales y económicas más adecuadas para nuestro proyecto, así como el fabricante. Encontramos y contactamos la empresa FLEXPPOINT® en la ciudad de Utah, Estados Unidos, y adquirimos 10 unidades de 7 cm cada uno con un costo total aproximado de \$80 USD. (*ver figura 6.9*).



Figura 6.9 Sensor de flexión.

Analizamos las dimensiones y la forma de los sensores con el objetivo de montarlos de la mejor manera sobre el guante. Nos colocamos el guante y sobre él, sobrepusimos los sensores para ver de qué manera podríamos fijarlos, encontrando que la mejor opción era fijarlos directamente sobre el guante pero utilizando guías de plástico flexible para asegurar el movimiento sobre el eje del movimiento de cada dedo y para evitar dañarlos.

Comenzamos primero con el dedo medio, observamos que un sólo sensor no era suficiente para abarcar todas las articulaciones de este dedo, ya que es el más largo de todos, lo mismo sucedía con los dedos anular e índice, entonces colocamos dos sensores en esos tres dedos, mientras que en los otros dos dedos: pulgar y meñique, un sólo sensor nos bastó para detectar el movimiento completo del dedo.

A continuación buscamos las guías que íbamos a utilizar para el correcto deslizamiento de los sensores de flexión; tomando en cuenta que son planos, tienen forma rectangular y son muy delgados. Probamos con varias cosas: grapas de papelería, alambre magneto y una guía de plástico, que nos pareció la mejor opción. Las dos primeras opciones las despreciamos porque los sensores de flexión se pueden dañar con facilidad permanentemente y además de que tendrían que estar lo más cercano posible a los huesos de la mano, excepto en las articulaciones; por lo anterior decidimos que la mejor opción era la guía de plástico flexible.

Fabricaríamos las guías utilizando un molde más ancho y de mayor espesor que el sensor, utilizando una lámina de acero o algún material rígido. Las dimensiones de las guías que necesitábamos tendrían 1 cm de largo y 0.5 cm de ancho aproximadamente, entonces cortamos una lámina con esas medidas y la limamos para evitar que el plástico se rompiera. Después para construirlas, utilizamos un tubo transparente de 0.7 cm de diámetro llamado *thermofit* o *termofit* que fácilmente cubría el molde, cortamos trozos de tubo de 1 cm de largo y las colocamos sobre el molde; aplicamos calor con una pistola de aire caliente para que el tubo se contrajera hasta alcanzar la forma del molde, hicimos 20 piezas para todos los dedos.

Posteriormente fijamos las guías en el guante, colocándose uno de nosotros el guante para mayor facilidad; fijamos una a una las guías sobre cada falange de los dedos con silicón y guiándonos con el sensor, es decir, haciendo una simulación de los movimientos individuales de cada dedo para que no hubiera obstrucción del deslizamiento de cada sensor. Una vez finalizado el montaje de las guías, fijamos los sensores en el guante, colocando una pequeña gota de silicón en la base de los conectores de plástico, evitando que el silicón tapara los orificios de los conectores. De la experiencia del prototipo anterior, antes de hacer pruebas de voltaje hicimos algunas con los movimientos de la mano y el guante para comprobar que los sensores y las guías quedarán bien fijos, así como también verificamos que no existieran obstrucciones.

Después de realizar movimientos repetidamente, encontramos que los sensores se deformaban y se atoraban, debido a que las guías estaban montadas directamente sobre los dedos y al cerrar la mano, la guía prensaba los sensores, evitando su deslizamiento y al mismo tiempo dañándolos. Despegamos nuevamente los sensores y las guías con cuidado para volver a montarlos de otra forma. Colocamos bases a las guías para evitar que estas prensaran a los sensores.

Ya que nosotros tenemos tamaños de mano diferente, elegimos la de tamaño más estándar para que se adaptara con más facilidad a las personas que pudieran utilizar el guante; en seguida comenzamos a hacer mediciones de las falanges de cada uno de los dedos para hacer los cortes correspondientes. Las medidas de los cortes se muestran en la *Tabla 6.1*.

Dedos	Falanges		
	Proximal (cm)	Media (cm)	Distal (cm)
Pulgar	2.0	-	2.5
Índice	1.5	2.0	3.5
Medio	1.5	2.2	4.0
Anular	1.3	2.0	3.3
Meñique	1.0	1.5	2.0

Tabla 6.1 Medidas de los cortes de las bases para las guías de los sensores de flexión

El material con el que fabricamos las bases lo obtuvimos de un transformador eléctrico descompuesto; quitamos el alambre de las bobinas y finalmente desarmamos el núcleo con el que obtuvimos las laminillas de material ferromagnético que utilizamos para las bases, con forma rectangular de 0.8 cm de ancho por 4.5 cm de largo. Cortamos las laminillas con base en las mediciones de la *Tabla 6.1*.

Una vez que hicimos todos los cortes, lijamos los elementos para evitar accidentes o que se dañaran los elementos del sistema. Con las medidas que teníamos de las bases, cortamos los trozos de tubo *termofit* para rehacer las guías, siguiendo el mismo procedimiento descrito anteriormente.

Con el guante colocado en la mano de uno de nosotros y apoyándonos sobre una base para evitar movimientos, fijamos primero las bases, después las guías y por último los sensores; utilizando silicón y revisando que estuvieran en línea para evitar errores. Realizamos movimientos con la mano para ver el funcionamiento del guante y encontramos que el deslizamiento de los sensores no se realizaba correctamente y además se despegaron las bases del guante.

Entonces repetimos el procedimiento, pero ahora utilizando pegamento instantáneo para conseguir una mejor fijación entre la laminilla y la piel del guante; y utilizando como guía los sensores sobrepuestos y un marcador para indicar la trayectoria que debería seguir el sensor y así evitar obstrucciones en el eje de movimiento de los dedos. Para finalizar fijamos con silicón los sensores en su posición, y nuevamente hicimos movimientos con la mano. Esta vez observamos un buen deslizamiento de los sensores y una mejor fijación que la anterior, sin embargo, el guante presento pliegues por el tipo de piel del guante y comenzó a deformarse lo que impedía una correcta flexión de los sensores. En las *figuras 6.10 (a) y 6.10 (b)* se muestran algunas imágenes sobre la construcción del segundo prototipo del guante con sistema sensorial.



Figura 6.10 (a) Construcción del segundo prototipo del guante con sistema sensorial.



Figura 6.10 (b) Construcción del segundo prototipo del guante con sistema sensorial.

6.3.1.3 Tercer Prototipo (Prototipo Final)

Para la construcción del tercer prototipo utilizamos un guante de estambre que se ajustaba totalmente a la mano. El material del nuevo guante no se deforma con el movimiento de los dedos de la mano, no presentaba pliegues o marcas que realmente fueran un obstáculo en la flexión de los sensores gracias a que se amolda mejor a la forma de la mano (Ver figura 6.14).

En la construcción reutilizamos las bases y los sensores de flexión del guante anterior, y volvimos a construir las guías tratando de mejorarlas, de tal manera que quedarán por unos milímetros más anchas que en el segundo prototipo, para obtener un mejor deslizamiento cuando se hacen movimientos de apertura entre los dedos.

Para rehacer las guías de la misma forma como en el guante de piel, cortamos una lámina de 1 cm de largo y 0.5 cm de ancho aproximadamente. Realizando el mismo procedimiento que en el prototipo anterior, construimos las 20 piezas nuevamente con el tubo *termofit*, en la figura 6.13 observamos algunas de estas piezas terminadas.



Figura 6.13 Bases y guías utilizadas para montar los sensores de flexión en el guante.

Debido a que el guante era de estambre, decidimos utilizar trozos de velcro, utilizando una de las dos partes del velcro para fijar las bases de las guías y los sensores en vez de emplear únicamente las laminillas de acero directamente sobre el guante, como lo hicimos en el guante de piel.

Las laminillas que soportan las guías de plástico y las bases de los sensores, fueron fijadas con pegamento instantáneo y silicón respectivamente sobre trozos de velcro; mientras que la otra parte del velcro la cosimos sobre la superficie del guante considerando la ubicación de cada una de las guías y las articulaciones de la mano en general.

Esta implementación nos permitió montar y desmontar cada uno de estos componentes sobre la superficie del guante, haciendo al prototipo mucho más práctico en las pruebas y a la vez más adaptable a cualquier tamaño y forma de mano (*Ver figura 6.14*).



Figura 6.14 Colocación de trozos de velcro en el guante de estambre previo a la fijación de las bases y los sensores.

A continuación uno de nosotros se puso el guante mientras que el otro colocó las guías de plástico en su sitio, cuidando siempre la alineación de éstas; en seguida fijamos la base de los sensores en la posición correspondiente e introducimos los sensores dentro de las guías.

Así quedó concluida la construcción del guante con sistema sensorial, en la *figura 6.15* se muestran algunas imágenes sobre la construcción del tercer prototipo, el prototipo definitivo.



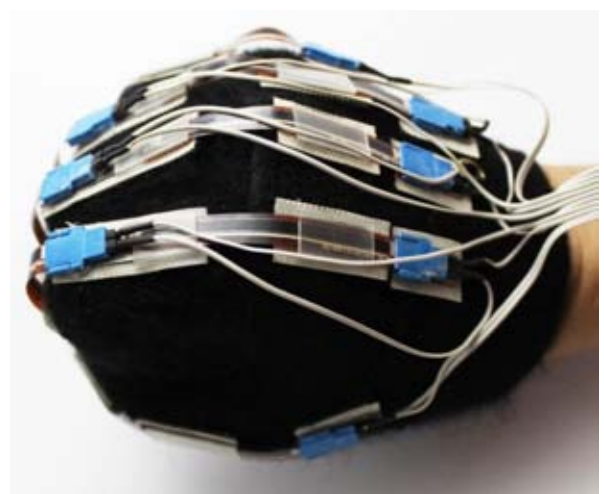




Figura 6.15 Construcción del prototipo final del guante con sistema sensorial.

6.3.2 Elección del microcontrolador

Con el guante terminado, comenzamos la selección del microcontrolador que formaría parte de nuestro sistema. Teníamos algunas opciones para elegir (descritas en el capítulo 4): el microcontrolador 8051 de Intel; el PIC 16F877 de Microchip y el microcontrolador MC68HC11F1 de Freescale Semiconductor. Elegimos la tercera opción porque ya lo teníamos junto con su tarjeta la tarjeta de aplicación y además esta tarjeta tiene integrados los sockets necesarios para trabajar en modo expandido y los periféricos a utilizar en esta aplicación.

El microcontrolador **MC68HC11F1** que seleccionamos, viene montado en la tarjeta de aplicación llamada **FACIL11_B** que fue desarrollada en la Facultad de Ingeniería de la UNAM, y que está diseñada para utilizar todos los recursos del microcontrolador como se muestra en la *figura 6.16*. Es por eso que para aplicarla adecuadamente en nuestro prototipo fue necesario recopilar los documentos que se refieren al microcontrolador y a la misma tarjeta, como es el caso del manual de referencia del fabricante y del manual de configuración de los recursos de la tarjeta respectivamente. Nuestro prototipo requirió de varios recursos internos del microcontrolador como: el convertidor analógico-digital (convertidor A/D), memoria RAM, memoria EEPROM, puertos de entrada y/o salida y puerto de comunicación serie asíncrona SCI.



Figura 6.16 Tarjeta de aplicación para el microcontrolador MC68HC11F1 (FACIL_11B).

6.3.3 Implementación del Sintetizador de Voz

6.3.3.1 Elección y adquisición del Sintetizador

Para la elección del sintetizador consideramos que el dispositivo que debería utilizar la síntesis de “Texto a Voz”, en inglés: “Text to Speech” (TTS); es decir, que recibiera cadenas de texto como entrada y las convirtiera en voz a la salida.

Investigamos y el primer sintetizador que encontramos fue el circuito integrado **WTS701EF/T** de la empresa WINBOND. Es un procesador de texto a voz que funciona con 3 Volts, recibe un formato de entrada en ASCII con una salida analógica/digital; tiene un diccionario en inglés, es decir, recibe texto en inglés debido a la fonética inglesa del diccionario. Aunque el costo del chip era muy bajo, aproximadamente de 7 dólares, lo descartamos de nuestro proyecto porque únicamente estaba disponible en inglés y chino mandarín.

La segunda opción era el sistema **RealSpeech TTS 2500** fabricado por la empresa ScanSoft, pero no obtuvimos respuesta cuando solicitamos información acerca del producto.

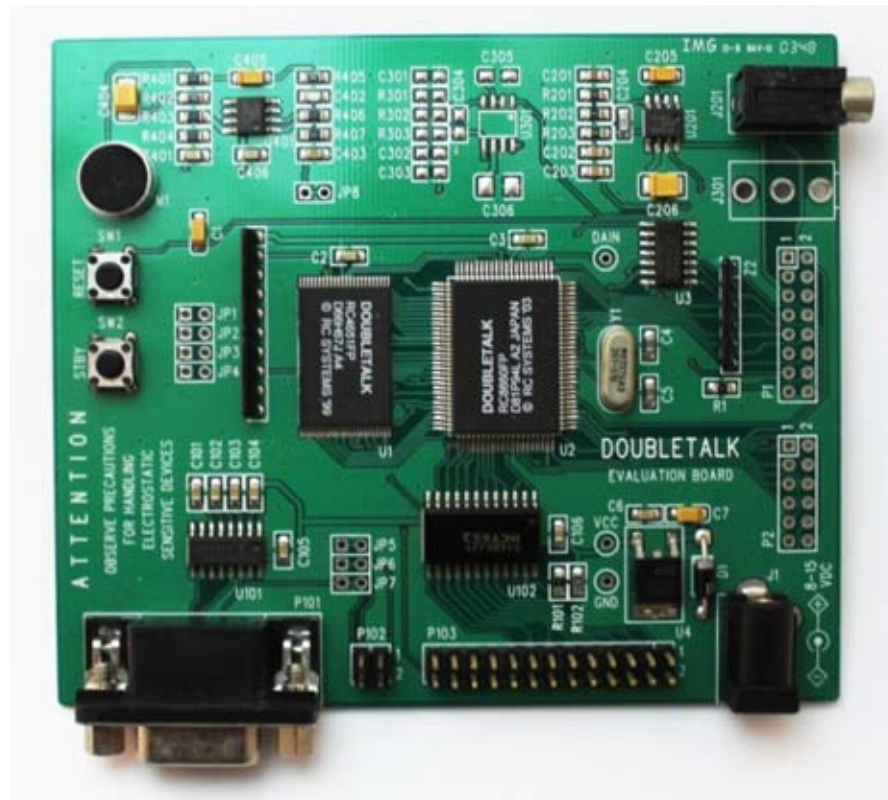
La tercera opción era el sintetizador **RC8660** de la empresa **RC Systems, Inc.** en Washington, Estados Unidos. Solicitamos información del producto y sobre las características que requeríamos para nuestro proyecto, obteniendo información de interés.

El chip viene montado en la tarjeta V8600A y puede “hablar” en español si se cargaba en éste un diccionario en español (reglas de letra-sonido). Este sintetizador utiliza fonemas en inglés americano implementados en el producto, y por lo tanto el acento de la voz es en inglés.

Analizando todas las opciones, finalmente seleccionamos el que más cumplía con todos nuestros requerimientos para nuestro proyecto, incluyendo un diccionario en español. Adquirimos el kit de evaluación del **sintetizador RC8660**, a un costo de 250 dólares. El kit incluía la tarjeta de evaluación **V8600A** del RC8660, un cable serie (DB-9), una bocina, un eliminador de 9 Volts y el software **RCStudio** utilizado para la comunicación entre una computadora y la tarjeta, y se utiliza también para descargar el diccionario en español (denominado diccionario de excepciones). La documentación del producto la obtuvimos del sitio: **www.rcsys.com**. En las *figuras 6.17 (a)* y *6.17 (b)* observamos imágenes del kit de evaluación del sintetizador y la tarjeta de aplicación.



(a) Kit de evaluación.



(b) Tarjeta de aplicación V8600A.

Figura 6.17 Kit de evaluación del sintetizador de voz RC8660

Posterior a la adquisición, investigamos y comprobamos su funcionamiento, realizando las pruebas necesarias que nos permitirían implementarlo en nuestro sistema. Con estas pruebas y con base al manual del chip RC8660 en la siguiente sección redactamos una síntesis de la operación del Sintetizador con las funciones básicas que requerimos para nuestra aplicación.

6.3.3.2 Operación y comandos de programación del sintetizador RC8660

Con el RC8660 es posible manipular los parámetros de la voz para obtener los tonos adecuados en la aplicación que se desarrolle. Todo atributo del sintetizador se define por medio de comandos que poseen una estructura como la siguiente:

<Carácter de comando> [<número>]<literal>

El carácter de comando por default es la tilde ~ en el software propio del sintetizador (RC8660 Studio) y en lenguaje ensamblador este carácter es sustituido por el número 01 en ASCII; el rango del número y la literal depende del comando a utilizar pues el valor numérico no siempre es utilizado. [5.2]

Por ejemplo, si queremos utilizar el modo texto del sintetizador, que se refiere a que el sintetizador procesara toda la información recibida como palabras y no carácter por carácter, es necesario enviar el siguiente comando: $\sim t$, que en valores hexadecimales equivale a enviar un 01 y un 74 (valor ASCII de la t) de manera consecutiva. [5.2]

Un caso en el que un comando requiere también de un valor numérico es el de la velocidad de pronunciación: $\sim 5s$ y si éste lo enviamos en base hexadecimal equivaldría a enviar 01, 39 y 73 consecutivamente. [5.2]

Es importante resaltar que el RC8660 no producirá sonido alguno ni ejecutará comandos hasta que reciba un CR (ASCII 13) o NULL (ASCII 00). Esto asegurará un completo análisis contextual de lo que pueda ser recibido por la entrada textual. Y si no es posible que el sintetizador reciba cualquiera de las dos señales se debe utilizar el comando de Time Out Delay. [5.2]

MODOS DE OPERACIÓN

Los modos de operación se utilizan para definir de qué tipo será la entrada del sintetizador, esto permite que el RC8660 produzca como salida una cadena enlazada, carácter por carácter o la pronunciación de fonemas (*figura 6.18*). [5.2]

Las literales para cada uno de los comandos son:

Texto: t, nt . $0 \leq n \leq 15$.

Carácter: c, nc . $0 \leq n \leq 31$.

Fonema: d .

nc y nt son utilizados como pausas o delays y el comando t además de habilitar el modo texto también deshabilita el diccionario de excepciones.



Figura 6.18 Modos de operación del sintetizador RC8660.

MARCADOR DE ÍNDICE (Index Marker)

A los Index Marker se les denomina “bookmarks” que pueden ser utilizados para no perder la pista de donde el RC8660 está “leyendo” un párrafo de texto. El comando está constituido por *nl*. El parámetro *n* es un número binario entre el 0 y 255 y no un número ASCII de una literal; así de este modo, hasta 256 marcadores pueden ser activados en cualquier momento. [5.2]

Cuando el RC8660 ha terminado de emitir la frase hasta un marcador, éste transmite el número del marcador por el pin TxD. Se transmite el marcador como un valor de un byte. [5.2]

DICCIONARIO DE EXCEPCIONES

En nuestra aplicación utilizamos un diccionario de excepciones en español proporcionado por el fabricante del chip, compilado y descargado con ayuda del software “RC8660 Studio”. [5.2]

El diccionario de excepciones hace posible la manera de interpretar una cadena de caracteres recibida por el RC8660. Es útil para corregir palabras mal pronunciadas, provocando la generación de tonos y/o reproduciendo sonidos pregrabados, e incluso para tener un idioma diferente al inglés. En algunos casos, un diccionario de excepciones puede incluso negar la necesidad de un pre-procesador de texto en aplicaciones que no pueden proveer de cadenas texto estándar. [5.2]

Los modos de texto a voz del RC8660 utilizan léxico y reglas de letra-sonido inglés para convertir el texto en voz. Las reglas de pronunciación determinan los fonemas, cada carácter será recibido con base en su posición relativa con cada palabra. El integrado Double Talk (texto a voz) analiza el texto aplicando estas reglas a cada palabra o carácter, dependiendo el modo de operación en uso. El diccionario de excepciones incrementa este proceso (incluso lo reemplaza). [5.2]

El diccionario de excepciones puede ser creado y editado con un procesador de texto o editor de texto que almacena documentos como archivos de texto estándar (ASCII). No obstante, el diccionario debe ser compilado al formato interno utilizado por el RC8660 antes de ser utilizado. [5.2]

La literal del comando para habilitar el diccionario es: **u**. [5.2]

STOP (CTRL+X)

El comando stop causa que el RC8660 detenga cualquier cosa que está haciendo y activa el buffer de entrada y comandos. Su código ASCII es el 24 (18 en base hexadecimal) y al ser escrito directamente en el RC8660 habilita a éste a reaccionar inmediatamente, incluso si el buffer de entrada está lleno. [5.2]

INTERFAZ SERIE DEL RC8660

El RC8660 tiene una interfaz serie asíncrona y un bus, ambos de 8 bits. Texto, comandos, datos generadores de tonos, datos de audio en tiempo real, etc., son transmitidos por ésta vía en uno de los puertos. Para máxima flexibilidad, es recomendable utilizar el puerto serie debido a que no todas las funciones son soportadas por la interfaz paralela. [5.2]

El puerto serie opera con 8 bits de datos (el bit menos significativo es el primero), 1 bit de stop, 1 bit de no paridad, y una velocidad estándar entre 300 y 115200 bps (bits por segundo). [5.2]

La tarjeta cuenta con una interfaz RS-232C para poder conectarla a la PC por el puerto serie. Es importante mencionar que el chip MAX232A no es necesario cuando los niveles lógicos de operación son compatibles con el RC8660 (0-5 V ó 0-3.3 V). Es por esto, que la tarjeta cuenta con dos interfaces para el puerto serie, una que utiliza la interfaz RS-232C y la otra no, sin olvidar que las líneas de transmisión (TxD), recepción (RxD), tierra (GND) y clear to send (CTS) son las mismas. [5.2]

CTS (Clear to send)

El puerto serie tiene un pin CTS# (Clear to send) que puede ser utilizado para controlar el flujo de los datos en serie hacia el RC8660. Sin embargo, no es necesario checar este pin antes de transmitir cada byte, todo dato es ruteado a través de un buffer de alta velocidad de 16 bytes dentro del RC8660 antes de ser almacenado en el buffer primario. CTS# podría ser verificado cada 8 bytes sin el riesgo de perder datos. [5.2]

El pin CTS# está en nivel bajo cuando el RC8660 está disponible para aceptar datos. Si el buffer de entrada se llena como resultado del último byte recibido, CTS# se pondrá en alto y permanecerá en alto hasta que el buffer esté disponible (ver *figura 6.19*). [5.2]

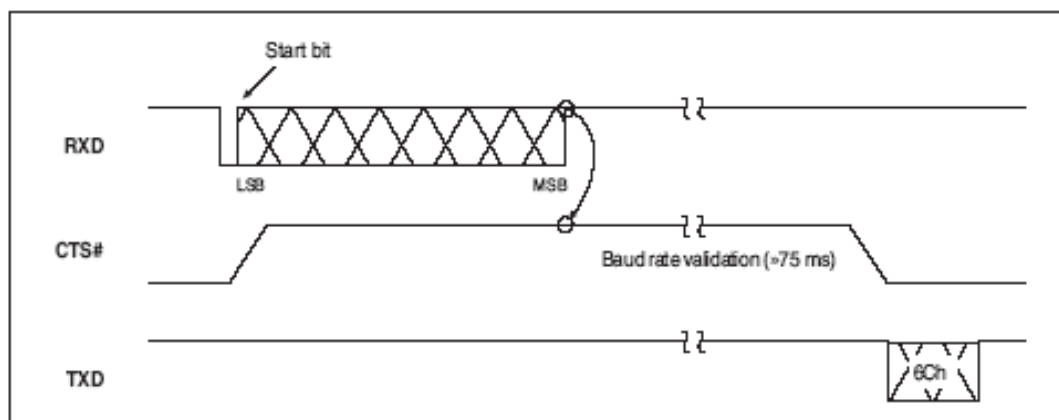


Figura 6.19 Detección de la Velocidad de Transmisión [5.2]

SELECCIÓN DE LA VELOCIDAD

La velocidad de transmisión puede ser programada utilizando cualquiera de estos tres métodos: por pines, auto detección y por comando. Por *pines* colocas el rango de velocidad de acuerdo a los niveles lógicos presentados en los pines BRS0-BRS3, como se muestra en la *Tabla 6.3*. *Auto detección* habilita en puerto serie la detección automática del rango de velocidad de los datos de entrada. Y utilizando *comandos* podemos cambiar la velocidad en cualquier momento, descartando los anteriores. [5.2]

BRS ₃	BRS ₂	BRS ₁	BRS ₀	Tasa de Baudios
0	0	0	0	300
0	0	0	1	600
0	0	1	0	1200
0	0	1	1	2400
0	1	0	0	4800
0	1	0	1	9600
0	1	1	0	19200
0	1	1	1	Auto detección
1	0	0	0	38400
1	0	0	1	57600
1	0	1	0	115200
Otros valores				Auto detección

Tabla 6.3 Configuración de la velocidad de transmisión [5.2]

6.3.4 Construcción física de la tarjeta de interfaz de acoplamiento.

Con base en el circuito armado después de la construcción del prototipo del guante con sistema sensorial, diseñamos el diagrama para el circuito impreso que utilizaríamos para conectar el guante con el microcontrolador; esta interfaz corresponde al bloque de **acondicionamiento de la señal** de nuestro diagrama de bloques general.

