



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA Y TECNOLOGÍA EN COMPUTACIÓN

***“SISTEMA DE CONTROL ELECTRÓNICO POR COMANDOS DE VOZ PARA
PROTOTIPO DE ROBOT COMPAÑERO.”***

TESIS DE GRADO

Previa obtención del título de:

INGENIERO EN ELECTRÓNICA Y COMPUTACIÓN

Presentada por:

JENDRY MANUEL VILLACIS SALAZAR

Riobamba – Ecuador

- 2008 –

A mí querida madre Teresa Salazar principal impulsadora del progreso y éxito alcanzado.

NOMBRE

FIRMA

FECHA

Dr. Romeo Rodríguez

DECANO DE LA FACULTAD

DE INFORMÁTICA Y ELECTRÓNICA

Ing. Paul Romero.

DIRECTOR DE LA ESCUELA DE

INGENIERÍA EN ELECTRÓNICA

Y TECNOLOGÍA EN COMPUTACIÓN

Ing. Franklin Moreno.

DIRECTOR DE TESIS

Ing. Hugo Moreno.

MIEMBRO DEL TRIBUNAL

Lcdo. Carlos Rodríguez

DIRECTOR

DEPARTAMENTO DE DOCUMENTACIÓN

NOTA DE LA TESIS:

“Yo, JENDRY MANUEL VILLACIS SALAZAR, soy responsable de las ideas, doctrinas y resultados expuestos en esta tesis, y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica De Chimborazo”

Jendry Manuel Villacís Salazar

ÍNDICE DE ABREVIATURAS

A	Amperio
CC	Corriente Continua
CI	Circuito Integrado
DTW	Alineamiento Temporal Dinámico o Dynamic Time Warping
Fig.	Figura o gráfico
I/O	Input output (entradas y salidas)
MOM	Modelos Ocultos De Markov
PCM	Pulse Code Modulation (Modulación por Código de pulsos)
PLL	phase lock loop
RNA	Redes Neuronales Artificiales
RSC-4x RPM	Rapid Prototyping Module (Módulo de reconocimiento de voz)
R.U.R.	Robots Universales de Rossum
SI	Speaker Independent (Reconocimiento Independiente del locutor)
SD	Speaker Dependent (Reconocimiento Dependiente del locutor)
SVWS	Speaker Verification Word Spot (reconocimiento y verificación de palabras)
V	Voltios
Vocoder	Voice Coder (Codificación de la voz)
-XM	Program Run (Ejecución del programa)

ÍNDICE GENERAL

PORTADA

DEDICATORIA

FIRMAS DE RESPONSABILIDAD

RESPONSABILIDAD DEL AUTOR

ÍNDICE DE ABREVIATURAS

ÍNDICE GENERAL

INTRODUCCIÓN

CAPITULO I

NOCIONES TEÓRICAS SOBRE EL RECONOCIMIENTO DE VOZ	15
1.1 CONCEPTO	15
1.2 FORMACIÓN DE LA VOZ	15
1.3 FONOLOGÍA Y FONÉTICA	16
1.3.1 ELEMENTOS DEL APARATO FONATORIO	17
1.3.2 ARTICULACIÓN DE LOS SONIDOS VOCALES	18
1.4 CARACTERÍSTICAS DEL SONIDO	21
1.4.1 INTENSIDAD	21
1.4.2 TONO	21
1.4.3 TIMBRE	21
1.5 CLASIFICACIÓN DE LOS SONIDOS DE VOZ	22
1.5.1 VOCALES Y CONSONANTES	22
1.5.2 ORALIDAD Y NASALIDAD	22
1.5.3 TONALIDAD	23
1.5.4 LUGAR Y MODO DE ARTICULACIÓN (CONSONANTES)	23

1.5.5 POSICIÓN DE LOS ÓRGANOS ARTICULATORIOS (VOCALES)	24
1.5.6 DURACIÓN	25
1.5.7 ALFABETO FONÉTICO INTERNACIONAL	25
1.5.8 FRECUENCIA DE UTILIZACIÓN DE LOS FONEMAS EN EL CASTELLANO	26
1.5.9 CARACTERÍSTICAS ACÚSTICAS DE LA IDENTIDAD DEL LOCUTOR	27
1.6 DIGITALIZACIÓN DE LA SEÑAL DE VOZ	28
1.6.1 CODIFICACIÓN DE VOZ	29
1.7 TÉCNICAS PARA EL RECONOCIMIENTO	32
1.7.1 ALINEAMIENTO TEMPORAL DINÁMICO O DYNAMIC TIME WARPING (DTW)	32
1.7.2 INTRODUCCIÓN A LOS MODELOS OCULTOS DE MARKOV (MOM)	39
1.7.3 REDES NEURONALES ARTIFICIALES	44