Tomando en cuenta que en nuestro sistema tenemos las siguientes tarjetas: del microcontrolador, del sintetizador y de la interfaz; y dado que en las tres estamos polarizando independientemente, decidimos colocar en una misma tarjeta los conectores para polarizar las tarjetas del microcontrolador y el sintetizador, y también en esa tarjeta tomaríamos las líneas para polarizar la parte correspondiente a la interfaz. En otras palabras, en una misma tarjeta implementaríamos la polarización de todas las tarjetas y la interfaz de acoplamiento del guante con el microcontrolador.

Considerando que la tarjeta del microcontrolador tiene la opción de polarizarla de dos maneras, la primera con una fuente de voltaje de 5 V y la segunda con un eliminador con rango de 9-12 V, debido a que tiene un regulador de voltaje LM7805; y observando

que la tarjeta del sintetizador también se alimenta por medio de un eliminador de 9 V, pensamos que con una sola fuente de alimentación podríamos polarizar todo el sistema. Para esto era necesario conocer las condiciones de polarización de las dos tarjetas y la interfaz.

El voltaje de alimentación de la interfaz es de 5 V, por lo tanto necesitaríamos un regulador de voltaje LM7805 para su polarización.

Ahora necesitábamos conocer el consumo de corriente de cada elemento para determinar las características que tendría que tener la fuente de alimentación de nuestro sistema. Para esto fue necesario medir de manera individual la corriente de cada tarjeta. Comenzamos primero con la tarjeta del microcontrolador, conectándole todos los dispositivos como: LEDs y memorias que utilizaríamos en el sistema; y utilizando un multímetro medimos la corriente que consumía, la lectura fue de 75 mA.

Seguimos con la tarjeta del sintetizador, conectándola a la computadora, enviando información continuamente y estableciendo el volumen del altavoz en el máximo nivel, midiendo de la misma manera la corriente, obteniendo el valor de 80 mA. Ahora faltaba conocer la corriente que consumía la interfaz. Para conocerla conectamos la tableta protoboard (donde teníamos implementada la interfaz), el guante y la tarjeta del microcontrolador, y de la misma manera medimos la corriente; teniendo en cuenta que la corriente es variable debido a los sensores del guante, tomamos una lectura de consumo máximo de corriente de 0.3 mA aproximadamente; esto se debe a la alta impedancia del divisor de voltaje. A continuación sumamos los valores de corriente para conocer la corriente total que consumiría nuestro sistema, obteniendo un valor de 158 mA.

Hicimos una prueba más, conectando todos los elementos para simular el comportamiento que tendría nuestro sistema operando: el guante, la interfaz, el microcontrolador y el sintetizador. Realizamos la medición de corriente del sistema encontrando aproximadamente el mismo valor que sumando las mediciones parciales. Por lo tanto bastaría polarizar el sistema con una fuente de 9-12 V que pudiera suministrar una corriente de al menos 158 mA. Con base en estas características utilizamos un eliminador de 12 V a 300 mA para polarizar todo el sistema.

La configuración del regulador de voltaje LM7805 que deberíamos utilizar en la interfaz de acuerdo a las hojas de especificaciones del fabricante para utilizarlo como regulador de una fuente de voltaje sugiere conectar dos capacitores; el primero entre la terminal de entrada y tierra, y el segundo entre la terminal de salida y tierra. El primero es importante para reducir las señales de ruido provenientes de la fuente de alimentación y debe estar lo más cercano posible al regulador y el segundo se coloca para estabilizar la salida del regulador de voltaje, así como para reducir el ruido en la interfaz. El primer capacitor es electrolítico con un valor de capacidad de $C1 = 47 \mu\text{F} - 16 \text{ V}$ y el segundo es un capacitor cerámico $C2 = 0.01 \mu\text{F}$.

El diagrama de bloques correspondiente a la interfaz se observa en la *figura 6.20*:

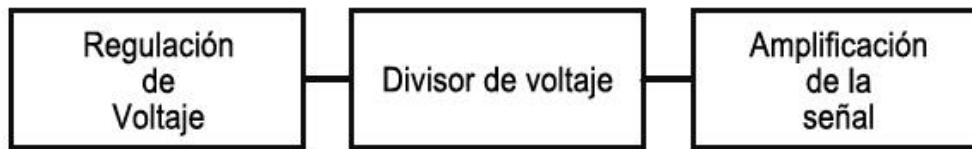


Figura 6.20 Diagrama de bloques de la interfaz de acoplamiento.

Los elementos que utilizamos para la realización del circuito impreso, así como la descripción de cada uno son:

- Conector Jack (1). Utilizado para conectar el eliminador a la tarjeta.
- Conector Molex de 2 pines (3). Para alimentar las otras tarjetas.
- Conector Molex de 10 pines (1). Para conectar el guante a la interfaz.
- Conector Molex de 5 pines (1). Para conectar la interfaz a la tarjeta del microcontrolador.
- Regulador de voltaje LM7805 (1). Descrito anteriormente.
- Capacitor C1: 47 μ F – 16 V (1). Descrito anteriormente.
- Capacitor C2: 0.01 μ F (1). Descrito anteriormente.
- Resistor de 33 k Ω (3). Para implementar el divisor de voltaje junto con los sensores de los dedos índice, medio y anular del guante.
- Resistor de 68 k Ω (3). Para implementar el divisor de voltaje junto con los sensores de los dedos pulgar y meñique del guante.
- Base para Circuito Integrado de Pin Torneado (5). Para los amplificadores Operacionales.
- Amplificador Operacional UA741 (5). Para acoplamiento de impedancias.

Antes de comenzar el circuito impreso, distribuimos cada uno de los elementos de manera que estuvieran en el lugar más conveniente para las conexiones con los demás elementos del sistema.

Para elaborar el diagrama de pistas utilizamos un programa llamado EAGLE que nos facilita la realización de circuitos impresos, este diagrama lo podemos ver en la *figura 6.21*.

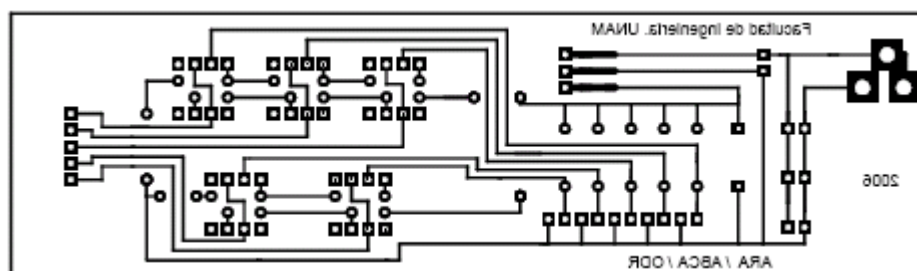
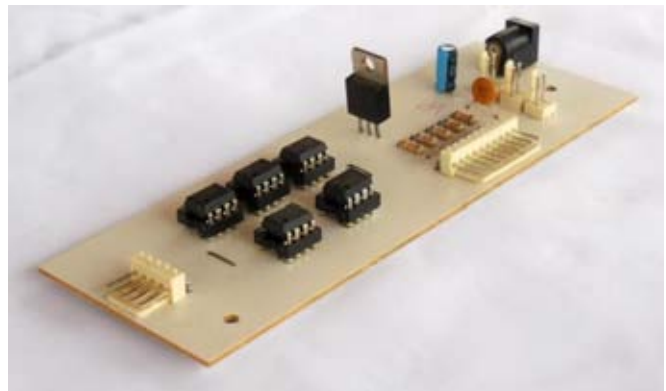
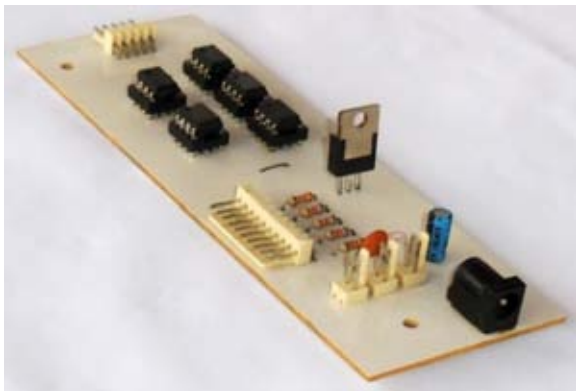
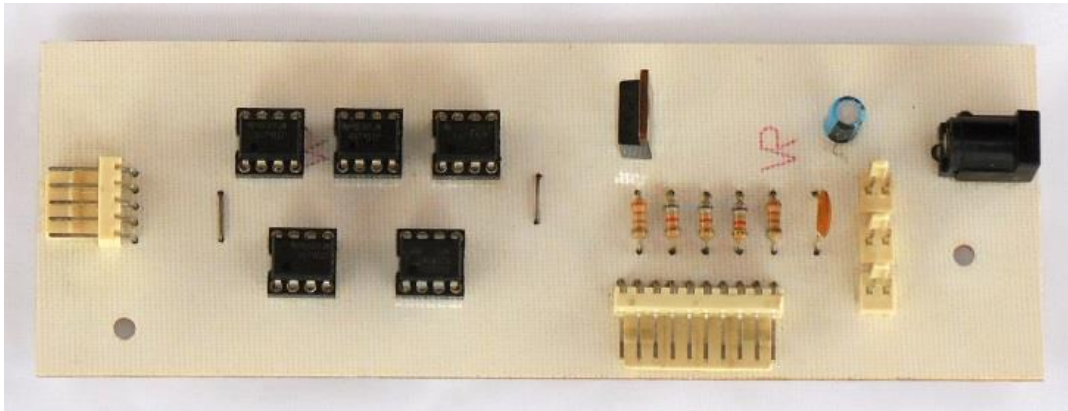


Figura 6.21 Diagrama de pistas para la tarjeta de interfaz.

El procedimiento que utilizamos para grabarlo en la placa de cobre es el siguiente:

- Imprimimos el diagrama de pistas una hoja especial (hoja satinada).
- Limpiamos perfectamente la placa.
- Colocamos la hoja encima de la placa, de tal forma que la cara de la hoja que contiene la tinta quedara frente a la placa.
- Utilizando con un trapo mojado y una plancha a temperatura media transferimos la tinta de la hoja a la placa de cobre.
- Remarcamos el diagrama con calcomanías especiales para circuito impreso.
- Sumergimos la placa en ácido férrico.
- Limpiamos los residuos de la placa.

Terminado el circuito impreso, lo perforamos, lo limpiamos y soldamos cada uno de los elementos en su posición, tomando en cuenta la polaridad de los elementos que así lo requerían. En la *figura 6.22* observamos la placa ya terminada.



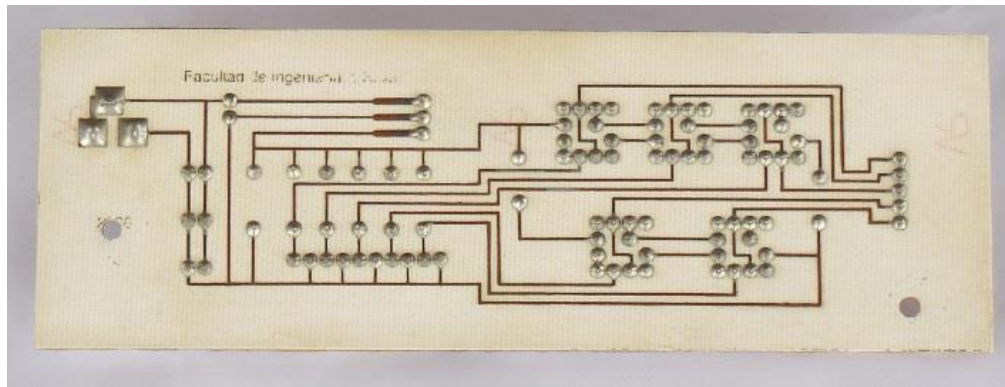


Figura 6.22 Tarjeta correspondiente a la interfaz de acoplamiento.

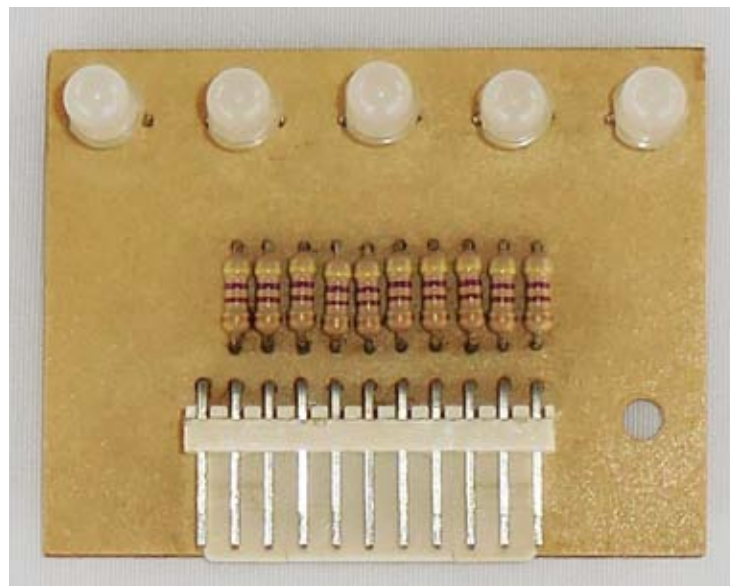
6.3.5 Construcción física de la tarjeta indicadora visual con LEDs para las posiciones de los dedos.

Con el objetivo de contar con un indicador visual para facilitar la realización de señas, construimos una tarjeta con cinco LEDs bicolores. Cada LED indica mediante un color la posición actual de cada dedo: abierta (verde), media (amarillo) y cerrada (rojo).

Los elementos que utilizamos para la realización del circuito impreso, así como la descripción de cada uno son:

- Conector Molex de 11 pines (1). Para conectar la tarjeta del microcontrolador a la tarjeta indicadora visual con LEDs.
- LEDs bicolores (5). Para visualizar la posición de cada uno de los 5 dedos del guante.
- Resistor de 470 Ω (10). Para polarizar los LEDs bicolores.

En la *figura 6.23* observamos la placa ya terminada.



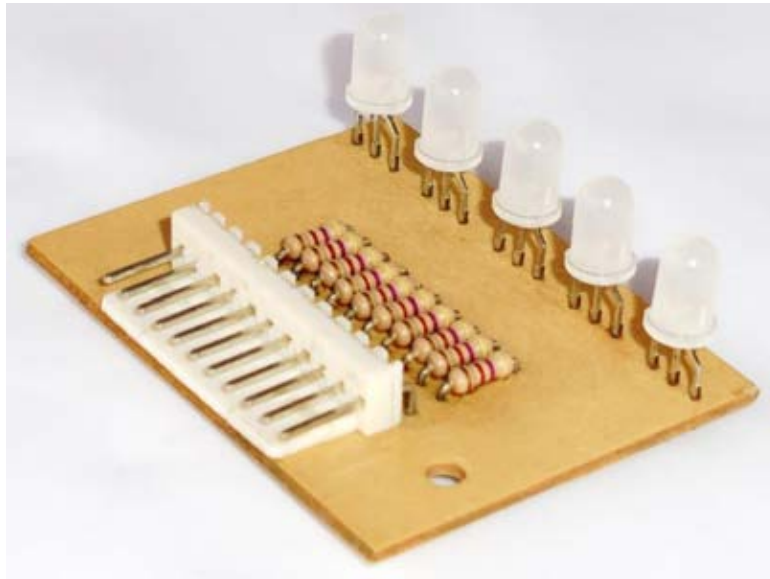


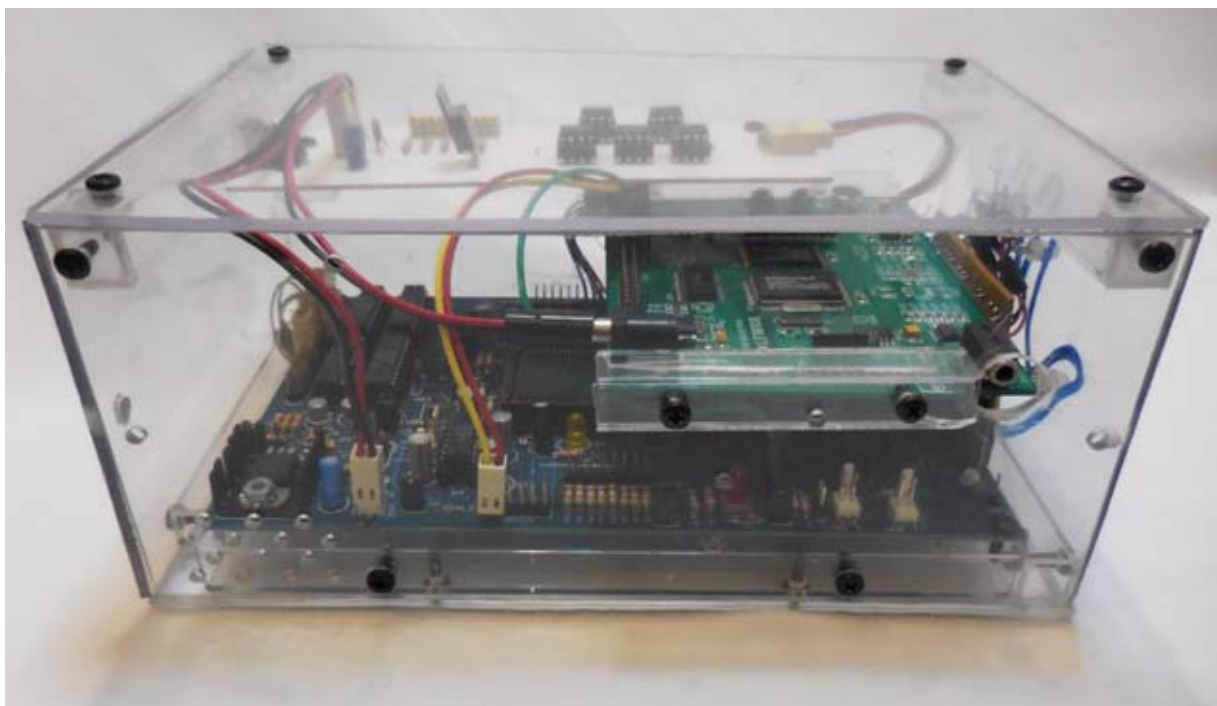
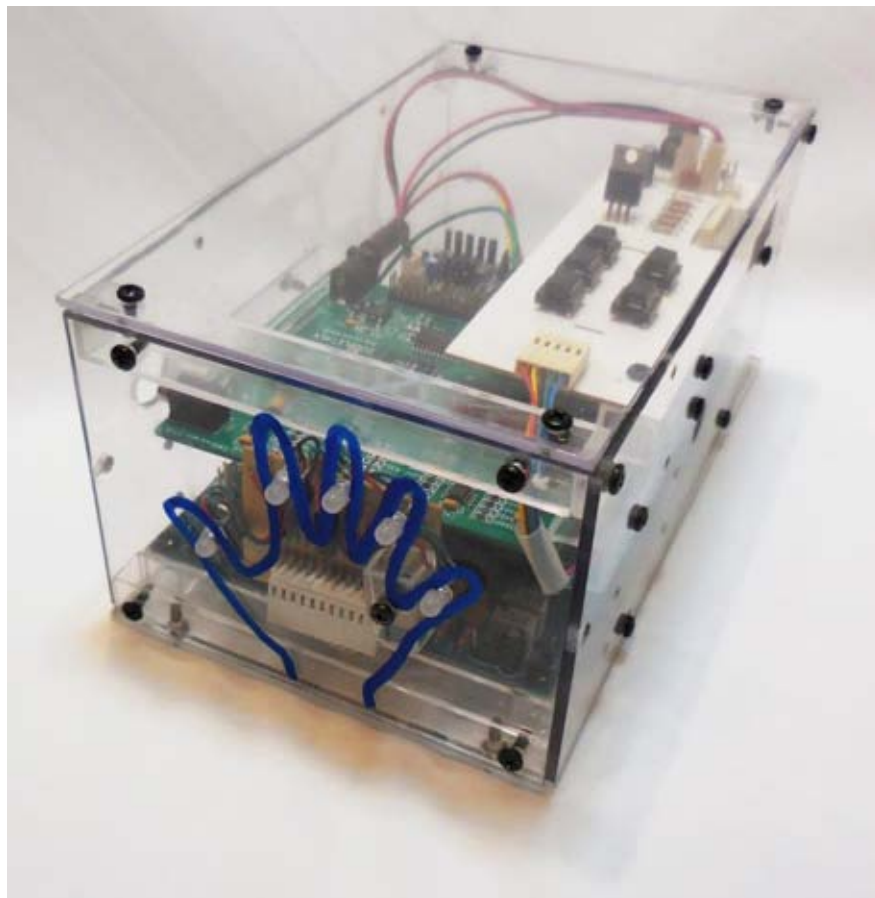
Figura 6.23 tarjeta indicadora visual con LEDs para las posiciones de los dedos.

6.3.6 Prototipo terminado del sistema traductor de señas dactilológicas a voz.

Terminadas las tarjetas, implementamos el sistema traductor de señas dactilológicas a voz. Para lograrlo era necesario integrar en un conjunto cada uno de los elementos que construimos y seleccionamos de manera que pudieran apreciarse, así como las conexiones eléctricas que se realizan entre ellas.

Construimos un gabinete de acrílico, lo cortamos y unimos con pegamento y tornillos para que tuviera una mejor durabilidad y portabilidad; también colocamos bases de acrílico para un mejor apoyo en la superficie donde se colocara el prototipo. A continuación realizamos las perforaciones necesarias hacia el exterior, en donde se conectarán el guante con sistema sensorial, la fuente de alimentación y el altavoz; en seguida montamos cada una de las tarjetas en su lugar correspondiente y se fijaron con tornillos al gabinete. Posteriormente se realizaron las conexiones utilizando cables y conectores Molex, y por último realizamos las conexiones hacia los elementos externos mencionados anteriormente, en la *figura 6.24* se muestran algunas imágenes correspondientes al prototipo terminado.

Finalmente realizamos las pruebas de funcionamiento definitivas, observando que todo el sistema funcionaba correctamente concluyendo así la construcción física de nuestro prototipo.



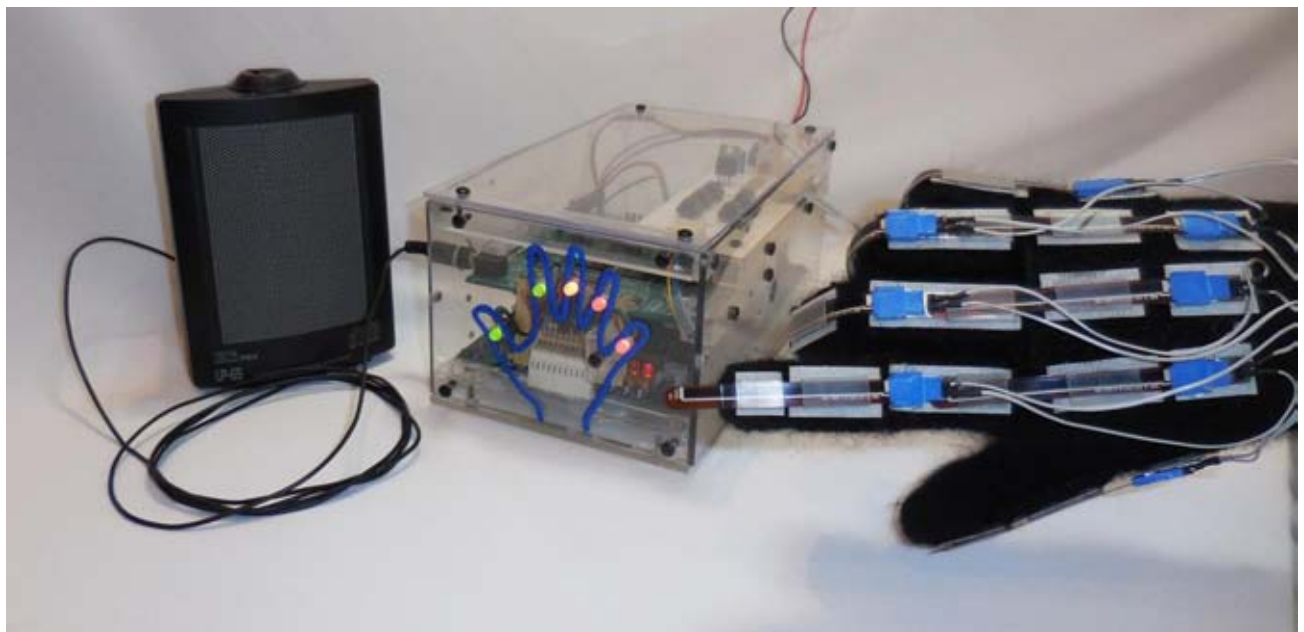


Figura 6.24 Prototipo terminado del sistema traductor de señas dactilológicas a voz.

CAPÍTULO 7

PRUEBAS Y RESULTADOS

7.1 PRUEBAS GUANTE SENSORIAL

7.1.1 Pruebas del primer prototipo

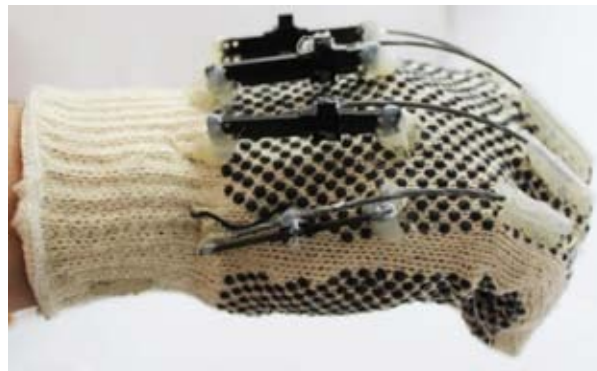


Figura 7.1 Primer prototipo del guante con sistema sensorial.

Durante las pruebas realizamos las conexiones eléctricas de los potenciómetros colocados en el guante en una tableta protoboard, utilizando la misma configuración de divisor de voltaje que usamos durante las pruebas con los potenciómetros. Tomamos lecturas de voltaje en relación al desplazamiento del cursor del potenciómetro en cada uno de los dedos, comenzando de izquierda a derecha con el dedo pulgar hasta el meñique. Registramos lecturas para tres posiciones diferentes de los dedos: **abierta**, **media** y **cerrada**.

Analizando las lecturas obtenidas, encontramos que había una pequeña, pero suficiente variación de voltaje entre las tres posiciones de cada uno de los dedos; también observamos variaciones entre diferentes tomas de lectura para el mismo dedo. Por lo tanto este prototipo sí nos serviría para nuestro propósito.

Posteriormente realizamos algunas señas del alfabeto dactilológico para comprobar la ergonomía del guante durante el uso, encontrando algunos problemas. Al momento de realizar movimientos (abrir y cerrar los dedos) y formar las señas, en el guante se formaban pliegues con la misma tela que obstruían el movimiento libre de los cables de acero que desplazan el cursor de los potenciómetros y como consecuencia no se podían

formar algunas señas; también observamos que los valores de voltaje se modificaban cuando formábamos señas en donde los dedos se cruzan, debido a que los dedos provocaban desplazamientos adicionales en otros dedos. Estas alteraciones ocasionarían errores en el sistema.

En general observamos que el sistema sensorial que estaba montado sobre el guante y los cables de acero con los que funciona, dificultaban los movimientos de la mano y la formación de señas.

Con la detección de estos problemas descartamos este primer prototipo, procediendo a reconstruirlo por completo.

7.1.2 Pruebas del segundo prototipo



Figura 7.2 Segundo prototipo del guante con sistema sensorial.

Para la realización de pruebas del segundo prototipo, con base en las interfaces que existen para conectar los sensores de flexión y teniendo en cuenta la señal de voltaje requerida, utilizamos una configuración divisor de voltaje que sirve para realizar una conversión simple de flexión a voltaje. El circuito de esta configuración se muestra en la *figura 7.3*.

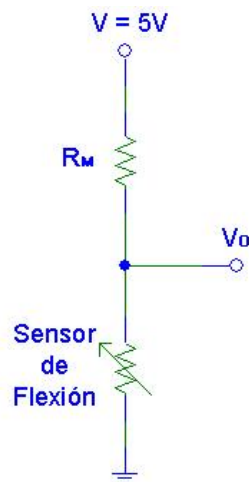


Figura 7.3 Conversor simple de flexión a voltaje.

En esta configuración, el voltaje de salida V_o se incrementa conforme aumenta la flexión. Para nuestro sistema fue muy importante esta configuración ya que la señal de salida era el voltaje. Ahora necesitábamos conocer el valor de R_M ; entonces realizamos mediciones para conocer el rango de valores de resistencia con los que trabaja el sensor. Comenzamos colocando el sensor en su posición natural, es decir, totalmente recto, posteriormente realizamos movimientos de curvatura sobre su eje de movimiento progresivamente para conseguir los valores de operación. En la siguiente *figura 7.4 (a) y (b)* se observan algunas posiciones del sensor y sus valores correspondientes de resistencia proporcionadas por el fabricante.

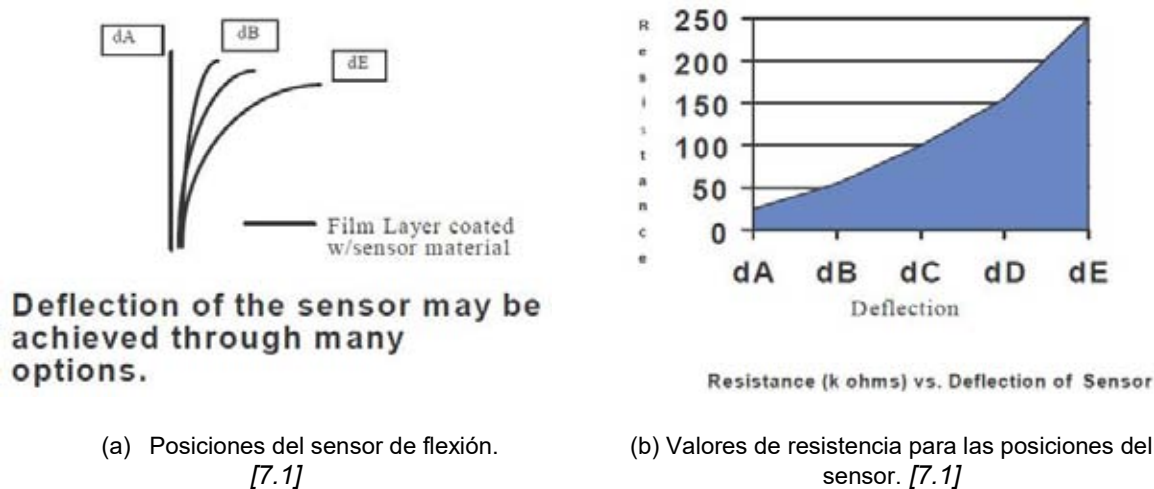


Figura 7.4 Posiciones del sensor de flexión y sus valores de resistencia proporcionados por el fabricante FLEXPOINT.

.Observando la gráfica anterior y tomando en cuenta los movimientos posibles que vamos a realizar con el dedo, se establece que el rango de valores del sensor de flexión para nuestra aplicación es de $14 \text{ k}\Omega$ - $73 \text{ k}\Omega$; estos valores corresponden a los movimientos de curvatura emulando el movimiento del dedo lo mejor posible. Con esta información continuaríamos con las pruebas de voltaje utilizando la configuración de la *figura 7.3*, conectando en serie resistencias de valores aproximados a los valores del rango del sensor.

De la misma manera que en el primer prototipo realizamos algunas señas del alfabeto dactilológico y movimientos de la mano en general, encontrando nuevamente un problema relacionado con el material, el guante de piel comenzó a deformarse por el uso. Por el movimiento, los nudillos de los dedos dejaron marcas en el guante y éste comenzó a sentirse cada vez más holgado y menos ajustado a la mano por lo que los pliegues también se hicieron evidentes. Estas deformaciones comenzaron a obstruir el movimiento de los sensores; este problema nos llevó a descartar este prototipo y buscar una nueva opción, un guante que no nos causara problemas después de utilizarlo en repetidas ocasiones como lo sería una vez puesto en uso.

Tomando como base este prototipo y considerando un buen funcionamiento en general, solo tendríamos que cambiar el guante e implementar los sensores de forma similar, pero realizando algunas mejoras.

7.1.3 Pruebas del tercer prototipo (Prototipo Final)



Figura 7.5 Tercer prototipo del guante con sistema sensorial.

De la misma forma que en los otros prototipos, realizamos pruebas de movimientos basándonos en la Lengua de Señas Mexicana, para garantizar que el guante fuera ergonómico y que no existieran deformaciones en los sensores o en el guante durante su uso.

En general durante las pruebas con el guante con sistema sensorial, encontramos un buen funcionamiento para los alcances de nuestro proyecto.

A continuación nos enfocamos a realizar las pruebas correspondientes a la caracterización del sensor de flexión para conocer el valor de la resistencia R_M , basándonos en el circuito de la *figura 7.3*. Conectamos una fuente de 5 V entre las terminales de V y tierra del circuito, y con ayuda de un multímetro digital hicimos unas gráficas del voltaje en función de la flexión, tomando lecturas de voltaje entre las terminales V_o y tierra. De acuerdo a esa configuración, conectamos un sensor de flexión ($14\text{ k}\Omega - 73\text{ k}\Omega$) en serie con diferentes resistencias de valores aproximados a los valores del rango del sensor como se puede observar en la *figura 7.6*.

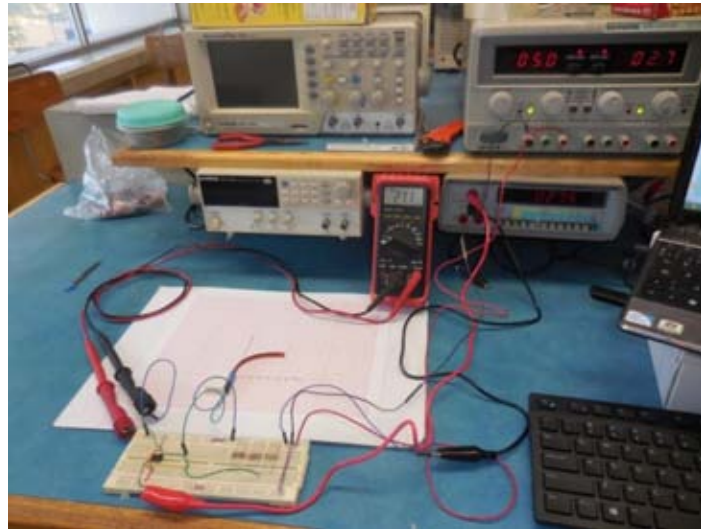
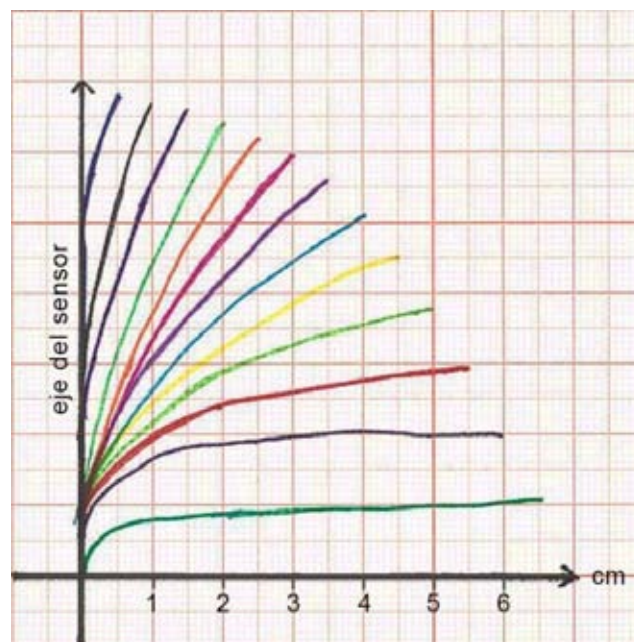
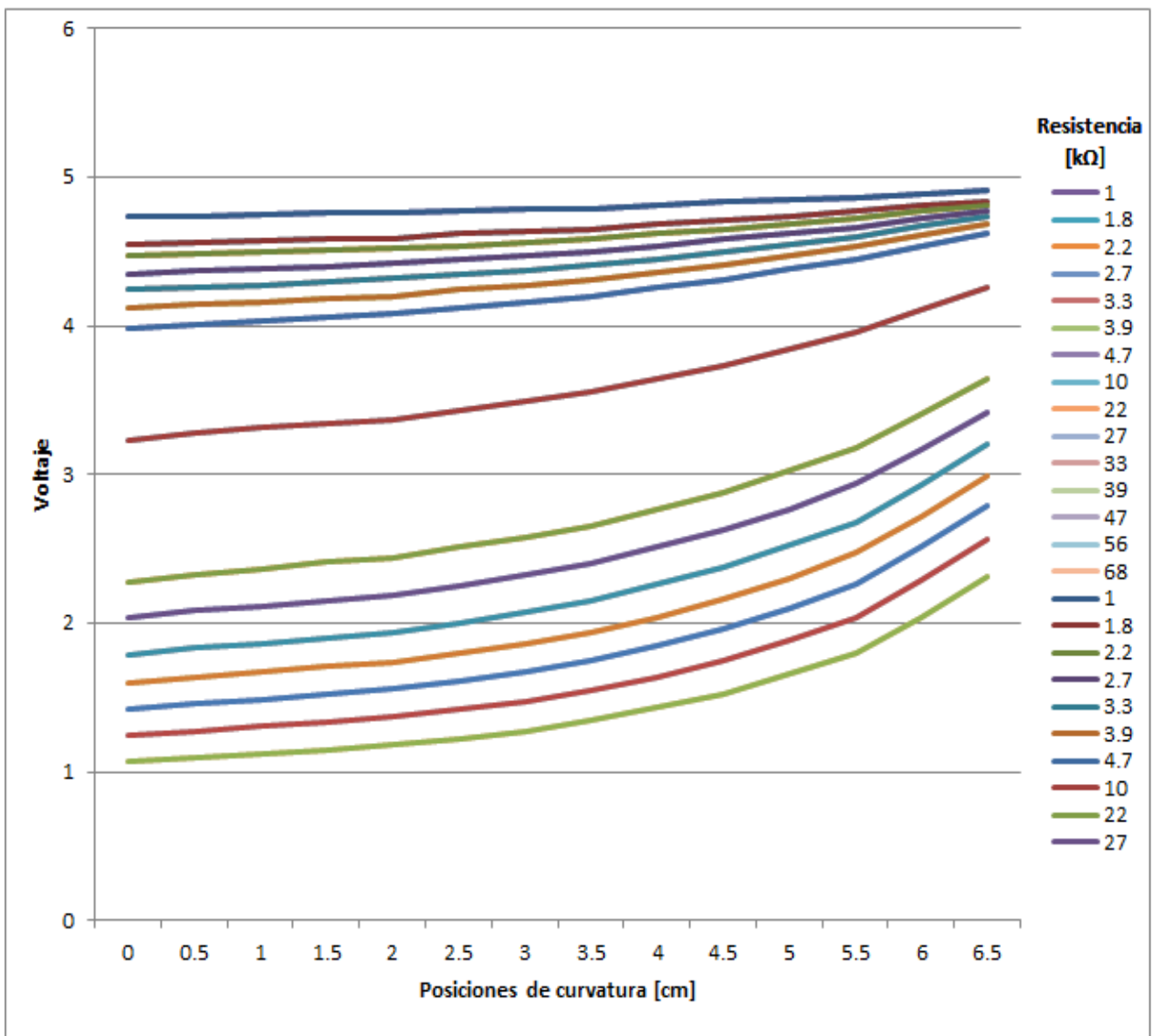


Figura 7.6 Pruebas con el sensor para encontrar los valores de la resistencia R_M .

Para las pruebas implementamos las conexiones en una tableta protoboard y haciendo una simulación de la curvatura que podría formar el sensor montado en el sistema, realizamos gradualmente movimientos de curvatura con el sensor para diferentes medidas desde 0, 0.5, 1, y así sucesivamente hasta 6.5 centímetros. Los valores de resistencias que utilizamos fueron: 1, 1.8, 2.2, 2.7, 3.3, 3.9, 4.7, 10, 22, 27, 33, 39, 47, 56 y 68 k Ω ; obteniendo una curva distinta para cada una. Los diferentes movimientos de curvatura que realizamos y las curvas correspondientes a la caracterización del sensor de flexión se muestran en las *figuras 7.7 (a) y (b)*.



(a) Movimientos de curvatura con el sensor de flexión.



(b) Curvas correspondientes a la caracterización del sensor de flexión.

Figura 7.7 Gráficas obtenidas durante las pruebas para la obtención de la resistencia R_M .

En nuestro sistema dos dedos utilizan un sensor de flexión y tres dedos utilizan dos, entonces la conexión de estos dos sensores la hicimos en serie para obtener un valor de resistencia mayor y por consiguiente hicimos otras gráficas para este caso. Estas gráficas a diferencia de las anteriores, las realizamos con los sensores montados sobre los dedos del guante, obteniendo un comportamiento similar para cada valor de resistencia probada. Los valores de voltaje registrados para la obtención de las curvas de la *figura 7.7* se muestran en el ANEXO A.

Terminada la caracterización de los sensores y con base en las curvas obtenidas, elegimos las resistencias que colocaríamos en serie con los sensores; decidimos que para un sensor la resistencia R_M debería ser de **33 k Ω** y para dos sensores utilizaríamos la de **68 k Ω** , debido a que con estos valores de resistencia se obtuvo una mayor variación en el voltaje respecto a la flexión, es decir una resolución adecuada para los movimientos que necesitábamos. Con los valores de resistencia seleccionados quedó definida la interfaz para conectar el guante con sistema sensorial y el microcontrolador (ver *figura 7.8*).

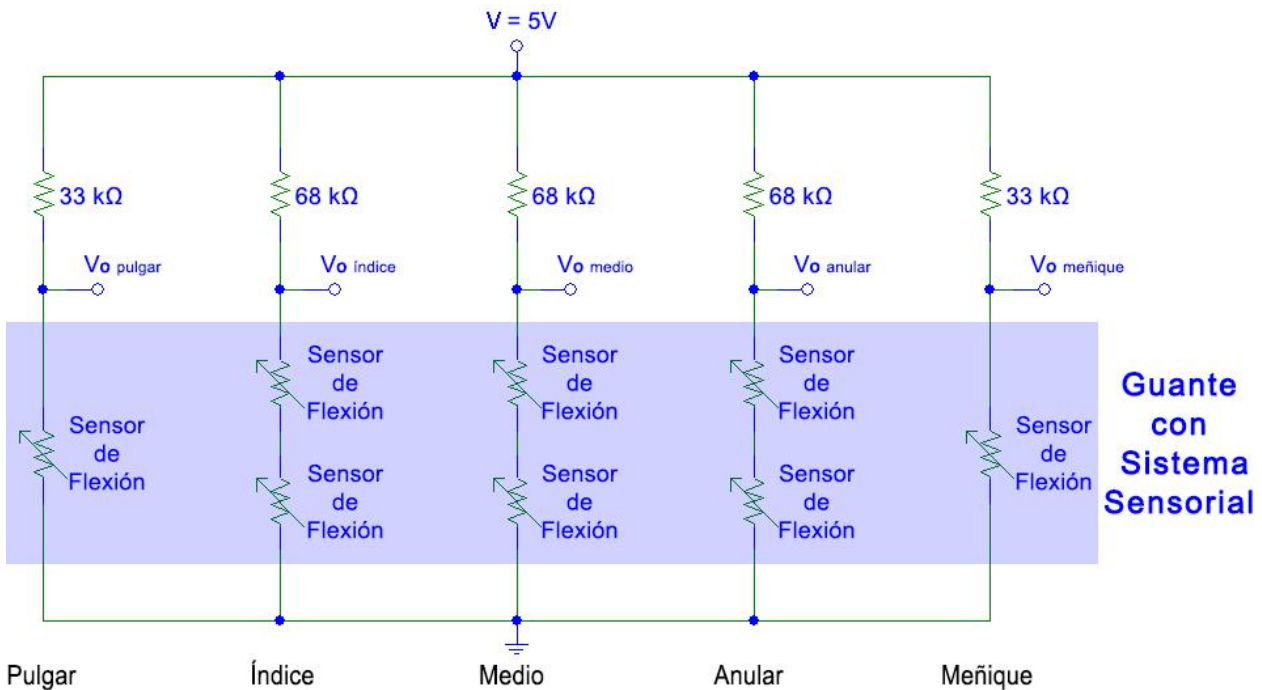


Figura 7.8 Interfaz para conectar el guante con sistema sensorial y el microcontrolador.

Utilizando la interfaz, ahora tendríamos que integrar el guante y el microcontrolador. Realizamos algunas pruebas, medimos algunas de las salidas V_o y observamos que el voltaje V_o no se estabilizaba por completo, existían variaciones al momento de tomar las lecturas cuando realizábamos movimientos de la mano, especialmente para los dedos que tienen dos sensores conectados en serie.

La explicación para este problema es la siguiente. Analizando la configuración de la *figura 7.8*, sabemos que el voltaje de salida V_o se incrementa con respecto a la flexión, y debido a que los valores de las resistencias seleccionadas y los sensores de flexión en conjunto son muy altos (los valores oscilan entre 50 k Ω y 214 k Ω dependiendo de la flexión), la alta impedancia que se presenta con el divisor genera errores en la toma de lecturas de la señal. Entonces es necesario compensar la corriente para estabilizar la señal. En este caso, es necesario implementar una etapa de amplificación para evitar este comportamiento.

7.2 PRUEBAS DEL GUANTE SENSORIAL CON LA ETAPA DE AMPLIFICACIÓN

Para integrar la interfaz con la etapa de amplificación, armamos el circuito mostrado en la *figura 7.9* en una tableta protoboard, lo alimentamos con una fuente de 5 V a las terminales de V y a su respectiva tierra, y con ayuda de un multímetro digital hicimos las mismas gráficas del voltaje en función de la flexión, tomando lecturas de voltaje entre las terminales V_o tierra, como lo habíamos hecho anteriormente.

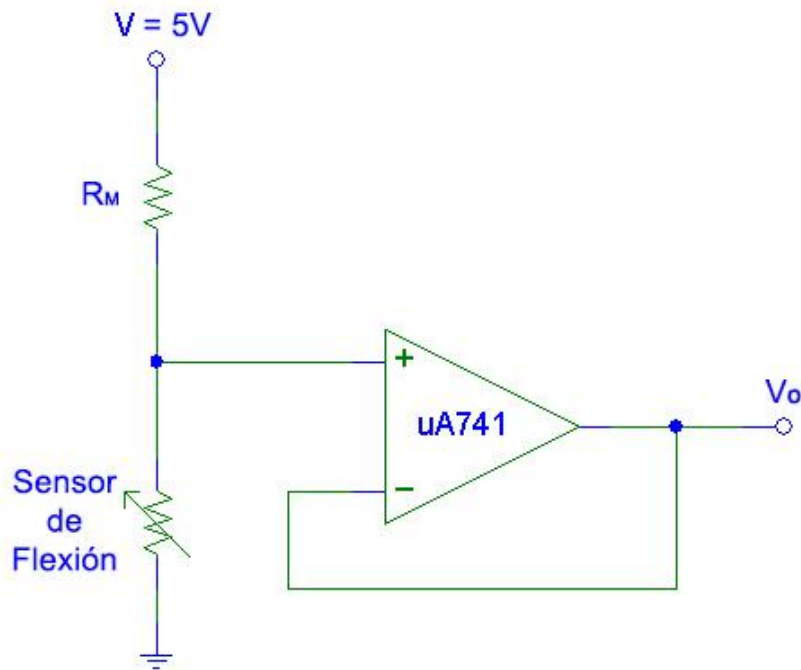


Figura 7.9 Conversor simple de flexión a voltaje con etapa de amplificación.

El circuito armado incluye un amplificador operacional (**UA741**), y se tomó como base una configuración tipo **seguidor de voltaje**, también conocido como **seguidor unitario** o **amplificador de acoplamiento** (ver *figura 7.10*). [7.1]

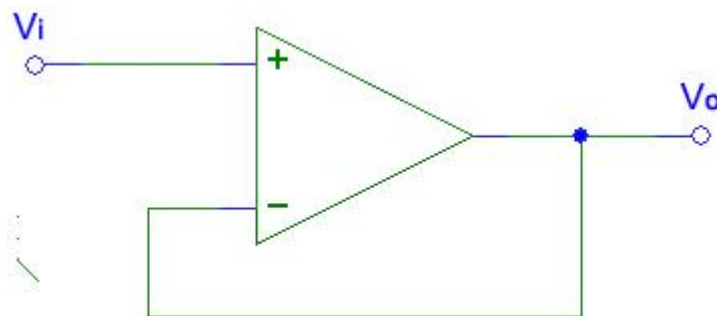


Figura 7.10 Configuración tipo seguidor de voltaje. [7.2]

El nombre de este circuito se debe a que el voltaje V_o sigue al voltaje V_i . A partir del concepto de amplificador ideal, el seguidor de voltaje esencialmente tiene una impedancia de entrada infinita y una impedancia de salida cero [7.2]. Aunque aparenta ser simple, el circuito es muy próximo a uno ideal porque la retroalimentación negativa es máxima. Como se puede observar, la resistencia de retroalimentación es cero. Por lo tanto, todo el voltaje de salida se retroalimenta hacia la entrada inversora del amplificador operacional. Debido a la existencia del cortocircuito virtual entre las entradas del amplificador operacional, el voltaje de salida es de magnitud igual al voltaje de entrada [7.3]:

$$V_o = V_i$$

lo que significa que la ganancia de voltaje en lazo cerrado (close loop CL) es [7.2]:

$$A_{CL} = 1 \text{ [7.3]}$$

Por lo tanto, es un circuito seguidor perfecto porque produce un voltaje de salida que es exactamente igual al voltaje de entrada (o suficientemente parecidos como para satisfacer casi cualquier aplicación) [7.3], sin inversión de polaridad o fase. [7.1]

Además la retroalimentación negativa máxima produce una impedancia de entrada en lazo cerrado que es mucho mayor que la impedancia de entrada en lazo abierto (2 M Ω para un 741C). También, la retroalimentación negativa máxima produce una impedancia de salida en lazo cerrado que es mucho menor que la impedancia de salida en lazo abierto (75 Ω para un 741C). Por tanto, se obtiene un método casi perfecto para convertir una fuente de alta impedancia en una fuente de baja impedancia. [7.3]

El seguidor de voltaje permite que fuentes de alta impedancia alimenten cargas de baja impedancia sin pérdidas de voltaje, es decir, puede usarse para aislar, o acoplar, un circuito a otro. (Los voltajes en los dos pares de terminales son los mismos, pero no puede fluir corriente de un par a otro). [7.3]

En nuestra implementación, la terminal (+) del amplificador operacional conocida como **entrada no inversora** es conectada con la señal de salida del divisor de voltaje tomada entre el sensor de flexión y la resistencia R_M , y al tener la retroalimentación con la **entrada inversora** (-) y la salida del amplificador se eliminan los **efectos de carga** (acción conocida como **regulación**) reflejados en la pérdida de voltaje a medida que aumenta la carga o demanda de corriente. Por lo tanto, la salida se toma de la retroalimentación y de esta manera se obtiene una corriente en un rango de miliamperios, mucho mayor que la que existe en la entrada.

La siguiente ecuación nos permite obtener el comportamiento a la salida del amplificador operacional para un conversor simple de flexión a voltaje como el de la *figura 6.19*, para esa configuración el voltaje de salida V_o incrementa con la flexión.

$$V_o = \frac{V}{1 + \frac{R_M}{R_{SF}}}$$

Tomando en cuenta el rango de valores del sensor de flexión y que el valor de resistencia registrado cuando existía mayor variación de flexión está alrededor de $36\text{ k}\Omega$, comentamos que durante las pruebas para encontrar el valor de R_M utilizamos valores de resistencias cercanos a ese valor, esto para poder hacer una comparación, verificar los resultados y deducir conclusiones. Para ello graficamos cada conjunto de valores resultado de varios puntos de flexión aplicados sobre el sensor, en conjunto con cada una de las resistencias.

Al comprobar que los valores de voltaje V_o obtenidos para los mismos valores de resistencias R_M coincidían con los obtenidos utilizando el circuito de la *figura 7.8*, y al observar una estabilización de la señal de voltaje, finalmente quedó definida la interfaz que utilizaríamos para conectar el guante con sistema sensorial y el microcontrolador. El Circuito correspondiente a la interfaz la podemos visualizar en la *figura 7.11*, dicha interfaz corresponde al bloque **Acondicionamiento de Señales** indicado en la *figura 6.1*.

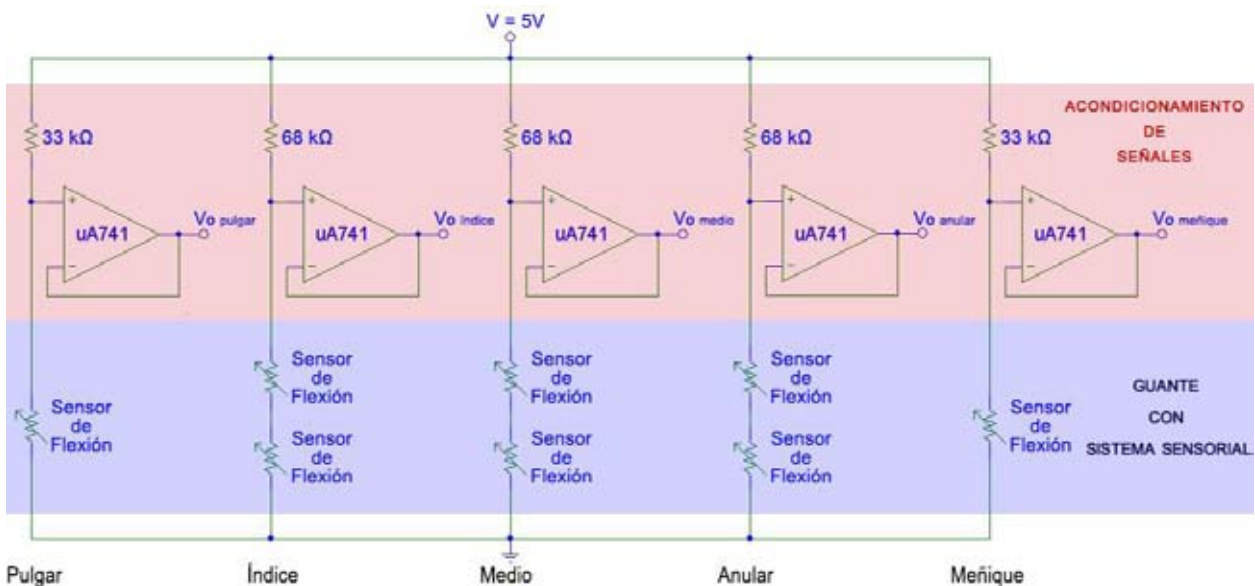


Figura 7.11 Interfaz con etapa de amplificación para conectar el guante con sistema sensorial.

Con el guante con sistema sensorial terminado y la interfaz de acoplamiento con etapa de amplificación (Acondicionamiento de Señales), continuamos con la integración con el microcontrolador.

7.3 PRUEBAS REALIZADAS ENTRE EL GUANTE SENSORIAL Y CONVERTIDOR A/D DEL MICROCONTROLADOR

Algunas características del convertidor A/D del microcontrolador que tomamos en cuenta para utilizarlo adecuadamente en nuestro proyecto son: que es un convertidor de 8 bits, consta de 8 canales de conversión, tiene entradas multiplexadas, es un convertidor de tipo aproximaciones sucesivas; cuenta con un registro de control y estado llamado el ADCTL, 4 registros de resultado de conversión ADR1-ADR4 y un registro de configuración llamado OPTION. [W6.1]

Para utilizar el convertidor A/D realizamos las conexiones a través del conector que corresponde al PUERTO E de la tarjeta FACIL11_B, configurando los registros necesarios para habilitarlo a través de programación. Por otra parte establecimos los límites de voltaje colocando los *jumpers* o puentes correspondientes en la tarjeta, para el límite inferior 0 V (V_{RL}) respectivamente y para el límite superior 5 V (V_{RH}), dentro de los cuales va a operar el convertidor A/D.

En la misma tarjeta modificamos la configuración de los jumpers: MODA=0 y MODB=0 para seleccionar el modo de operación **bootstrap**. Por otra parte conseguimos el programa llamado **IASM11** para editar los programas en lenguaje ensamblador con extensión “asm” y compilarlos para generar archivos “s19”, que pueden descargarse y ejecutarse en la memoria interna del microcontrolador (RAM o EEPROM). Para descargar los archivos a la memoria interna del microcontrolador configuramos los jumpers y utilizamos otro programa llamado **PCBUG11**.

Es importante mencionar que los programas IASM11 y PCBUG11 están diseñados para ejecutarse en sistemas operativos anteriores a Windows XP y en hardware con velocidades muy inferiores a las actuales (por debajo de 300 MHz); con estas condiciones recurrimos a trabajar en una computadora con esas características para desarrollar nuestra aplicación.

Verificamos el funcionamiento de los programas y la comunicación entre el PCBUG11 y el microcontrolador. Posteriormente hicimos el primer programa donde utilizamos el convertidor A/D y un sensor de flexión, conectándolo de acuerdo a la interfaz con su etapa de amplificación (ver *figura 7.11*). El programa encendía los 8 LEDs que la tarjeta del microcontrolador tiene colocados a la salida del PUERTO B; encendía los 4 LEDs de la parte baja cuando el voltaje de salida en el circuito V_o era menor a 2.7 V y los otros 4 LEDs de la parte alta cuando el voltaje era mayor a ese valor.

En seguida realizamos pruebas con otro recurso interno del microcontrolador, el **Serial Communications Interface SCI**; que es un puerto de comunicación serie asíncrona, por el cual estableceríamos la comunicación entre el microcontrolador y el sintetizador de voz. El SCI cuenta con una unidad de transmisión y una unidad de recepción independientes una de la otra, es decir, puede transmitir y recibir datos simultáneamente (modo **full-duplex**); dispone de 5 registros: el registro BAUD para configurar la velocidad de transmisión, los registros SCCR1 y SCCR2 para su configuración, el registro SCSR para verificar el estado de la transmisión y por último el registro de datos SCDR para enviar y recibir de datos. [W6.1]

La tarjeta del microcontrolador FACIL11_B está diseñada para que el microcontrolador se comunique con otros dispositivos a través del SCI con dos estándares de comunicación: el RS-232 y el TTL. En esta etapa de nuestra aplicación podíamos utilizar cualquiera de los dos en la tarjeta, pero aún no conocíamos con cuál trabajaría el sintetizador, cabe señalar que la tarjeta se comunica con la computadora a través del RS-232, debido a que es el estándar que se maneja en las computadoras para transmisión serie.

Con las pruebas concluidas ya teníamos las condiciones para establecer la comunicación de la tarjeta microcontrolador con otro dispositivo, el sintetizador de voz.

7.4 PRUEBAS DE OPERACIÓN CON EL KIT DE EVALUACIÓN DEL SINTETIZADOR RC8660

Para realizar pruebas con el kit de evaluación del sintetizador RC8660 seguimos las indicaciones del manual “DoubleTalk RC8660. Evaluation kit”, instalamos en una computadora el software RCStudio para trabajar con la tarjeta entrenadora a través del puerto serie, conectamos la bocina de 6 Ω a la tarjeta del RC8660 y la alimentamos con 9 V por medio de un eliminador.

Cuando ejecutamos el software para comunicarnos con el sintetizador, se activó una ventana conformada por tres secciones: el menú principal, la ventana en donde se teclea el texto a reproducir y la sección en donde se configuran los parámetros de la voz, tanto el tipo como la entonación que se requieren (*figura 7.12*).

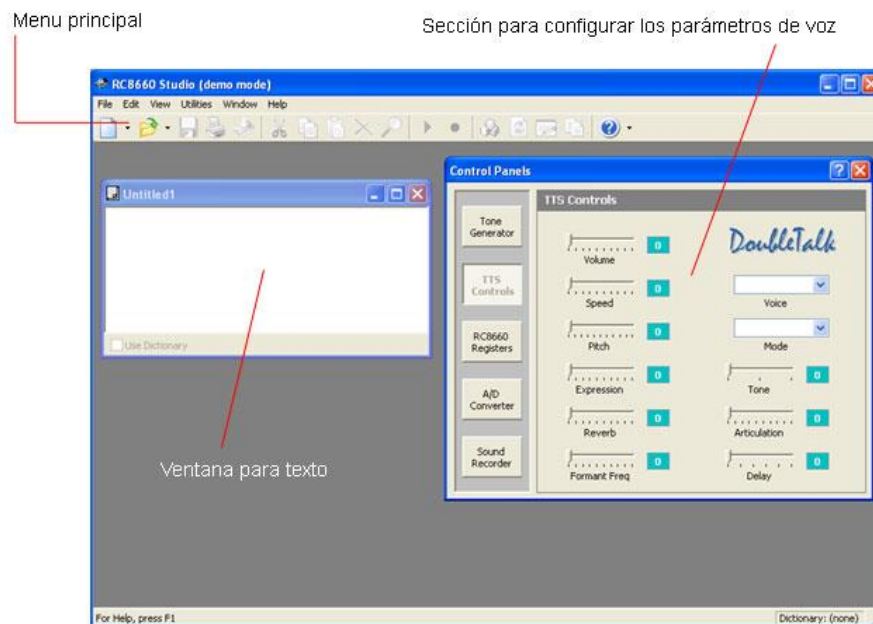


Figura 7.12 Interfaz gráfica del software RCStudio.

Tecleamos texto y al reproducirlo escuchamos que las frases en español tenían una traducción fonética inglesa. Investigamos en el manual del software todo lo relacionado con el diccionario en español y encontramos que éste es llamado “**diccionario de excepciones**” debido a que está constituido por un conjunto de reglas fonéticas que determinan como debe pronunciar las sílabas en español el sintetizador, por ejemplo estas son dos reglas del diccionario:

$$z = |s| |ei| |t| |a|$$

$$10000 = |d| |y| |ei| |s| |m| |i| |||$$

Cargamos el diccionario de excepciones en el sintetizador a través del software para comenzar a utilizarlo y, de acuerdo a las especificaciones del manual, para utilizar al diccionario es necesario habilitarlo o deshabilitarlo a través del comando `~u`. Probamos nuevamente tecleando texto en español, habilitando el diccionario con el comando `~u` y al reproducirlo obtuvimos la pronunciación deseada en español.

Por otra parte, dado que la tarjeta del sintetizador no tiene un indicador visual para saber si enciende o no, grabamos un saludo de bienvenida (greeting message) que permite escuchar una frase cuando alimentamos la tarjeta o cada vez que oprimimos el botón RESET.

El greeting message que grabamos en esa prueba contiene este código:

Renglón	Comandos / Cadenas	Descripción
1	<code>~3s~6v~80p~7o</code>	Configuración de parámetros de la voz
2	<code>~t~u</code>	Habilita el diccionario de excepciones y el modo texto
3	<code>~3*</code>	Tonos telefónicos
4	Hola amigos	Texto a reproducir en modo texto
5	<code>~c~u~acr</code>	Habilita el diccionario de excepciones y el modo carácter; y los caracteres a reproducir: a, c y r
6	<code>~2*</code>	Tonos telefónicos

Tabla 7.1 Comandos y cadenas utilizados para grabar el greeting message en el sintetizador de voz

En este mensaje, el carácter `~` indica que es un comando el texto que lo precede. El primer renglón contiene la configuración de algunos parámetros de la voz, en el segundo renglón se habilita el diccionario de excepciones y el modo texto que es lo contrario del modo carácter con el que se puede deletrear una palabra como se muestra en el quinto renglón. Y los renglones 3 y 6 reproducen tonos telefónicos. Es importante mencionar que las minúsculas y mayúsculas se utilizan indistintamente.

Para conocer cómo se establecía la comunicación entre la tarjeta del sintetizador y la computadora, utilizamos el software **Advanced Serial Port Monitor** para escanear el puerto serie durante la comunicación entre estos. Esta acción fue necesaria debido a que el manual "DoubleTalk RC8660" no contenía información sobre la comunicación serie y requeríamos esa información para continuar nuestro proyecto. Los resultados que obtuvimos del puerto serie en tiempo real se muestran en el *ANEXO B*.

Realizamos una prueba más con el objetivo de comprobar que los resultados obtenidos del escaneo fueran los correctos; utilizando el programa DOS debug y lenguaje C enviamos los comandos por el puerto serie, y logramos que el sintetizador funcionara correctamente.

Finalmente hicimos pruebas en el microcontrolador y el sintetizador para para verificar su comunicación y funcionamiento.

7.4.1 Ejemplo de prueba del sintetizador

En este ejemplo se muestra la manera de enviar datos al sintetizador utilizando la interfaz serie de cualquier dispositivo que pueda realizar este tipo de transmisión, el ejemplo muestra los datos que deberá recibir el sintetizador para pronunciar cualquier palabra (en nuestro caso la palabra utilizada fue: UNAM).

Utilizando la información descrita anteriormente y, considerando que el sintetizador detecta la velocidad de transferencia automáticamente (atributo por default en el chip) y que el diccionario de excepciones en español ya ha sido descargado en el chip, habilitamos el diccionario para hacer el siguiente ejemplo.

En la siguiente tabla se muestra el ejemplo, la primera columna contiene los comandos en forma literal y en la segunda columna mostramos la traducción en base hexadecimal para su uso en lenguaje ensamblador. *Nota: Los datos de la tabla deberán enviarse al sintetizador en forma consecutiva por el puerto serie para obtener la palabra que deseamos escuchar.*

Comandos y caracteres	Datos en base hexadecimal
CARÁCTER CR (ASC11 13)	\$0D
STOP (CTRL + X)	\$18
MODO TEXTO (~t)	\$01
	\$74
Fin de comando ⁽¹⁾	\$00
HABILITO DICCIONARIO (~u)	\$01
	\$75
Fin de comando	\$00
INDEX MARKER (~99i)	\$01
	\$39
	\$39
	\$69
PALABRA ("UNAM")	\$75
	\$6E
	\$61

	\$6D
Coma ⁽²⁾	\$2C
INDEX MARKER (~98i)	\$01
	\$39
	\$38
	\$69
Fin de comando	\$00

Tabla 7.2 Comandos y caracteres, y su equivalente en hexadecimal enviados al sintetizador de voz para pronunciación de la palabra UNAM

⁽¹⁾ Se recomienda utilizar el valor ASCII 00 como fin de comando.

⁽²⁾ La coma es utilizada al final de una frase.

7.5 INTEGRACIÓN Y PRUEBAS DE LOS ELEMENTOS DEL SISTEMA: GUANTE CON SISTEMA SENSORIAL, INTERFAZ DE ACOPLAMIENTO, MICROCONTROLADOR Y SINTETIZADOR DE VOZ.

Con la información necesaria de ambos dispositivos: microcontrolador y sintetizador de voz, comenzamos a integrar nuestro sistema, es decir, de manera parcial acoplamos todos los elementos que construimos y seleccionamos: guante con sistema sensorial, la interfaz de acoplamiento, el microcontrolador y el sintetizador de voz.

Analizamos de la Lengua de Señas Mexicana las imágenes de las posiciones de los dedos en cada seña y tomando en cuenta la alta sensibilidad de los sensores a la flexión, llegamos a la conclusión de que lo mejor sería asignar tres posiciones a cada dedo: **abierta**, **media** y **cerrada**; entonces, si tenemos cinco dedos y tres posiciones por cada dedo, podríamos formar $5^3=125$ combinaciones en nuestro sistema, aunque muchas combinaciones serían imposibles de realizar por los propios movimientos naturales de la mano.

Continuamos la integración del sistema; primero lo hicimos con el guante, la interfaz de acoplamiento y el microcontrolador; realizamos algunas pruebas para observar el comportamiento de los sensores del guante, la interfaz y el convertidor A/D, encontrando un buen funcionamiento.

Tomando en cuenta que los dedos de la mano tendrían tres posiciones (abierta, media y cerrada) diseñamos e implementamos un algoritmo que programaríamos para que un dedo de la mano encendiera tres LEDs de diferente color correspondientes a cada posición; un LED de color verde para la posición abierta, uno amarillo para la posición media y otro rojo para la posición cerrada. Para encender los LEDs utilizaríamos algunos pines del PUERTO C de la tarjeta del microcontrolador.

Para definir las tres posiciones de cada dedo de la mano fue necesario establecer correctamente los límites, es decir, fijamos un rango de valores para la posición media, y a

partir de esta referencia establecimos las otras dos posiciones. El algoritmo se obtuvo como sigue: Con un multímetro digital tomamos diferentes muestras de valores de voltaje para la posición media de cada dedo, con base en ellas se formaron los límites para esa posición y se estableció que estaría definida por un rango de valores (ver ANEXO C); a partir de la posición media se establecieron las posiciones abierta y cerrada, es decir, si el valor era mayor al límite superior, entonces tendríamos una posición abierta, por el contrario si el valor era menor al límite inferior, entonces tendríamos una posición cerrada, en la *figura 7.13* podemos visualizar las posiciones y la definición de los límites.

Tomando como base el dedo medio, tomamos las muestras y establecimos los rangos de valores en todos los dedos. Posteriormente convertimos los límites de voltaje en valores hexadecimales para definirlos en el programa y que el compilador pudiera interpretarlos. Para saber la resolución de voltaje capturado por el convertidor A/D consideramos que el voltaje que entraría por cada canal estaría en un rango de 0 - 5 V, y dividimos el nivel más alto (5 V) entre el número de combinaciones obtenidas con 8 bits (2^8), por lo tanto la unidad en base hexadecimal representaría 0.01953125 Volts \approx 19.53 mV.

Y para asociar dichos valores consideramos que en el sistema binario únicamente se tienen dos posibles valores 0 o 1 y dado que el convertidor A/D es de 8 bits, se obtuvo que el valor máximo que podríamos tener es de $2^8 = 256$, un rango de 0 a 255 que representa los límites de voltaje de 0 a 5 V en un valor asociado decimal, y de esta manera ya teníamos la información necesaria para programar los valores.

Adicionalmente para obtener una cantidad adecuada de valores posibles fue necesario hacer una división del valor de $5\text{ V} \div 2 = 2.5\text{ V}$ y seguir dividiendo entre 2 hasta obtener un valor bajo de 0.15625 V equivalente a 156.25 mV.



Fig. 7.13 Definición de las posiciones del guante y sus límites de voltaje asociados.

Siguiendo con el programa que encendería tres LEDs dependiendo de las posiciones de la mano, utilizamos el dedo medio del guante conectando los cables de su sensor a la interfaz de acoplamiento; de ahí, el voltaje de salida V_o lo conectamos al canal 0 del convertidor A/D (AN0). Para encender y apagar los LEDs, utilizaríamos 3 de los 8 pines del PUERTO C que programaríamos como salidas a través del registro DDRC. Colocamos 3 LEDs, uno verde, uno amarillo y uno rojo en una tableta protoboard, conectando en serie una resistencia de $470\ \Omega$ a cada uno en la terminal ánodo, y las otras terminales (cátodos) conectándolas a tierra (ver *figura 7.14*).

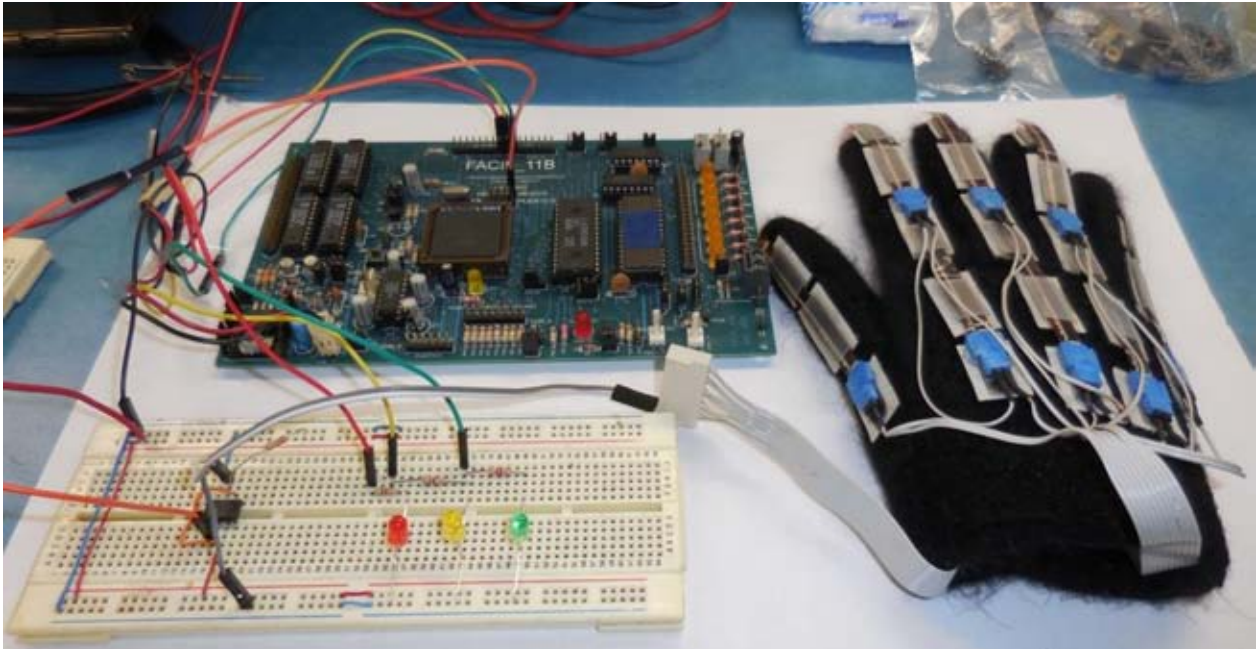


Figura 7.14 Conexiones de los diferentes módulos utilizadas en la ejecución del programa que enciende tres LEDs utilizando el sensor del dedo medio.

Editamos y compilamos el programa (ver *Programa 1* del ANEXO D), colocando en éste los valores de voltaje en su valor correspondiente hexadecimal (que habíamos tomado en las muestras) para el rango de la posición media, después lo descargamos en la memoria RAM interna del microcontrolador y lo ejecutamos, encontrando que funcionaba correctamente.

Dado el éxito del *Programa 1*, decidimos integrar en ese momento el sintetizador de voz con el objetivo de analizar el comportamiento de los cuatro elementos de nuestro sistema en conjunto. Para integrar el sintetizador desarrollamos un nuevo programa tomando como base el programa anterior, en otras palabras ahora con el algoritmo encenderíamos tres LEDs a partir de la posición del dedo (abierto, medio o cerrado) y además el sintetizador generaría en voz una letra asociada a cada estado. Las letras que establecimos fueron, la letra **a** para la posición **abierto**, la letra **b** para la posición **medio** y la letra **c** para la posición **cerrado** (Ver *Tabla 7.3*).

Editamos el nuevo programa (ver *Programa 2* del ANEXO D) en el que configuramos todo lo necesario para la conversión A/D y la transmisión serie, es decir, para que al recibir la señal del sensor conectado a la interfaz de acoplamiento, se realizara la conversión y dependiendo de la posición del dedo, se enviara al sintetizador de voz la letra correspondiente. Una vez terminado, lo compilamos y descargamos en la memoria RAM interna del microcontrolador para ejecutarlo y observar los resultados.

Detectamos errores en la etapa de comunicación entre el microcontrolador y el sintetizador. Lo que sucedía era que cuando hacíamos cambios de una posición a otra, como anteriormente habíamos visto, los LEDs funcionaba bien, pero cuando el sintetizador generaba la voz correspondiente a la letra, éste la repetía muchas veces y cuando

cambiábamos de posición continuaba diciendo la letra de la posición anterior y en algunos casos se retrasaba hasta en más de dos movimientos. La explicación de este problema fue que el sintetizador tiene internamente un buffer en el que se almacena la información que va a convertir a voz, ahora bien, cuando nosotros estamos en una posición enviamos de forma continua la misma letra asociada a tal estado y al cambiar de posición, en el buffer aún existen los datos de la posición anterior y es por eso que se origina el retraso.

Resolvimos el problema colocando condiciones en el programa para que únicamente enviara la información una sola vez si la posición continuaba siendo la misma, y no enviara otro dato hasta cambiar de posición. Probamos nuevamente el programa en el sistema y funcionaba correctamente.

Analizando los resultados del *Programa 2*, determinamos que la parte básica de nuestro sistema ya estaba implementada; ahora tendríamos que hacer un algoritmo más completo considerando los 5 dedos y no sólo para generar letras, sino frases completas.

Posición del dedo	Color de LED	Letra enviada al sintetizador
Abierta	Verde	a
Media	Amarillo	b
Cerrada	Rojo	c

Tabla 7.3 Relación entre la posición del dedo, color de LED y letra enviada al sintetizador de voz, utilizados en el *Programa 2*

Comenzamos a desarrollar el nuevo algoritmo, éste debería ser capaz de enviar frases completas del microcontrolador hacia el sintetizador de voz, a partir de las señales provenientes del guante con sistema sensorial y la interfaz de acoplamiento. Tomando en cuenta que el uso de LEDs en el programa anterior había sido de mucha utilidad para identificar en qué posición se encontraba el dedo, decidimos que para este programa sería de mayor utilidad debido a que utilizaríamos cinco dedos en vez de uno solo; pero pensamos que si para un dedo utilizábamos 3 LEDs, entonces para 5 dedos serían 15, que para nuestra aplicación serían bastantes en cuanto a espacio se refiere. Investigamos y encontramos que existen LEDs bicolores (ver *figura 7.15*), que en realidad encienden de tres maneras: verde, rojo y amarillo dependiendo de la polarización; por tal motivo decidimos utilizar 5 LEDs de ese tipo en vez de 15 para optimizar espacio y para mejorar estéticamente nuestro proyecto. Los implementamos en la misma tableta protoboard conectándoles en serie resistores de 470Ω a cada una de las terminales de los ánodos de cada LED y en los otros extremos conectándolos a la tierra del sistema para polarizarlos correctamente. Utilizamos 10 líneas para encender los 5 LEDs y debido a que la mayoría de los puertos del microcontrolador son de 8 bits, entonces ocupamos 2 puertos, obtuvimos 8 del PUERTO C y 2 del PUERTO B, conectando cada línea de ambos puertos al resistor correspondiente en la tableta. (ver *figura 7.16*)



Figura 7.15 LED bicolor.

De manera similar al programa anterior, hicimos que primero encendieran los LEDs con el programa de acuerdo a las posiciones de los dedos. Para esto hicimos un pequeño algoritmo (ver *ANEXO D*) que convirtiera todas las señales de los sensores, las procesara y encendiera los LEDs conforme a las posiciones.

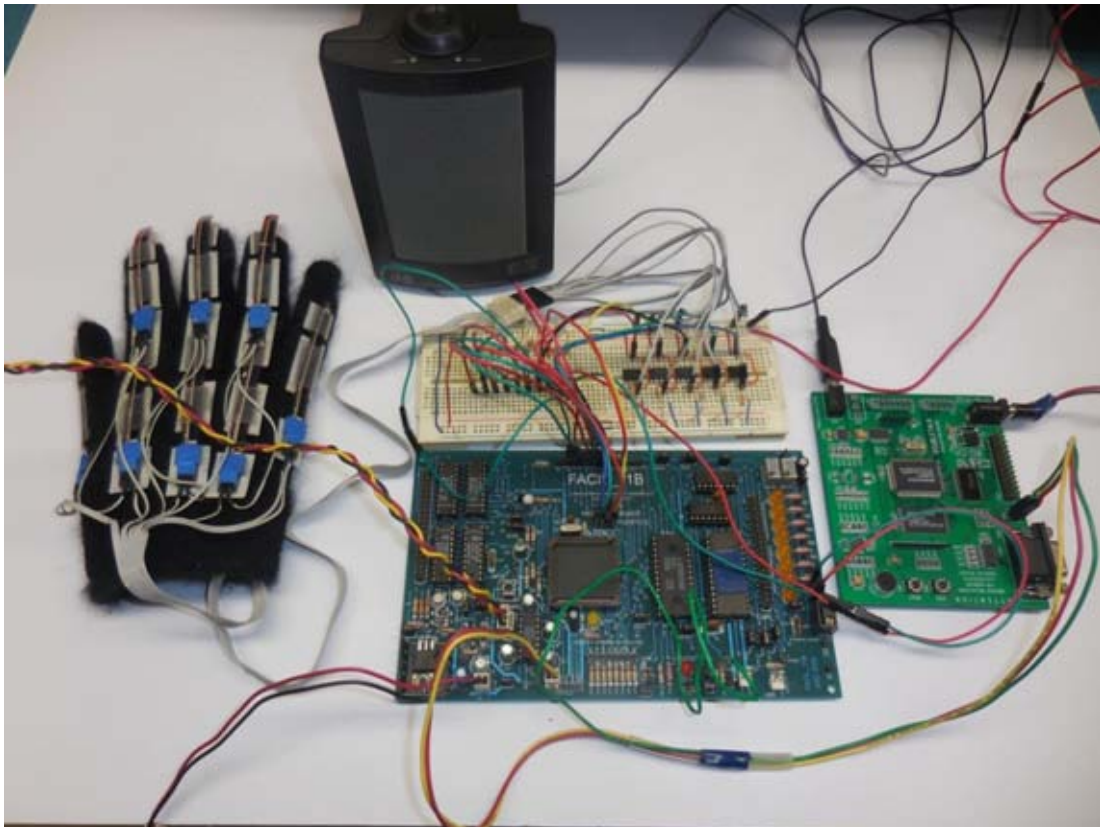


Figura 7.16 Conexiones de los cinco LEDs bicolors con los sensores del guante.

Para introducir las señales de los sensores al convertidor A/D, conectamos las terminales de cada uno de los sensores del guante a la interfaz de acoplamiento, implementando la conexión en una tableta protoboard y de ahí las salidas las conectamos hacia los pines del convertidor A/D de acuerdo a la *Tabla 7.4*.

Dedo	Canal de conversión
Índice	AN0
Medio	AN1
Anular	AN2
Meñique	AN3
Pulgar	AN4

Tabla 7.4 Conexión de los sensores de los dedos del guante y los canales del convertidor A/D

Terminadas las conexiones de los dispositivos y con base en el algoritmo que hicimos, editamos el programa. De forma general, primero se realiza la conversión de 4 entradas de los sensores conectados a las terminales (AN0-AN3) y a continuación se convierte la otra entrada conectada a AN4, tomando en cuenta que el convertidor A/D tiene las entradas multiplexadas. En seguida leímos los resultados de las conversiones y comparamos de forma individual con los datos correspondientes a cada posición de los dedos. Por último, enviamos la información de las posiciones a los puertos para encender los LEDs y observar el estado de los dedos de la mano. Este proceso se hizo de manera continua para detectar los posibles cambios de posición.

El programa no funcionaba bien, debido a que sólo encendían correctamente los LEDs relacionados con los dedos: índice, medio, anular y meñique; pero el LED correspondiente al dedo pulgar solo encendía de color verde y no cambiaba su estado. Revisamos el código fuente de nuestro programa y no encontramos errores, también revisamos la continuidad de los cables de conexión y estaba bien, entonces consideramos la opción de que el canal AN4 del convertidor A/D podría estar dañado y conectamos el cable a la entrada al canal AN5, hicimos los cambios necesarios en el programa y nuevamente hicimos pruebas, encontrando que funcionaba correctamente. Revisamos el manual de configuración de la tarjeta y encontramos que ese pin tiene asociada una función para trabajar con un display LCD y tenía colocado un jumper. Posteriormente quitamos el jumper, nuevamente colocamos las conexiones como estaban y todo funcionó correctamente.

A continuación tendríamos que integrar al programa la etapa de comunicación entre el microcontrolador y el sintetizador de voz. En esta etapa enviaríamos frases completas en lugar de un solo carácter. Para esto deberíamos implementar un algoritmo más completo que el anterior, ya que como habíamos mencionado, ahora estábamos trabajando con cinco dedos. Recordando que en el programa donde se enviaban letras tuvimos el problema con la repetición, ahora con el nuevo algoritmo tendríamos que evitar ese error.

La estructura del algoritmo es la siguiente: asignamos un valor hexadecimal diferente (de 2 bytes) a cada una de las tres posiciones (cerrada, media y abierta) de los cinco dedos, con el propósito de tener bien identificada la posición de cada dedo. Conocidas las cinco posiciones, sumamos sus respectivos valores de cada dedo para obtener un valor único. A la suma de valores del conjunto de posiciones de los cinco dedos registradas en el mismo instante lo llamamos **estado del guante**. Con este estado

conoceríamos bien que seña está realizando la persona que utilice el guante, asimismo evitaríamos errores en el momento de asociar las frases.

Comenzamos la implementación del algoritmo, agregando al programa anterior la configuración del SCI y la configuración del sintetizador necesaria para enviar dichas frases en modo texto. En esta parte del proyecto nos encontramos con un nuevo problema relacionado con el modo de operación con el que estábamos trabajando (modo bootstrap), específicamente con la memoria interna del microcontrolador. Al programar el nuevo algoritmo en el que consideramos el estado de los cinco dedos de la mano, el código fuente (asm) se incrementó bastante y como consecuencia el programa que almacenamos en la memoria interna del microcontrolador (s19) también aumento de tamaño.

La capacidad de la RAM interna del microcontrolador es de 1024 bytes y los programas anteriores ocupaban menos espacio en memoria, pero el nuevo programa generado tenía un tamaño aproximado de 2 kB. Para la solución de este problema recurrimos a trabajar en **expanded mode** o **modo expandido**. Para seleccionar este modo de operación modificamos la configuración de los jumpers de la tarjeta MODA=1 y MODB=1 que establecen los modos de operación del microcontrolador; también fue necesario emplear memorias externas (RAM y EPROM), colocándolas en los sockets que la tarjeta del microcontrolador tiene disponibles. Las memorias RAM y EPROM que utilizamos en la tarjeta son de 32 kB.

Antes de implementar la etapa del sintetizador, realizamos pruebas con el programa que encendía los LEDs para ver si estábamos trabajando correctamente en el modo expandido. Consultamos el mapa de memoria para ubicar la dirección de la memoria RAM en la que podríamos comenzar a programar (ver *ANEXO E*). Enseguida cambiamos en nuestro programa la dirección de inicio y verificamos nuevamente el comportamiento del guante, encontrando un mal funcionamiento porque el microcontrolador no obedecía los cambios de estado en el guante. Revisamos el código fuente del programa y no encontramos errores, después investigamos en el manual de referencia de Freescale Semiconductor del MC68HC11F1 sobre el modo de operación expandido, encontrando que para comunicar el microcontrolador con las memorias externas es necesario utilizar el PUERTO B y el PUERTO F como bus de direcciones y el PUERTO C como bus de datos; ésta fue la causa del error, debido a que nosotros estábamos utilizando dos de esos puertos para encender los LEDs. Ahora tendríamos que pensar como encenderíamos los LEDs, si únicamente nos quedaba disponible el PUERTO A que es de 8 bits y nosotros necesitábamos 10 líneas para encenderlos.

Al revisar el manual de configuración de la tarjeta FACIL11_B encontramos que tiene la modalidad de habilitar en forma adicional dos puertos de salida (SA y SB) y dos más de entrada (EA y EB), para los cuales existe un determinado intervalo de direcciones para acceder a ellos (ver *ANEXO E*). Nuevamente hicimos las modificaciones necesarias en el programa para encender los LEDs a través de estos puertos y visualizar los resultados en forma satisfactoria.

Ya que logramos trabajar en modo expandido, continuamos con nuestro programa. Como habíamos mencionado, solo tendríamos que integrar la etapa de comunicación del microcontrolador con el sintetizador.

Para enviar las frases tendríamos que encontrar la manera de transmitir carácter por carácter como lo habíamos hecho anteriormente, en lenguaje ensamblador hay una instrucción que el compilador interpreta en el manejo de cadenas de caracteres, esta instrucción es **FCC** precedida de la cadena que vamos a enviar entre comillas dobles, implementando un pequeño ciclo para que vaya enviando los caracteres uno a uno y al final indicando el fin de la cadena con la instrucción **FCB**. Cabe mencionar que esta implementación es muy útil, debido a que únicamente tendríamos que modificar la cadena que quisiéramos enviar en el programa, en lugar de modificar las sentencias carácter por carácter. Pero nuevamente nos encontramos con un problema, que el compilador que estábamos utilizando, no reconoce estas instrucciones. Entonces utilizamos el software diseñado para la tarjeta **FACIL11_B** llamado PUMMA11, desarrollado en la Facultad de Ingeniería también.

El PUMMA11 es muy sencillo, práctico y tiene más funciones que el programa con el que habíamos estado trabajando, además de que con el mismo software se pueden descargar los programas en el microcontrolador. Además este software puede trabajar con sistemas operativos actuales y con hardware más reciente.

Compilamos los programas (ver *Programa 3* del ANEXO D) con el software, lo descargamos en la memoria externa de la tarjeta e hicimos algunas pruebas comprobando que funcionaba correctamente, además ya no repetía las frases cuando manteníamos la misma señal con el guante.

7.5.1 Mejoramiento del algoritmo para no repetir las frases.

Durante las pruebas realizadas surgió un problema asociado al algoritmo del **estado del guante**, porque que se repetían los resultados al efectuar la suma de los cinco valores obtenidos de las posiciones de los dedos, que representaban en conjunto una **señal**. Esto ocasionaría que dos señales diferentes produjeran la misma frase, por lo que rediseñamos el algoritmo para que anulara esta repetición por completo. El diseño del algoritmo quedó como sigue: aplicamos la operación OR a cuatro dedos (D1 or D2 or D3 or D4), y utilizamos el quinto dedo como selector de bloques. La operación OR funciona adecuadamente debido a la distribución de los valores en binario asignados a cada una de las posiciones de los dedos, Ver *Tabla 7.5*.

Posición	Dedos									
	D5		D4		D3		D2		D1	
Abierta	01	00000001	40	01000000	10	00010000	04	00000100	01	00000001
Media	02	00000010	80	10000000	20	00100000	08	00001000	02	00000010
Cerrada	03	00000011	C0	11000000	30	00110000	0C	00001100	03	00000011

Tabla 7.5 Valores hexadecimales y binarios asociados a cada posición del dedo

En esta última tabla podemos observar con mayor claridad que con la operación OR se obtiene un valor único para cada una de las combinaciones posibles resultantes de las posiciones (abierto, medio y cerrado) de los cuatro dedos, debido a que en los valores del D1 (dedo meñique) al D4 (dedo índice) se utilizan dos bits diferentes.

El diagrama de flujo del algoritmo se puede observar en el *ANEXO F*. El programa del guante con el algoritmo mejorado para no repetir las frases se puede consultar en el *Programa 4* del *ANEXO D*.

7.6 PRUEBAS DE PROTOTIPO TERMINADO DEL SISTEMA TRADUCTOR DE SEÑAS DACTILOLÓGICAS A VOZ.

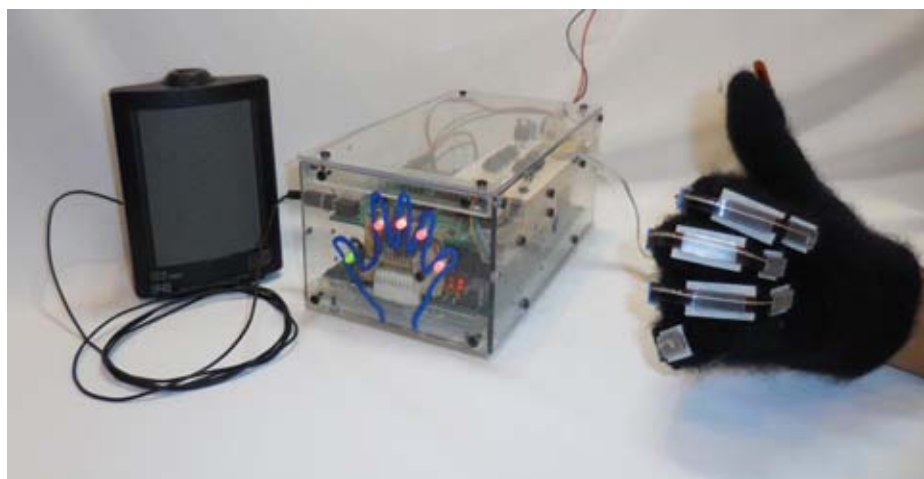


Figura 7.17 Prototipo terminado del sistema traductor de señas dactilológicas a voz.

Con el prototipo terminado, realizamos las pruebas con base en la siguiente tabla, que muestra las frases almacenadas en el microcontrolador que pueden ser generadas a voz cuando con el guante se forman las señas correspondientes. Las señas elegidas para generar dichas frases corresponden a algunas letras del alfabeto dactilológico.

Letra del alfabeto Dactilológico	Posición de los Dedos					Valor Selector Banco	D4 or D3 or D2 or D1	Frase Hablada
	D5	D4	D3	D2	D1			
A	01	C0	30	0C	03	01	FF	¿Me siento mal, me podría ayudar?
CH	01	40	10	0C	03	01	5F	¿Donde esta la salida?
K	01	40	20	0C	03	01	6F	¡Buenos días!
C	02	80	20	0C	03	02	AF	Quiero comprar leche
D	02	40	30	0C	03	02	7F	¿Me podría indicar esta dirección?
F	02	C0	10	04	01	02	D5	Gracias, es usted muy amable
N	02	40	10	0C	03	02	5F	Me da un boleto por favor
B	03	40	10	04	01	03	55	¿Cuál es el precio de este artículo?
E	03	C0	30	0C	03	03	FF	¿Qué hora es?
I	03	C0	30	0C	01	03	F7	Disculpe, no escucho

Posiciones

■	Abierta
■	Media
■	Cerrada

Dedos del guante

D5:	Pulgar
D4:	Índice
D3:	Medio
D2:	Anular
D1:	Meñique

Tabla 7.6 Posiciones de los dedos de las manos para la generación de frases

Al llevar a cabo las pruebas obtuvimos resultados satisfactorios, pues logramos generar las frases elegidas con el lenguaje manual establecido, los sensores colocados en el guante detectaron las posiciones de cada uno de los dedos de manera correcta, aunque es importante señalar que estos se deben ajustar en el guante para cada tamaño de mano y se deben realizar pruebas previas para asegurar la generación correcta de las frases para que el usuario no tenga dificultades en formar cada una de las posiciones de los dedos.

Finalmente, podemos concluir que logramos diseñar e implementar un prototipo constituido por un guante con sensores basado en un sistema analógico-digital para la traducción del lenguaje manual a través de la síntesis de voz, el sistema es capaz de detectar determinados movimientos de la mano para generar voz a través de un sintetizador, permitiendo de esta forma la comunicación por medio del canal auditivo.

CONCLUSIONES

Con base en el sistema traductor de señas dactilológicas a voz que hemos desarrollado y en relación a los objetivos planteados al inicio de este trabajo exponemos las siguientes conclusiones.

Al desarrollar este tipo de aplicación, nos dimos cuenta de lo complejo que resulta ser la implementación de los movimientos de la mano dentro de un sistema electrónico programado. Analizando el comportamiento individual de cada uno de los dedos, resulto mucho más sencillo considerar tres estados en los mismos: abierto, medio y cerrado, para un manejo de señal resultante de un sistema sensorial. Aun así, continuo siendo complejo el tratar de definir los movimientos del dedo pulgar, por la mayor variedad de movimientos que éste presenta.

En cuanto a las resistencias (sensores de flexión) empleados para el registro de las señales de voltaje generadas por el movimiento de la mano, resultan óptimos y apropiados en la aplicación por su flexibilidad y dimensiones. Sin embargo, resulta ser un inconveniente la adquisición de los mismos dado que este tipo de sensores no son tan comercializados ni tan conocidos aún en nuestro país.

Quizá parte del material ajeno a la electrónica con el que está elaborado el dispositivo tal como los rieles, velcro, guante de costura, pegamento, etc., no resulta ser el óptimo, o bien, de la mejor calidad, sin embargo, es funcional pues no presenta inconveniente alguno al trabajar con este, ni mucho menos afecta la funcionalidad y por mucho es más económico que otro tipo de materiales.

Consideramos que una de las etapas clave que nos permitieron un mejor registro del movimiento de los dedos de la mano, es la etapa de amplificación con la que se logró mejorar la señal de voltaje, punto primordial para nuestro sistema. Aquí los circuitos operacionales empleados en conjunto con resistencias para lograr la mejor señal de entrada a nuestro sistema representan nuestra etapa de acondicionamiento.

El microcontrolador como microprocesador optimizado resulta ser un chip muy útil como motor del funcionamiento programado en el desarrollo y control de aplicaciones electrónicas. Además, en nuestro caso, el microcontrolador MC68HC11F1 dentro de sus sockets que tiene integrados para seleccionar la modalidad de trabajar en modo expandido, cuenta con periféricos que nos permitieron la comunicación con el guante y el sintetizador de voz. Además de que nuestro prototipo requirió de varios recursos internos

del microcontrolador como: el convertidor analógico-digital A/D, memoria RAM, memoria EEPROM, puertos de entrada y/o salida y puerto de comunicación serie asíncrona SCI.

En relación a lo anterior, trabajar en modo expandido en el microcontrolador, no solo nos permitió almacenar en memoria el programa codificado para su funcionalidad, sino también hacer trabajar al MC68HC11F1 en forma autónoma desde su memoria EEPROM, haciéndolo independiente de una computadora.

Por su parte, el sintetizador de voz empleado en este proyecto nos resulta muy práctico y útil, no solo porque contiene los periféricos necesarios para la comunicación con el microcontrolador, sino también por permitir el uso de un diccionario en idioma Español y una funcionalidad variada configurable a través de software. En nuestro caso, aunque todo es configurable vía firmware, la configuración de la señal de audio en cuanto a su periodo y amplitud, traducida en tono y velocidad, hacen posible la producción artificial del habla humana.

Cabe mencionar que aunque nuestro dispositivo está limitado por el número de movimientos, traducido en un número reducido de frases, el sistema puede ser mucho más completo aplicando una lógica con logaritmos para mejorar aún más la señal del sistema sensorial, lo cual ampliaría no solo la resolución en el registro de tales movimientos, sino también, se generarían muchas más frases o palabras.

La idea plasmada a través de este trabajo puede ser mejorada con el paso del tiempo ya sea con la sustitución de componentes, o bien, con la implementación de módulos, lógica o algoritmos más completos que permitan una mejor aplicación. Sin embargo, lo que pretendemos es cumplir con la funcionalidad en esencia y haber logrado el objetivo que nos planteamos al inicio de este trabajo.

Realizando este trabajo nos damos cuenta de la relación que guardan las diversas disciplinas de la Ingeniería. Si nos pusiéramos a analizar cada dispositivo, cada sistema, cada implementación, proyecto, etc. relacionado con la ingeniería, nos daremos cuenta que detrás de cada uno de estos existe el trabajo en conjunto de diversas áreas. Para nosotros este proyecto fue interesante y muy bueno haber combinado conocimientos de electrónica y computación.

Proyectos como este sistema traductor representan un precedente para el desarrollo de este tipo de aplicaciones, sobre todo en países como el nuestro, en el sentido de fomentar el desarrollo de tecnología propia que ayuda al mejoramiento de la vida en sociedad. No solo es una aplicación para hacer uso de nuestros conocimientos teóricos, sino un trabajo de investigación e implementación que resulto darnos una gran satisfacción porque hemos demostrado que como ingenieros podemos ayudar a personas con discapacidades a tener una vida de calidad.

Valores de voltaje registrados en función de los movimientos de curvatura del sensor de flexión para diferentes valores de resistencias para la obtención de la resistencia R_M .

Sin etapa de amplificación

		Valores de resistencia (k Ω)															Sensor (k Ω)
		1	1.8	2.2	2.7	3.3	3.9	4.7	10	22	27	33	39	47	56	68	
Curvatura del sensor de flexión (cm)	0	4.73	4.55	4.47	4.35	4.24	4.12	3.98	3.23	2.28	2.03	1.79	1.59	1.42	1.24	1.07	18.445
	0.5	4.74	4.56	4.48	4.37	4.26	4.14	4.01	3.28	2.33	2.08	1.83	1.63	1.46	1.27	1.1	19.185
	1	4.75	4.57	4.49	4.38	4.27	4.16	4.03	3.31	2.36	2.11	1.87	1.67	1.49	1.3	1.12	19.795
	1.5	4.76	4.59	4.51	4.4	4.29	4.18	4.06	3.34	2.41	2.15	1.9	1.71	1.52	1.34	1.15	20.590
	2	4.76	4.59	4.52	4.42	4.32	4.2	4.08	3.37	2.44	2.19	1.94	1.74	1.56	1.37	1.18	21.180
	2.5	4.77	4.62	4.54	4.44	4.34	4.24	4.12	3.43	2.51	2.25	2	1.8	1.61	1.42	1.22	22.280
	3	4.78	4.63	4.56	4.47	4.37	4.27	4.16	3.49	2.58	2.32	2.07	1.86	1.67	1.47	1.28	23.550
	3.5	4.79	4.65	4.59	4.5	4.41	4.31	4.2	3.55	2.65	2.4	2.15	1.94	1.74	1.54	1.34	25.230
	4	4.81	4.68	4.62	4.54	4.45	4.36	4.26	3.64	2.77	2.51	2.26	2.04	1.85	1.64	1.43	27.500
	4.5	4.83	4.71	4.65	4.58	4.5	4.41	4.31	3.73	2.88	2.62	2.37	2.16	1.96	1.74	1.53	30.080
	5	4.85	4.74	4.69	4.62	4.55	4.47	4.38	3.84	3.03	2.77	2.52	2.3	2.1	1.88	1.65	33.800
	5.5	4.86	4.77	4.72	4.66	4.6	4.53	4.45	3.96	3.18	2.94	2.68	2.47	2.26	2.03	1.8	38.350
	6	4.88	4.81	4.77	4.72	4.67	4.61	4.54	4.11	3.41	3.17	2.93	2.72	2.51	2.29	2.04	46.410
6.5	4.91	4.84	4.81	4.77	4.73	4.68	4.62	4.26	3.64	3.42	3.2	2.99	2.79	2.56	2.31	57.420	

Con etapa de amplificación

		Valores de resistencia (k Ω)														Sensor (k Ω)	
		1	1.8	2.2	2.7	3.3	3.9	4.7	10	22	27	33	39	47	56		68
Curvatura del sensor de flexión (cm)	0	4.46	4.46	4.46	4.36	4.24	4.12	3.98	3.24	2.29	2.03	1.9	1.9	1.9	1.9	1.9	18.445
	0.5	4.46	4.45	4.45	4.37	4.26	4.14	4.01	3.27	2.33	2.07	1.9	1.9	1.9	1.9	1.9	19.185
	1	4.45	4.45	4.45	4.39	4.28	4.16	4.03	3.31	2.37	2.11	1.91	1.9	1.9	1.9	1.9	19.795
	1.5	4.45	4.45	4.45	4.41	4.3	4.19	4.06	3.35	2.41	2.15	1.92	1.9	1.9	1.9	1.9	20.590
	2	4.45	4.45	4.45	4.42	4.32	4.21	4.08	3.38	2.45	2.19	1.95	1.9	1.9	1.9	1.9	21.180
	2.5	4.46	4.46	4.46	4.45	4.35	4.24	4.12	3.44	2.51	2.25	2.01	1.9	1.9	1.9	1.9	22.280
	3	4.46	4.46	4.46	4.45	4.38	4.28	4.16	3.49	2.58	2.32	2.07	1.91	1.9	1.9	1.9	23.550
	3.5	4.46	4.46	4.45	4.45	4.41	4.31	4.21	3.56	2.67	2.41	2.16	1.95	1.9	1.9	1.9	25.230
	4	4.45	4.45	4.45	4.45	4.45	4.37	4.26	3.65	2.78	2.52	2.27	2.05	1.9	1.9	1.9	27.500
	4.5	4.46	4.46	4.46	4.45	4.45	4.41	4.32	3.74	2.89	2.63	2.38	2.16	1.96	1.9	1.9	30.080
	5	4.46	4.46	4.46	4.46	4.46	4.45	4.38	3.85	3.03	2.78	2.52	2.31	2.1	1.91	1.9	33.800
	5.5	4.46	4.46	4.46	4.46	4.46	4.45	4.45	3.95	3.18	2.93	2.68	2.47	2.26	2.03	1.9	38.350
	6	4.46	4.46	4.46	4.46	4.46	4.46	4.46	4.11	3.41	3.18	2.94	2.73	2.52	2.29	2.05	46.410
6.5	4.47	4.46	4.46	4.46	4.46	4.46	4.46	4.26	3.63	3.42	3.19	2.98	2.78	2.55	2.31	57.420	

Resultados obtenidos del programa Advanced Serial Port Monitor:**PALABRA SOFIA EN GREETING MESSAGE**

- *EN MS-DOS*

C:\pbs>TYPE PB.GRT

```
↑▲@255wsafia
```

```
18
1E
01
255
W
SOFIA
```

- *ADVANCED SERIAL PORT MONITOR*

```
#0D [len=26]
#0D[len=0]
#18#1E#1255wsafia#0
#4#0
#0D [len=13]
```

PALABRA SOFIA EN TIEMPO REAL

```
#0D [len=25]
#18
c
#1t#0 #199isofi,#198i#0
b
#18#112?#0
#2#622#5#5#1#5#1#0 #5#0#1□#0 #0#0#0#0#0€y#7#0#0
```

PALABRA SOFIA EN TIEMPO REAL (definiendo parámetros desde el programa)

VOZ DE ROBO ROBERT

```
#17o#0#112?#0
#2#66P#5#5#1#0#1#1 #5#6#1□#0 #0#0#0#1#7€y#7#0#0
#154f#0#180p#0#10e#0#16r#0#11t#0
```

*Sin diccionario:

```
#0D [len=25]
#18#1t#0 #199isofi,#198i#0
cb
#18#112?#0
#2#66P#5#5#1#0#1#0 #5#6#1□#0 #0#0#0#1#7€y#7#0#0
```

*Con diccionario:

```
#0D [len=25]
#18#1t#1u#0 #199isofi,#198i#0
cb
#18#112?#0
#2#66P#5#5#1#0#1#1 #5#6#1□#0 #0#0#0#1#7€y#7#0#0
```

PALABRA “SOFI” EN TIEMPO REAL (definiendo parámetros desde el programa)

VOZ DE ROBO ROBERT

- Texto 1
 - ~7o~u
 - sofi

Salida:

```
#0D [len=31]
#18#1t#0 #17o#1u #199isofi,#198i#0
cb
#18#112?#0
#2#66P#5#5#1#0#1#1 #5#6#1□#0 #0#0#0#1#7€y#7#0#0
```

27 bytes recibidos

- Texto 2
 - ~7o

sofi

Salida:

```
#0D [len=31]
#18#1t#0 #17o #199isofi,#198i#0
cb
#18#112?#0
#2#66P#5#5#1#0#1#0 #5#6#1□#0 #0#0#0#1#7€y#7#0#0
• Texto 3
  ~t~u~5v~5s~80p~0e~6r~54f~1x~5a
  sofi
```

Salida:

```
#0D [len=56]
#18
c
#1t#0 #1t#1u#15v#15s#180p#10e#16r#154f#11x#15a #199isofi,#198i#0
b
#18#112?#0
#2#66P#5#5#1#0#1#1 #5#6#1□#0 #0#0#0#1#7€y#7#0#0
```

CAMBIO DEL TIPO DE VOZ

Cambio a la voz por default (Perfect Paul)

```
#10o#0#112?#0
#2#622#5#5#1#5#1#1 #5#0#1□#0 #0#0#0#0#0€y#7#0#0
#150f#0#150p#0#15e#0#10r#0#10t#0
```

CUANDO INICIA EL PROGRAMA SE GENERAN ESTAS CADENAS

```
#0D[len=0]
#18#199i#0
c
#1@#0#16?#0
#1#1#1#1#0#2#0#3
#112?#0
#0 #0#0#0#0#0€y#7#0#0
#10o#0#112?#0
#2#622#5#5#1#5#1#0 #5#0#1□#0 #0#0#0#0#0€y#7#0#0
#150f#0#150p#0#15s#0#15v#0#11x#0#15e#0#15a#0#10o#0
```

Lecturas de voltaje para determinar los límites (rango de valores) para la posición media de los dedos.

Dedo meñique

Resistencia = 33 [kΩ]

Lectura 1	Lectura 2
3.98	4.18
4.10	4.43
4.28	4.36
4.05	4.23
4.38	4.06
4.33	4.21
4.40	4.38
3.98	4.40
4.35	4.00
4.27	4.28

Rango para la posición media:

Valor mínimo	Valor máximo
3.98	4.43

Dedo anular

Resistencia = 68 [kΩ]

Lectura 1	Lectura 2
3.65	3.77
3.43	3.70
3.73	3.63
3.48	3.52
3.55	3.47
3.80	3.94
3.89	3.48
3.70	3.58

3.56	3.92
3.90	3.84

Rango para la posición media:

Valor mínimo	Valor máximo
3.43	3.94

Dedo medio

Resistencia = 68 [kΩ]

Lectura 1	Lectura 2
3.56	3.56
3.42	3.46
3.65	3.44
3.88	3.62
3.71	3.41
3.81	3.57
3.84	3.61
3.73	3.75
3.50	3.84
3.86	3.44

Rango para la posición media:

Valor mínimo	Valor máximo
3.41	3.88

Dedo índice

Resistencia = 68 [kΩ]

Lectura 1	Lectura 2
4.02	3.75
3.65	3.88
3.76	3.96
3.61	3.94
3.87	3.65
3.98	3.72
3.67	3.87

3.62	3.89
3.76	4.01
4.00	3.86

Rango para la posición media:

Valor mínimo	Valor máximo
3.61	4.02

Dedo pulgar

Resistencia = 33 [kΩ]

Lectura 1	Lectura 2
3.58	3.59
3.66	3.63
3.63	3.61
3.60	3.80
3.78	3.94
4.00	3.80
3.67	3.59
3.90	3.99
3.79	3.94
3.85	3.57

Rango para la posición media:

Valor mínimo	Valor máximo
3.57	4.00

Programa 1: Muestra en el Puerto C el comportamiento del dedo medio.

```

; ASIGNACION DE DIRECCIONES
PORTC EQU $06
DDRC EQU $07
OPTION EQU $39
ADCTL EQU $30
ADR1 EQU $31

        ORG $0100
inicio  LDX #$1000
        LDAA #$FF
        STAA DDRC,X ;se configura PUERTO C

; CONFIGURACION DEL CONVERTIDOR ANALOGICO-DIGITAL

        LDAA #$80
        STAA OPTION,X ;Se enciende el convertidor A/D
        LDAA #$30
        STAA ADCTL,X ;Se habilita escaneo continuo y
; ;se seleccionan conversion de cuatro canales
; ;aun no seleccionados

; PROGRAMA PRINCIPAL

sell    LDAA #$30
        STAA ADCTL,X
        JSR espera

; PRIMER CANAL
;-----
        LDAA #$00
        STAA PORTC,X
        LDAA ADR1,X
        CMPA #$B7
        BLO compl
        BSET PORTC,X,$01 ; indice cerrado.
        JMP sell

compl   CMPA #$AA
        BLO abier1
        BSET PORTC,X,$03 ; indice medio
        JMP sell

abier1 BSET PORTC,X,$02 ; indice abierto
        JMP sell

;Esperar conversion completa de 4 canales
;-----
espera BRCLR ADCTL,X,$80,espera
        RTS
;-----

```

Programa 2: Despliega en el puerto C el comportamiento del dedo medio y transmite por el puerto serie el carácter correspondiente (a, b o c).

```

PORTC EQU $06
DDRC EQU $07
OPTION EQU $39
ADCTL EQU $30
ADR1 EQU $31
BAUD EQU $2B
SCCR1 EQU $2C
SCCR2 EQU $2D
SCSR EQU $2E
SCDR EQU $2F

```

* ORIGEN DEL PROGRAMA, CONFIGURACION DE PUERTOS, SCI y CAD

```

;-----
inicio   ORG $100
         LDX #$1000
         LDAA #$FF
         STAA DDRC,X      ;se configura PUERTO C

; CONFIGURACION DEL PUERTO SERIE

         LDAA #$30
         STAA BAUD,X
         LDAA #$00
         STAA SCCR1,X    ; TRANSMITIR 8 BITS DE DATOS
         LDAA #$0C
         STAA SCCR2,X    ;INTERRUPCIONES DESHABILITADAS
                           ;HABILITAR TRANSMISOR Y RECEPTOR

; CONFIGURACION DEL CONVERTIDOR ANALOGICO-DIGITAL

         LDAA #$80
         STAA OPTION,X  ;Se enciende el convertidor A/D
         LDAA #$30
         STAA ADCTL,X   ;Se habilita escaneo continuo y
;                               ;se seleccionan conversion de cuatro canales
;                               ;aun no seleccionados

; PROGRAMA PRINCIPAL

sell     JSR config
         LDAA #$30
         STAA ADCTL,X
         JSR espera

;----- Primer canal -----
         LDAA ADR1,X
         CMPA #$BB
         BLO comp1
         LDAB #$01
         STAB PORTC,X  ;pulgar cerrado
         BSR ind_cer
         BRA sell

comp1    CMPA #$AA
         BLO abier1
         LDAB #$03
         STAB PORTC,X ;pulgar medio
         BSR ind_med
         BRA sell

```

```
abier1  LDAB #$02
        STAB PORTC,X ;pulgar abierto
        JSR  ind_ab
        BRA  sell

;-----
config  LDAA #$00
        BSR  enviar
        LDAA #$01
        BSR  enviar
        LDAA #$75 ; ~u habilito diccionario
        BSR  enviar
        LDAA #$00
        BSR  enviar
        RTS

index   LDAA #$01
        BSR  enviar
        LDAA #$39
        BSR  enviar
        LDAA #$39
        BSR  enviar
        LDAA #$69 ;Index Marker
        BSR  enviar
        RTS

;-----
ind_ab  BSR  index
        LDAA #'A'
        BSR  enviar
        BSR  coma
        RTS

ind_med BSR  index
        LDAA #'B'
        BSR  enviar
        BSR  coma
        RTS

ind_cer BSR  index
        LDAA #'C'
        BSR  enviar
        BSR  coma
        RTS

;-----
coma    LDAA #$2C
        BSR  enviar

        LDAA #$01
        BSR  enviar
        LDAA #$39
        BSR  enviar
        LDAA #$38
        BSR  enviar
        LDAA #$69
        BSR  enviar
        LDAA #$00;Index Marker
        BSR  enviar
        RTS

enviar  BRCLR SCSR,X,$C0,enviar
        STAA SCDR,X
        RTS

        END
```

Programa 3: Traduce las señas formadas con los cinco dedos de la mano en voz.

```

* ASIGNACION DE DIRECCIONES
;-----
PORTA    EQU    $00
DDRA     EQU    $01
PORTSA   EQU    $1900
PORTSB   EQU    $19FF
AUXA     EQU    $30
AUXB     EQU    $31
ESTADO   EQU    $20
OPTION   EQU    $39
ADCTL    EQU    $30
ADR1     EQU    $31
ADR2     EQU    $32
ADR3     EQU    $33
ADR4     EQU    $34
BAUD     EQU    $2B
SCCR1    EQU    $2C
SCCR2    EQU    $2D
SCSR     EQU    $2E
SCDR     EQU    $2F

* ORIGEN DEL PROGRAMA, CONFIGURACION DE PUERTOS, SCI y CAD
;-----
                ORG    $8000
inicio      LDX    #$1000    ; para apuntar a los registros

* Configuracion del SCI

                LDAA   #$30
                STAA   BAUD,X    ; Se configura la velocidad a 9600[baudios]
                LDAA   #$00
                STAA   SCCR1,X   ; Transmitir 8 bits de datos
                LDAA   #$0C
                STAA   SCCR2,X   ; Interrupciones deshabilitadas
;                                     Habilitar transmisor y receptor

* CONFIGURACION DEL CONVERTIDOR ANALOGICO-DIGITAL

                LDAA   #$80
                STAA   OPTION,X  ; Se enciende el convertidor A/D

* PROGRAMA PRINCIPAL
;-----
                JSR    config
                JSR    limpia

sel1           JSR    edo_gte
                JSR    limpia
                LDAA   #$30      ; Escaneo continuo y se seleccion de 4 canales
                STAA   ADCTL,X   ; del primer grupo
                JSR    esperap
                LDY    #$4000
                JMP    indi

sel2           LDAA   #$25      ; Escaneo continuo y se seleccion del 1 canal
                STAA   ADCTL,X   ; del segundo grupo
                JSR    espera
                JMP    pulg

* PRIMER CANAL: DEDO INDICE
;-----
indi          LDAA   #$00
                STAA   AUXA,Y

```

```

        STAA PORTSA
        LDAA ADR1,X
        CMPA #$B7          ; indice cerrado.
        BLO  comp1
        BSET AUXA,Y,$01
        LDAB #$03
        BRA  edo_indi

comp1   CMPA #$AA          ; indice medio
        BLO  abier1
        BSET AUXA,Y,$03
        LDAB #$02
        BRA  edo_indi

abier1  BSET AUXA,Y,$02
        LDAB #$01          ; indice abierto
        BRA  edo_indi

edo_indi LDAA AUXA,Y
        STAA PORTSA
        STAB $4000
        BRA  medi

```

* SEGUNDO CANAL: DEDO MEDIO

```

;-----
medi    LDAA ADR2,X
        CMPA #$C7          ; medio cerrado
        BLO  comp2
        BSET AUXA,Y,$04
        LDAB #$06
        BRA  edo_medi

comp2   CMPA #$AF          ; medio medio
        BLO  abier2
        BSET AUXA,Y,$0C
        LDAB #$05
        BRA  edo_medi

abier2  BSET AUXA,Y,$08
        LDAB #$04          ; medio abierto
        BRA  edo_medi

edo_medi LDAA AUXA,Y
        STAA PORTSA
        STAB $4001
        BRA  anul

```

* TERCER CANAL: DEDO ANULAR

```

;-----
anul    LDAA ADR3,X
        CMPA #$C7          ; anular cerrado
        BLO  comp3
        BSET AUXA,Y,$10
        LDAB #$09
        BRA  edo_anul

comp3   CMPA #$B3          ; anular medio
        BLO  abier3
        BSET AUXA,Y,$30
        LDAB #$08
        BRA  edo_anul

abier3  BSET AUXA,Y,$20
        LDAB #$07          ; anular abierto
        BRA  edo_anul

```

```

edo_anul LDAA AUXA,Y
        STAA PORTSA
        STAB $4002
        BRA  meni

* CUARTO CANAL: DEDO MENIQUE
;-----
meni    LDAA ADR4,X
        CMPA #$DF          ; menique cerrado
        BLO  comp4
        BSET AUXA,Y,$40
        LDAB #$0C
        BRA  edo_meni

comp4   CMPA #$AB          ; menique medio
        BLO  abier4
        BSET AUXA,Y,$C0
        LDAB #$0B
        BRA  edo_meni

abier4  BSET AUXA,Y,$80
        LDAB #$0A          ; menique abierto
        BRA  edo_meni

edo_meni LDAA AUXA,Y
        STAA PORTSA
        STAB $4003
        JMP  sel2

* QUINTO CANAL: DEDO PULGAR
;-----
pulg    LDAA #$00
        STAA AUXB,Y
        STAA PORTSB
        LDAB ADR2,X
        CMPB #$B4          ; pulgar cerrado
        BLO  comp5
        BSET AUXB,Y,$01
        LDAB #$0F
        BRA  edo_pulg

comp5   CMPB #$A3          ; pulgar medio
        BLO  abier5
        BSET AUXB,Y,$03
        LDAB #$0E
        BRA  edo_pulg

abier5  BSET AUXB,Y,$02
        LDAB #$0D          ; pulgar abierto
        BRA  edo_pulg

edo_pulg LDAA AUXB,Y
        STAA PORTSB
        STAB $4004
        JMP  sel1

***** SUBRUTINAS *****

* Configuracion del Sintetizador RC8660
;-----
config  LDAA #$00
        JSR  enviar
        LDAA #$01
        JSR  enviar

```

```

        LDAA #$75          ; ~u habilito diccionario
        JSR  enviar
        LDAA #$00
        JSR  enviar
        RTS

* Limpiar direcciones de memoria para comparacion
;-----
limpia  LDAA #$00
        LDY  #$4000
sigte   STAA $00,Y
        INY
        CPY  #$4005
        BNE  sigte
        RTS

* Estado del guante
;-----
edo_gte CLRB
        LDY  #$4000
sigtel  ADDB $00,Y
        INY
        CPY  #$4005
        BNE  sigtel
        CMPB ESTADO,Y   ; Se compara el resultado anterior de suma para no repetir
        BEQ  no_rep
        STAB ESTADO,Y   ; Se almacena el resultado de la suma
        CMPB #$2C
        BEQ  frase1
        CMPB #$2A
        BEQ  frase2
        CMPB #$28
        BEQ  frase3
        CMPB #$27
        BEQ  frase4
        CMPB #$25
        BEQ  frase5
        BRA  no_rep

* Esperar conversion completa de 4 canales en el CAD
;-----
esperap JSR  pausa
espera  BRCLR ADCTL,X,$80,espera
        RTS

***** FRASES *****

frase1  LDY  #cad1
        BRA  ini_cad
frase2  LDY  #cad2
        BRA  ini_cad
frase3  LDY  #cad3
        BRA  ini_cad
frase4  LDY  #cad4
        BRA  ini_cad
frase5  LDY  #cad5
        BRA  ini_cad

ini_cad BSR  index
        BSR  cadena
no_rep  RTS

cadena  LDAA $00,y

```

```

        CMPA #$00
        BEQ salir
        BSR enviar
        INY
        BRA cadena
salir   BSR coma
        RTS

```

* Index Marker: inicio de cadena: antes de la cadena se envia 99i

```

;-----
index  LDAA #$01
        BSR enviar
        LDAA #$39
        BSR enviar
        LDAA #$39
        BSR enviar
        LDAA #$69
        BSR enviar
        RTS

```

* Index marker: Fin de cadena: se envia despues de la cadena ,98i

```

;-----
coma   LDAA #$2C
        BSR enviar

        LDAA #$01
        BSR enviar
        LDAA #$39

        BSR enviar
        LDAA #$38
        BSR enviar
        LDAA #$69
        BSR enviar
        LDAA #$00
        BSR enviar
        RTS

```

* Enviar caracteres por el SCI

```

;-----
enviar BRCLR SCSR,X,$C0,enviar
        STAA SCDR,X
        RTS

```

* Pausa

```

;-----
pausa  LDY  #$FFFF
decr   DEY
        CPY  #$0000
        BNE  decr
        RTS

```

***** CADENAS *****

```

cad1   FCC  "necesito ayuda"
        FCB  $00

cad2   FCC  "¿Donde esta la salida?"
        FCB  $00

cad3   FCC  "¡Buenos dias!"
        FCB  $00

cad4   FCC  "Quiero comprar leche"
        FCB  $00

```



```

cad5      FCC  "¿Como llego al hospital?"
          FCB  $00
*****
;          END
; Fin del programa
;-----

```

Programa 4: Traduce las señas formadas con los cinco dedos de la mano en voz, e incluye un algoritmo para no repetir las frases.

```

* ASIGNACION DE DIRECCIONES
;-----
PORTA     EQU  $00
DDRA      EQU  $01
PORTSA    EQU  $1900
PORTSB    EQU  $19FF
AUXA      EQU  $30
AUXB      EQU  $31
ESTADO    EQU  $20
ESTADOP   EQU  $21
OPTION    EQU  $39
ADCTL     EQU  $30
ADR1      EQU  $31
ADR2      EQU  $32
ADR3      EQU  $33
ADR4      EQU  $34
BAUD      EQU  $2B
SCCR1     EQU  $2C
SCCR2     EQU  $2D
SCSR      EQU  $2E
SCDR      EQU  $2F

* ORIGEN DEL PROGRAMA, CONFIGURACION DE PUERTOS, SCI y CAD
;-----
          ORG  $2000
inicio   LDX  #$1000    ; para apuntar a los registros

* Configuracion del SCI
          LDAA #$30
          STAA BAUD,X    ; Se configura la velocidad a 9600[baudios]
          LDAA #$00
          STAA SCCR1,X   ; Transmitir 8 bits de datos
          LDAA #$0C
          STAA SCCR2,X   ; Interrupciones deshabilitadas
;                               Habilitar transmisor y receptor

* CONFIGURACION DEL CONVERTIDOR ANALOGICO-DIGITAL
          LDAA #$80
          STAA OPTION,X ; Se enciende el convertidor A/D

* PROGRAMA PRINCIPAL
;-----
          JSR  config
          JSR  limpia

sell     JSR  edo_gte
          JSR  limpia
          LDAA #$30      ; Escaneo continuo y se seleccion de 4 canales
          STAA ADCTL,X   ; del primer grupo

```

```

        JSR  esperap
        LDY  #$4000
        JMP  indi
sel2    LDAA  #$25          ; Escaneo continuo y se seleccion del 1 canal
        STAA ADCTL,X      ; del segundo grupo
        JSR  espera
        JMP  pulg

```

* PRIMER CANAL: DEDO INDICE

```

;-----
indi    LDAA  #$00
        STAA AUXA,Y
        STAA PORTSA
        LDAA ADR1,X
        CMPA #$CE          ; indice cerrado.
        BLO  comp1
        BSET AUXA,Y,$01
        LDAB #$C0
        BRA  edo_indi

comp1   CMPA  #$B9          ; indice medio
        BLO  abier1
        BSET AUXA,Y,$03
        LDAB #$80
        BRA  edo_indi

abier1  BSET  AUXA,Y,$02
        LDAB #$40          ; indice abierto
        BRA  edo_indi

edo_indi LDAA  AUXA,Y
        STAA PORTSA
        STAB $4000
        BRA  medi

```

* SEGUNDO CANAL: DEDO MEDIO

```

;-----
medi    LDAA  ADR2,X
        CMPA #$C7          ; medio cerrado
        BLO  comp2
        BSET AUXA,Y,$04
        LDAB #$30
        BRA  edo_medi

comp2   CMPA  #$AF          ; medio medio
        BLO  abier2
        BSET AUXA,Y,$0C
        LDAB #$20
        BRA  edo_medi

abier2  BSET  AUXA,Y,$08
        LDAB #$10          ; medio abierto
        BRA  edo_medi

edo_medi LDAA  AUXA,Y
        STAA PORTSA
        STAB $4001
        BRA  anul

```

* TERCER CANAL: DEDO ANULAR

```

;-----
anul    LDAA  ADR3,X
        CMPA #$CA          ; anular cerrado
        BLO  comp3
        BSET AUXA,Y,$10

```

```

        LDAB #$0C
        BRA  edo_anul

comp3   CMPA  #$B0          ; anular medio
        BLO  abier3
        BSET AUXA,Y,$30
        LDAB #$08
        BRA  edo_anul

abier3  BSET  AUXA,Y,$20
        LDAB #$04          ; anular abierto
        BRA  edo_anul

edo_anul LDAA  AUXA,Y
        STAA PORTSA
        STAB $4002
        BRA  meni

* CUARTO CANAL: DEDO MENIQUE
;-----
meni    LDAA  ADR4,X
        CMPA  #$E3          ; menique cerrado
        BLO  comp4
        BSET AUXA,Y,$40
        LDAB #$03
        BRA  edo_meni

comp4   CMPA  #$CC          ; menique medio
        BLO  abier4
        BSET AUXA,Y,$C0
        LDAB #$02
        BRA  edo_meni

abier4  BSET  AUXA,Y,$80
        LDAB #$01          ; menique abierto
        BRA  edo_meni

edo_meni LDAA  AUXA,Y
        STAA PORTSA
        STAB $4003
        JMP  sel2

* QUINTO CANAL: DEDO PULGAR
;-----
pulg    LDAA  #$00
        STAA AUXB,Y
        STAA PORTSB
        LDAB ADR2,X
        CMPB  #$CD          ; pulgar cerrado
        BLO  comp5
        BSET AUXB,Y,$01
        LDAB #$03
        BRA  edo_pulg

comp5   CMPB  #$B7          ; pulgar medio
        BLO  abier5
        BSET AUXB,Y,$03
        LDAB #$02
        BRA  edo_pulg

abier5  BSET  AUXB,Y,$02
        LDAB #$01          ; pulgar abierto
        BRA  edo_pulg

edo_pulg LDAA  AUXB,Y

```

```

        STAA PORTSB
        STAB $4004
        JMP sell

***** SUBRUTINAS *****

* Configuracion del Sintetizador RC8660
;-----
config  LDAA #$00
        JSR enviar
        LDAA #$01
        JSR enviar
        LDAA #$75          ; ~u habilito diccionario
        JSR enviar
        LDAA #$00
        JSR enviar
        RTS

* Limpiar direcciones de memoria para comparacion
;-----
limpia  LDAA #$00
        LDY #$4000
sigte   STAA $00,Y
        INY
        CPY #$4005
        BNE sigte
        RTS

* Estado del guante
;-----
edo_gte CLRB
        CLRA
        LDY #$4000
sigte1  ORAB $00,Y
        INY
        CPY #$4004
        BNE sigte1
        ADDA $00,Y        ; almaceno en A el estado del pulgar
        CMPB ESTADO,Y    ; Se compara el resultado anterior de la or para no repetir
        BNE habla
        CMPA ESTADOP,Y
        BEQ no_rep1
habla   STAA ESTADOP,Y    ;Se almacena contenido de $4005 en la variable ESTADOP
        STAB ESTADO,Y    ; Se almacena el resultado de la OR en la variable ESTADO
        CMPA #$01
        BNE banco2
banco1  CMPB #$FF
        BEQ frase1
        CMPB #$5F
        BEQ frase2
        CMPB #$6F
        BEQ frase3
        BRA no_rep        ;fin de banco1
banco2  CMPA #$02
        BNE banco3
        CMPB #$AF
        BEQ frase4
        CMPB #$7F
        BEQ frase5
        CMPB #$D5
        BEQ frase6
        CMPB #$5F
        BEQ frase7
        BRA no_rep        ;fin de banco2
banco3  CMPB #$55

```

```

        BEQ frase8
        CMPB #$FF
        BEQ frase9
        CMPB #$F7
        BEQ frase10
no_rep1  BRA no_rep      ;fin de banco3

* Esperar conversion completa de 4 canales en el CAD
;-----
esperap JSR pausa
espera  BRCLR ADCTL,X,$80,espera
        RTS

***** FRASES *****

frase1  LDY #cad1
        BRA ini_cad
frase2  LDY #cad2
        BRA ini_cad
frase3  LDY #cad3
        BRA ini_cad
frase4  LDY #cad4
        BRA ini_cad
frase5  LDY #cad5
        BRA ini_cad
frase6  LDY #cad6
        BRA ini_cad
frase7  LDY #cad7
        BRA ini_cad
frase8  LDY #cad8
        BRA ini_cad
frase9  LDY #cad9
        BRA ini_cad
frase10 LDY #cad10
        BRA ini_cad

ini_cad BSR index
        BSR cadena
no_rep  RTS

cadena  LDAA $00,y

        CMPA #$00
        BEQ salir
        BSR enviar
        INY
        BRA cadena
salir   BSR coma
        RTS

* Index Marker: inicio de cadena: antes de la cadena se envia 99i
;-----
index   LDAA #$01
        BSR enviar
        LDAA #$39
        BSR enviar
        LDAA #$39
        BSR enviar
        LDAA #$69
        BSR enviar
        RTS

* Index marker: Fin de cadena: se envia despues de la cadena ,98i
;-----
coma    LDAA #$2C

```

```
BSR enviar

LDAA #$01
BSR enviar
LDAA #$39

BSR enviar
LDAA #$38
BSR enviar
LDAA #$69
BSR enviar
LDAA #$00
BSR enviar
RTS

* Enviar caracteres por el SCI
;-----
enviar BRCLR SCSR,X,$C0,enviar
      STAA SCDR,X
      RTS

* Pausa
;-----
pausa LDY  $FFFF
decr  DEY
      CPY  $0000
      BNE decr
      RTS
***** CADENAS *****

cad1  FCC  "¿Me siento mal, me podría ayudar?"
      FCB  $00

cad2  FCC  "¿Donde esta la salida?"
      FCB  $00

cad3  FCC  "¡Buenos dias!"
      FCB  $00

cad4  FCC  "Quiero comprar leche"
      FCB  $00

cad5  FCC  "¿Me podría indicar esta dirección?"
      FCB  $00

cad6  FCC  "Gracias, es usted muy amable"
      FCB  $00

cad7  FCC  "Me da un boleto por favor"
      FCB  $00

cad8  FCC  "¿Cual es el precio de este articulo?"
      FCB  $00

cad9  FCC  "¿Qué hora es?"
      FCB  $00

cad10 FCC  "Disculpe, no escucho"
      FCB  $00
*****
;      END
; Fin del programa
;-----
```

Mapas de memoria con los que puede operar la CMT FACIL_11BB

La CMT FACIL_11BB puede operar con diferentes mapas de memoria; cuando la misma opera en modo single-chip o boot-strap, los mapas correspondientes son los naturales del microcontrolador 68HC11F1, al operar el mismo en los modos mencionados, ver las Figuras A.1 y A.2.



Figura A1. Mapa de memoria de la CMT FACIL_11B operando en modo Boot-Strap.

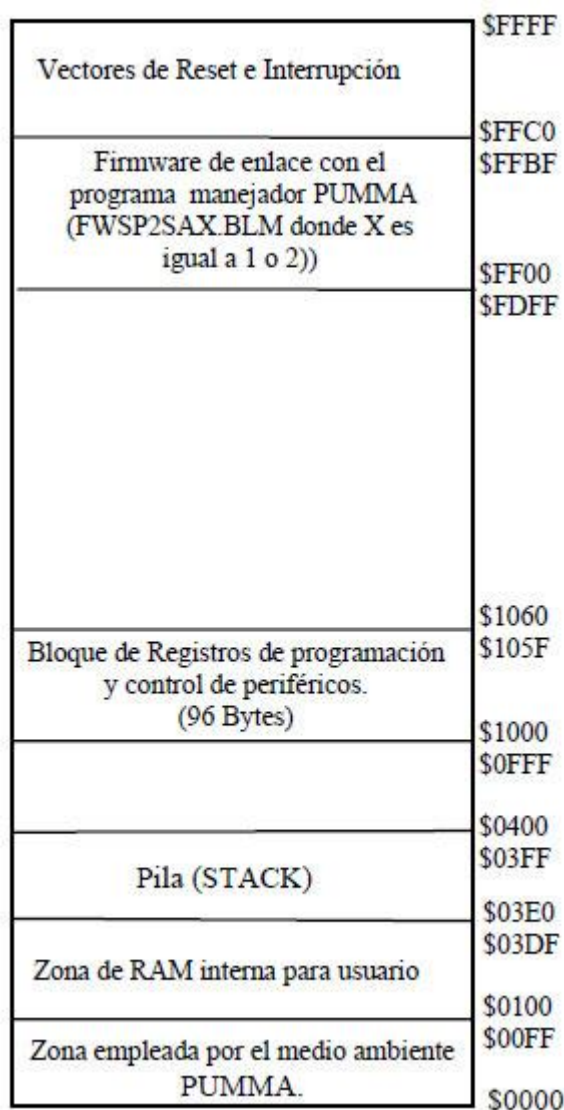


Figura A2. Mapa de memoria de la CMT FACIL_11B operando en modo Single-Chip.

Mapas de memoria de la CMT FACIL_11B operando en modo EXPANDIDO

Para la operación en modo expandido pueden configurarse dos mapas de memoria denominados como Mapa EA y Mapa EB.

\$FFFF	Vectores de Reset e Interrupción
\$FFC0 \$FFBF	Zona de EEPROM interna para usuario
\$FEA0 \$FE9F	Firmware de enlace con manejador PUMMA EEPROM interna (FWFCLEA2.BLM)
\$FE00 \$DFFF	EPROM externa para usuario.
\$E000 \$DFFF	RAM externa para usuario.
\$C000 \$BFFF	
\$2000 \$1FFF	Submapa alterno de Puertos
\$1C00 \$1BFF	Submapa de Puertos
\$1800 \$17FF	
\$1060 \$105F	Registros de control y programación de periféricos.
\$1000 \$0FFF	
\$0400 \$03FF	Pila (Stack)
\$03E0 \$03DF	Zona de RAM interna para usuario
\$0100 \$00FF	Zona empleada por el medio ambiente PUMMA
\$ 0000	

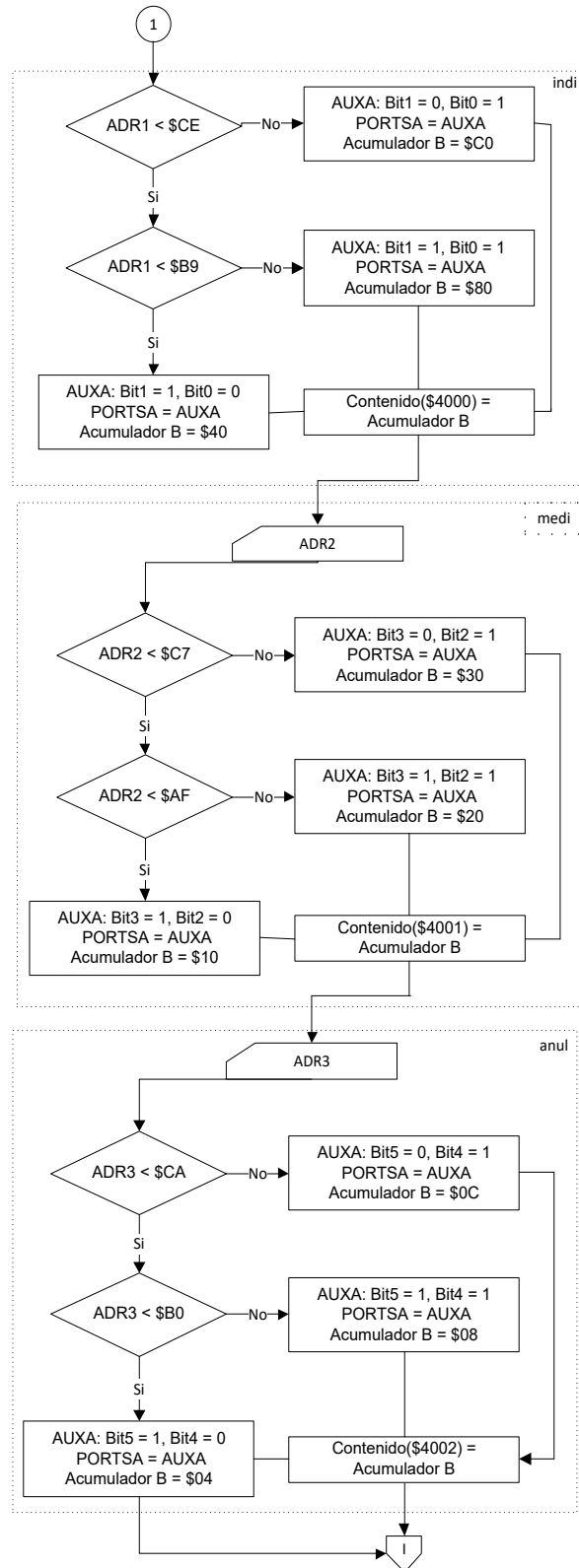
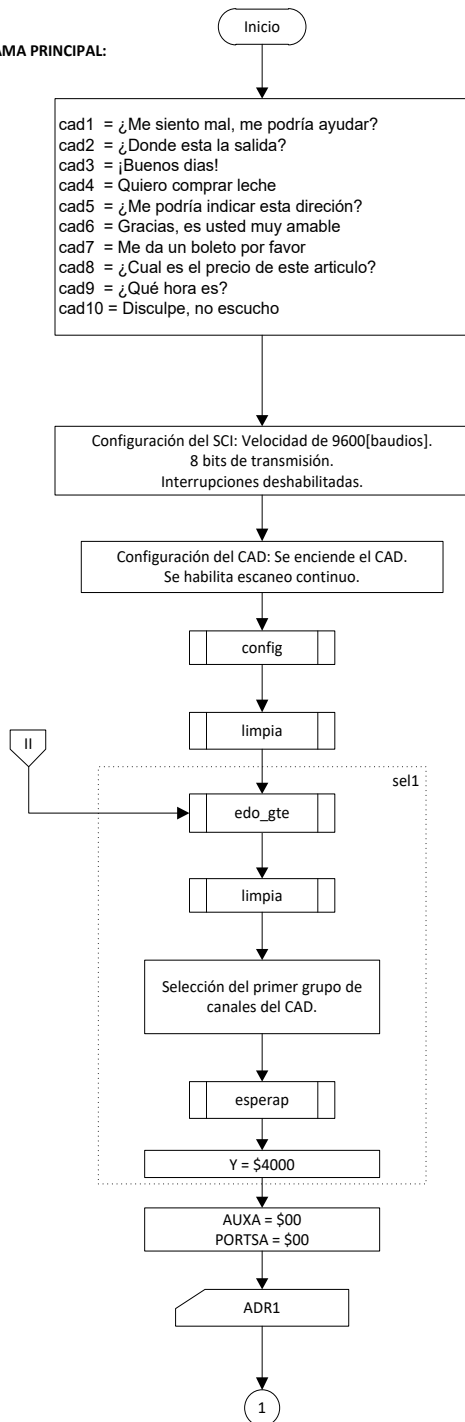
Figura A3. Mapa de memoria EA de la CMT FACIL_11B operando en modo expandido.

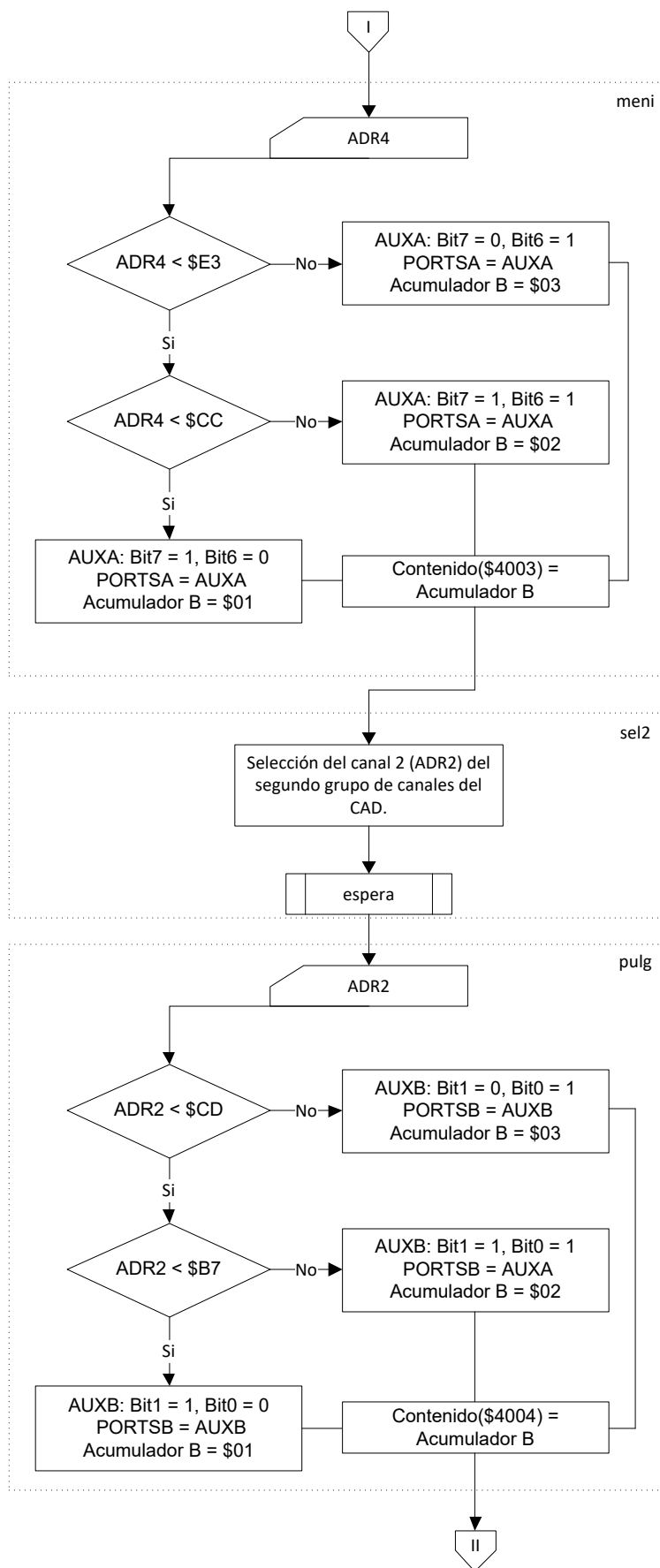
\$FFF	Vectores de Reset e Interrupción
\$FFC0 \$FFBF	Zona de EEPROM interna para usuario
\$FEA0 \$FE9F	Firmware de enlace con manejador PUMMA EEPROM interna (FWFCLEB2.BLM)
\$FE00 \$DFDF	EPR0M externa para usuario.
\$8000 \$7FFF	RAM externa para usuario
\$2000 \$1FFF	Submapa alternativo de Puertos
\$1C00 \$1BFF	Submapa de Puertos
\$1800 \$17FF	RAM externa para usuario
\$1060 \$105F	Registros de control y programación de periféricos.
\$1000 \$0FFF	RAM externa para usuario
\$0400 \$03FF	Pila (Stack)
\$03E0 \$03DF	Zona de RAM interna para usuario
\$0100 \$00FF	Zona empleada por el medio ambiente PUMMA
\$ 0000	

Figura A4. Mapa de memoria EB de la CMT FACIL_11B operando en modo expandido.

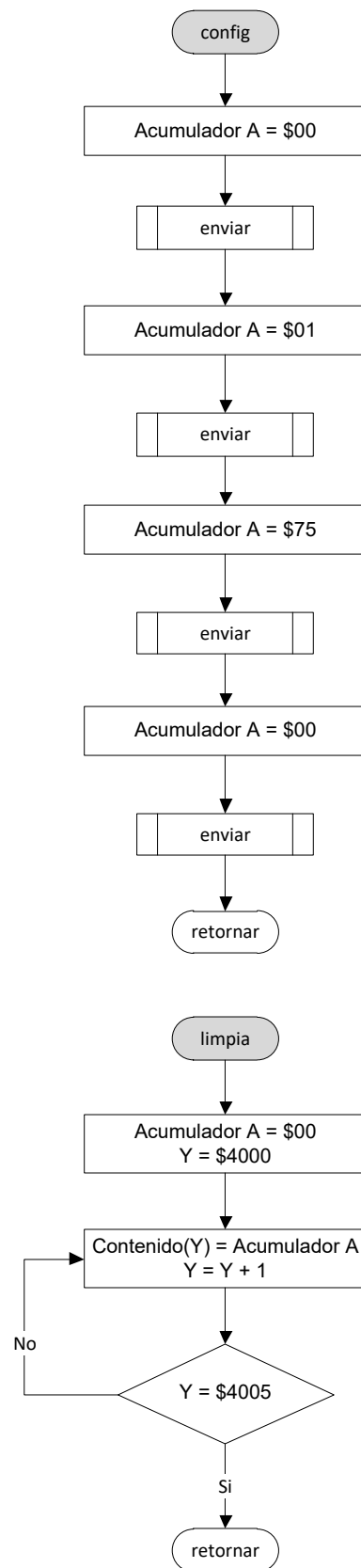
ANEXO F

PROGRAMA PRINCIPAL:

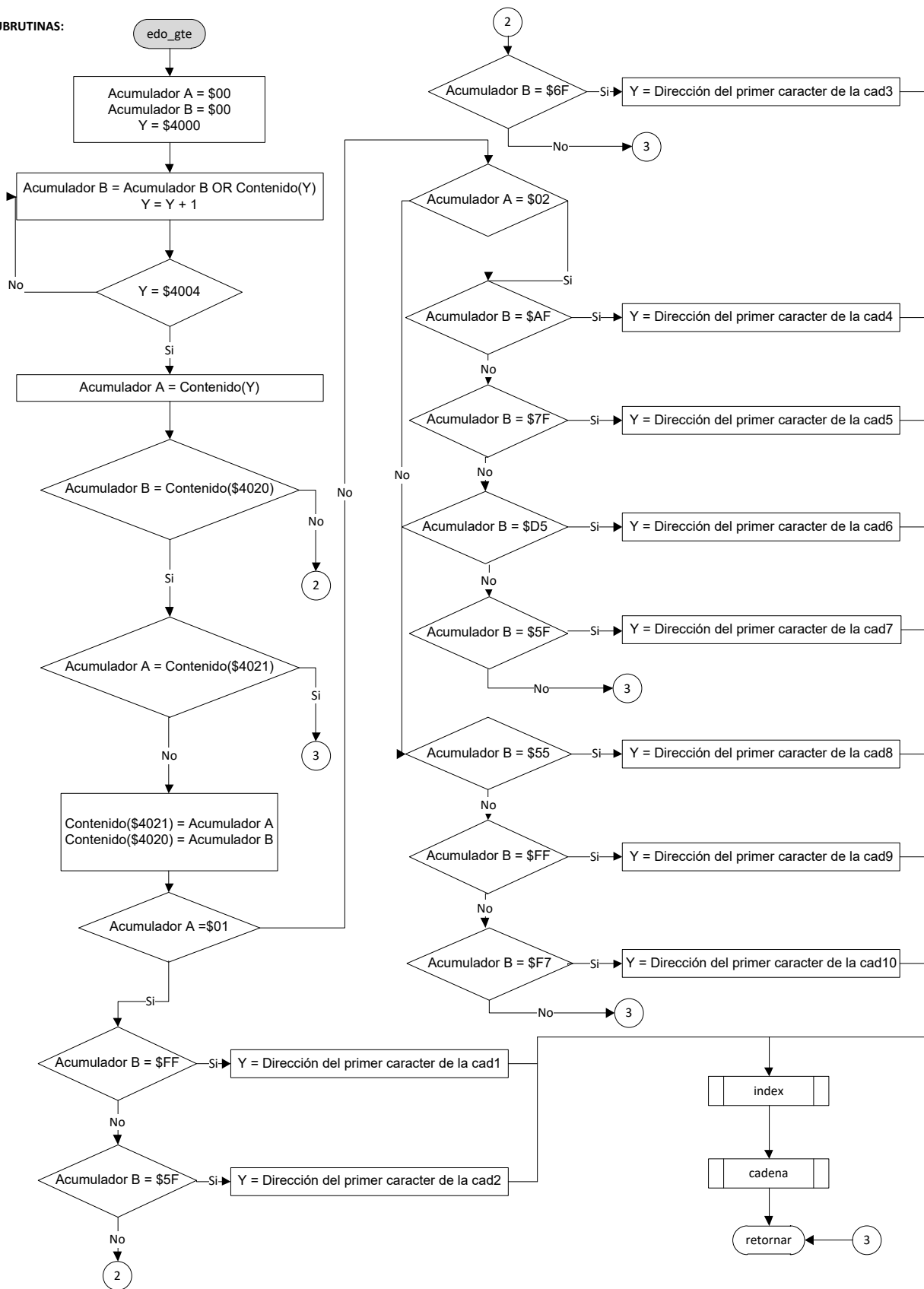




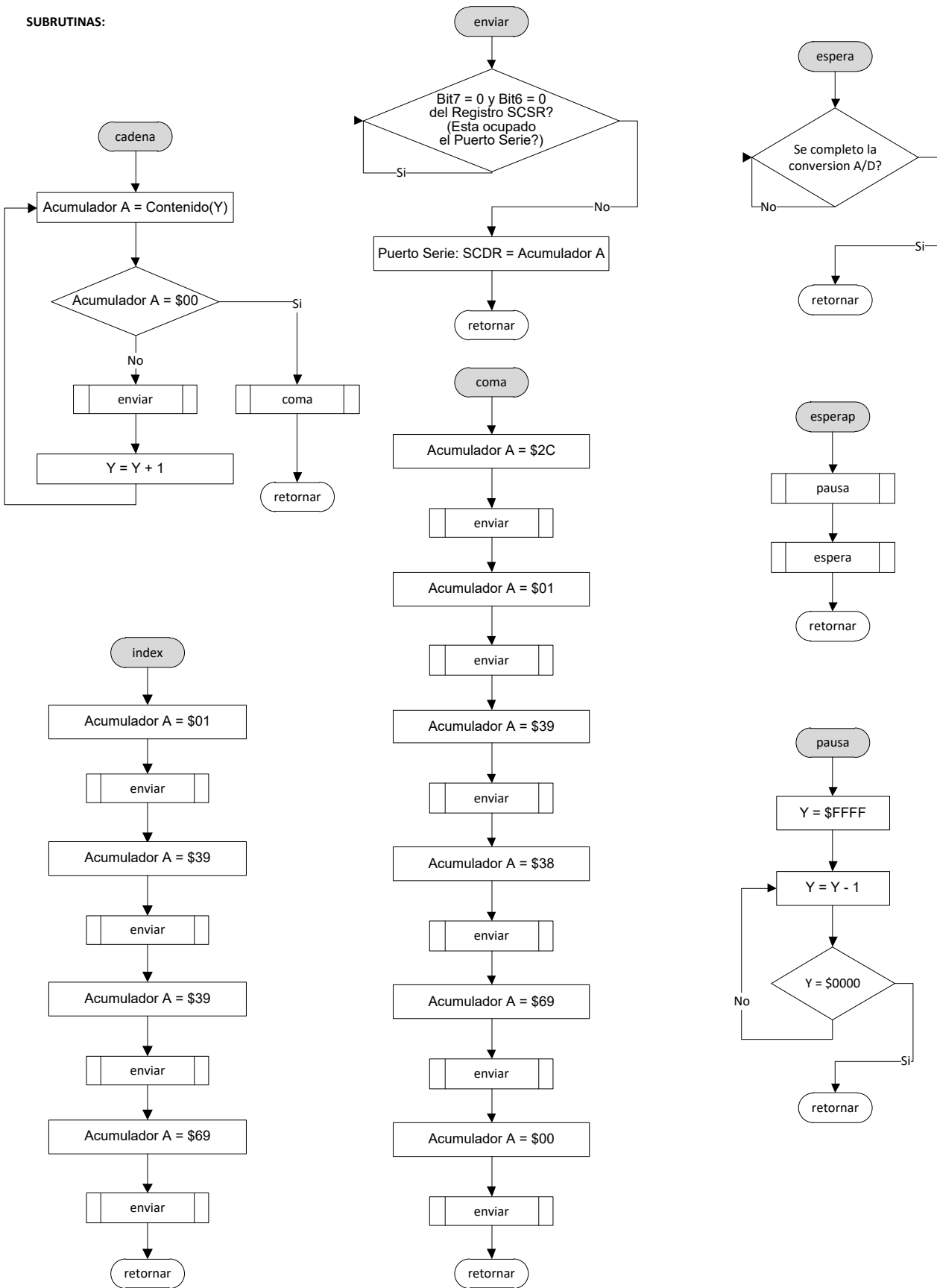
SUBROUTINAS:



SUBROUTINAS:



SUBROUTINAS:



CAPITULO 2

- [2.1] **“The art of sign language”**
Christopher Brown
Editorial Thunder Bay Press
San Diego California.
2002.
- [2.2] **“Los niños con deficiencias auditivas”**
Jean – Claude Lafon
Ed. Masson, S. A.
1ª edición 1987.
España.
- [2.3] **“Alumno Sordo Integrado. Guía para profesores de educación en general”**
Verónica de la Paz C.
Margarita Miranda J.
Lucía Rojas T.
Editorial Universitaria
3ª edición.
Septiembre de 1998.
Impreso en Chile.
- [2.4] **“Sordera. Comunicación y Aprendizaje”**
Coordinadora: Núria Silvestre Benach
Cristina Cambra
Cristina Laborda
Angels Mies
Ana Ramspott
Núria Rosich
Carrasumada Serrano
Jesús Valero
Editorial Masson, S. A.
Barcelona (España), 1998.
Impreso en España.
- [2.5] **“La Comunidad Sorda y sus derechos lingüísticos”**
Johan Cristian Cruz Cruz
Margarita Moreno – Bonett
Rosa María Álvarez González
Javier Torres Parés
Primera Edición 2014.
Impreso y hecho en México.
Universidad Nacional Autónoma de México.
- [2.6] **“Comunicación Manual”**
Ma. Esther Serafín García
Editorial Libre Acceso, A.C.
Décimo Tercera Edición, Octubre 2003.
Tomo I y Tomo II.

- [2.7] **“Comunicar a través del silencio”**
Isabel de los reyes Rodríguez Ortiz
Secretaría de Publicaciones Vicerrectorado de Investigación.
Sevilla España, 2005.
- [2.8] **“Sordera y Comunicación”**
Karlo Gajic Liska
Amparo Marant Gimeno
Ediciones Aljibe.
Edición 2010.
- [2.9] Human Anatomy & Physiology
Elaine Nicpon Marieb
Editorial Benjamin Cummings, ©2001
5th ed.

CAPITULO 3

- [3.1] **“Tecnologías del habla: Síntesis de voz”**
Javier Cuétara Priede
México, UNAM
2002
- [3.2] **“Manual de prácticas de transductores y convertidores eléctricos”**
Gloria Mata Hernández, Juan Manuel Gómez González, Ricardo Garibay Jiménez
Facultad de Ingeniería UNAM, División de Ingeniería Eléctrica, Electrónica y
Computación.
Departamento de Ingeniería de Control
1993.
- [3.3] **“Transductores y Acondicionadores de Señal”**
Ramón Pallás Areny
Marcombo Boixareu Editores
Impreso en España
13ª Edición.
1989.
- [3.4] **“Tratamiento Digital de Señales”**
Proakis, J.G y Manolakis D. G.
Editorial Prentice Hall, Madrid, 1998
3ª Edición.
- [3.5] **“Digital Speech Processing, Synthesis, and Recognition”.**
Second Edition, Revised and expanded. Sadaoki Furui. Editorial Marcel Dekker, Inc. New York.
2001.
- [3.6] **“Data Acquisition Techniques Using PCs”**
Howard Austerlitz
Editorial Academic Press
2nd ed.
Printed in the United States of America, 2003
- [3.7] **“Voice Production and Speech”**
Greta Colson
Editorial Longman, 1995.
3rd ed.

CAPITULO 4

- [4.1] **“Microcontroller Projects in C for the 8051”**
Ibrahim Dogan.
Editorial Newnes.
1ª edición.
Impreso en Gran Bretaña 2000.
- [4.2] **“Microcontroladores PIC. La solución en un CHIP”**
J. Ma. Angulo Usategui.
E. Martín Cuenca.
Angulo Martínez.
Editorial Paraninfo.
1a edición.
Impreso en España 1997.
- [4.3] **“68HC12 Microcontroller. Theory and Applications”**
Pack Daniel J.
Barreto Steven F.
Editorial Prentice Hall.
1ª edición.
Impreso en EUA 2002.
- [4.4] **“Microcontroladores 8051 y 8052”**
Bernard Odant.
Editorial Paraninfo.
España, 1995.
- [4.5] **“Microcontrolador 68HC11: Fundamentos, recursos y programación ”**
Cristina Doblado Alcázar
Juan González Gómez
Andrés Prieto-Moreno
Juan José San Martín
Microbótica.

CAPITULO 5

- [5.1] **“WTS701. WINBOND SINGLE-CHIP TEXT-TO-SPEECH PROCESSOR”**
Winbond Electronics Corporation
Publication Release Date: May 2003
Revision 3.09
- [5.2] **“DoubleTalk RC8660. CMOS, 3.3 Volt / 5 Volt. Voice Synthesizer Chipset”**
RC Systems.

CAPITULO 6

- [6.1] **“Bend Sensor Technology”**
Electronic Interface
Design Guide
Flexpoint Sensor Systems, Inc. 1997
- [6.2] **“TARJETA FACIL_11B”**
Antonio Salvá Calleja
Facultad de Ingeniería, UNAM, Abril de 2002.

- [6.3] **“Especificaciones de configuración de mapas de memoria y puertos de la tarjeta facil_11b”**
Antonio Salvá Calleja
Facultad de Ingeniería, UNAM
Abril de 2002
- [6.4] **“DoubleTalk RC8660. Evaluation Kit”**
User's Guide 1.0
RC Systems, Inc.

CAPITULO 7

- [7.1] **“Robert L. Boylestad, Louis Nashelsky”**
Prentice-Hall Hispanoamericana S.A.
Impreso en México, 1994
5a Edición
- [7.2] **“Fundamentals of Operational Amplifiers and Linear Integrated Circuits”**
Howard M. Berlin, Frank C. Getz Jr.
Merrill Publishing Company
Printed in the United States of America, 1990
- [7.3] **“Principios de Electrónica”**
Albert Paul Malvino
McGraw-Hill
6a Edición
Impreso en España, 1999
- [7.4] **“DESCRIPCIÓN DE CONECTORES Y LOCALIZACIÓN DE COMPONENTES PARA LA TARJETA FACIL_11B”**
Antonio Salvá Calleja
Facultad de Ingeniería, UNAM
Abril de 2002

REFERENCIAS DE INTERNET

CAPITULO 2

- [W2.1] **Lengua de Signos**
<http://www.proel.org/index.php?pagina=alfabetos/signos>
- [W2.2] **Alfabeto Dactilológico de los sordociegos españoles**
<http://www.asocide.org/solotexto/alfabeto.htm>
- [W2.3] **Alfabeto dactilológico**
<http://descubriendoolenguaje.blogspot.mx/2008/05/alfabeto-dactilolgico.html>
- [W2.4] <http://80.24.82.195/redaccionjoven/hemeroteca/Numero12/esp/mon/alfabeto/alfabeto.html>
- [W2.5] **Movimientos articulares**
<http://moviarti.blogspot.mx/2015/06/>

CAPITULO 3

- [W3.1] **“Estado del arte en Tecnología del Habla “**
L. Hernández Gómez, F. J. Caminero Gil, UNIVERSIDAD POLITÉCNICA DE MADRID
C. de la Torre Munilla, L. Villarrubia Grande, TELEFÓNICA INVESTIGACIÓN Y DESARROLLO
<http://www.gaps.ssr.upm.es/es/investigacion/tecnologia-del-habla>
- [W3.2] **“The basic Properties of Speech”**
http://www-mobile.ecs.soton.ac.uk/speech_codecs/
- [W3.3] **“A Short Introduction to Text-to-Speech Synthesis by Thierry Dutoit”**
TTS research team, TCTS Lab.
<http://tcts.fpms.ac.be/synthesis/introtts.php>
<http://person2.sol.lu.se/JohanFrid/webapps/pmwiki/pub/files/IntroTTSDutoit.pdf>

CAPITULO 4

- [W4.1] <http://www.intel.com/>
- [W4.2] **“A Brief History of Microprocessors “**
<https://www.coursehero.com/file/p465hit/WAHID-2015-16-Term-1-A-Brief-History-of-Microprocessors-The-first/>
- [W4.3] **Hardware of the PIC16F877**
http://www.piclist.com/images/www/hobby_elec/e_pic877.htm

CAPITULO 5

- [W5.1] **El sintetizador**
<http://www.duiops.net/newage/newsinte.htm>
- [W5.2] **RC Systems. Módulo V8600A.**
<http://www.rcsys.com/v860x.htm>

[W5.3] **Vocalizer - Text-to-speech technology for more personalized interactions.**
<http://www.nuance.com/for-business/text-to-speech/vocalizer/index.htm>

[W5.4] **V8600A CMOS, 5 Volt-only Voice Synthesizer Module.**
<http://rcsys.com/Downloads/v8600a.pdf>

[W5.5] <http://www.tid.es/premios/SenalizadorBluetooth.pdf>

[W5.6] **WTS701EF Datasheet(PDF)**
<http://html.alldatasheet.com/html-pdf/91285/WINBOND/WTS701EF/499/2/WTS701EF.html>

[W5.7] **Control de un sintetizador de voz SP03 con PIC**
<http://robotypic.blogspot.mx/2011/04/control-de-un-sintetizador-de-voz-sp03.html>

[W5.8] <http://www.tid.es/premios/SenalizadorBluetooth.pdf>

[W5.9] <http://www.ScanSoft.com>

CAPITULO 6

[W6.1] **“Technical Data. MC68HC11F1”**
Freescale Semiconductor, Inc.
<http://www.nxp.com/assets/documents/data/en/data-sheets/MC68HC11F1.pdf>

CAPITULO 7

[W7.1] **“Serial port monitor software - serial port, com port and rs232 port monitor, sniffer and analyzer tool”**
<https://www.aggsoft.com/serial-port-monitor.htm>