CAPITULO II

ROBÓTICA	49
2.1 CONCEPTO DE ROBOT	49
2.2 CLASIFICACIÓN DE LOS ROBOTS	50
2.2.1. ROBOTS PLAY-BACK.	50
2.2.2. ROBOTS CONTROLADOS POR SENSORES	50
2.2.3. ROBOTS CONTROLADOS POR VISIÓN	50
2.2.4. ROBOTS CONTROLADOS ADAPTABLEMENTE	51
2.2.5. ROBOTS CON INTELIGENCIA ARTIFICIAL	51
2.4 ROBOTS CON VEHÍCULOS DE RODADO "ROVERS"	53
2.5 COMPONENTES DE LOS ROBOTS	54
2.5.1 SENSORES	54
2.5.2 INTERFACES FÍSICAS.	57
2.5.3 TRANSISTORES.	58

2.5.4 RELÉS.	59
2.5.5 MOVIMIENTO.	60
2.5.5 ENGRANAJES	63

CAPITULO III

MICROCONTROLADORES Y MODULO DE RECONOCIMIENTO DE VOZ	66
3.1 ¿QUE ES UN MICROCONTROLADOR?	66
3.1.2 ¿DIFERENCIA ENTRE MICROPROCESADORES Y MICROCONTROLADORES?	67
3.1.3 CARACTERISTICAS DE PIC'S UTILIZADOS	68
3.1.4 MICROCONTROLADOR RSC-4128	74
3.2 ESTUDIO MÓDULO DE RECONOCIMIENTO DE VOZ	75
3.2.1 MODULO DE RECONOCIMIENTO DE VOZ VR STAMP.	76

CAPITULO IV

APLICACIÓN Y DESARROLLO	82
4.1 FUNCIONAMIENTO GENERAL	82
4.2 HARDWARE DEL ROBOT	83
4.2.1 SISTEMA DE ENGRANAJES:	84
4.3 REQUERIMIENTOS ELÉCTRICOS Y ELECTRÓNICOS	85
4.3.1 DISEÑO DE LA PLACA PRINCIPAL DE CONTROL	86
4.3.2 MODULO CONTROL MOTORES	86
4.3.3 TRANSMISIÓN Y RECEPCIÓN INALÁMBRICA	87
4.3.4 CIRCUITO PARA EL CONTROL DEL CUERPO DEL ROBOT	88
4.3.4 SENSORES DE OBSTÁCULO	89
4.4 PROGRAMA DE APLICACIÓN	89

4.4.1 REQUERIMIENTOS PARA PROGRAMAR VR STAMP	90
4.4.2 PROGRAMACIÓN PIC16F877A	109
4.2 EVALUACIÓN DEL MÓDULO RECONOCIMIENTO DE VOZ	114
4.2.1 EVALUACIÓN EN AMBIENTES IDEALES	114
4.2.2 EVALUACIÓN EN AMBIENTES CON RUIDO MODERADO	114
4.2.3 EVALUACIÓN EN AMBIENTES CON RUIDO ALTO	115
CONCLUSIONES	
RECOMENDACIONES	
RESUMEN	
SUMARY	
GLOSARIO	
ANEXOS	
BIBLIOGRAFÍA	

ÍNDICE DE FIGURAS

FIG. I.1: COMUNICACIÓN ORAL	16
FIG. I.2: APARATO FONATORIO.....	17
FIG. I.3: CORTE ESQUEMÁTICO DE LA LARINGE EN PLANO HORIZONTAL	18
FIG. I.4: FACES DE LA REPRESENTACIÓN DEL SONIDO	19
FIG. I.5: CLASIFICACIÓN DE LOS CODIFICADORES DE VOZ.	30
FIG. I.6: COMPARACIÓN ENTRE LA DISTANCIA EUCLIDEANA Y EL ALINEAMIENTO TEMPORAL.	33
FIG. I.7: ALINEAMIENTO TEMPORAL ENTRE DOS SERIES TEMPORALES.	33
FIG. I.8: MATRIZ DE COSTO Y DISTANCIA MÍNIMA PARA EL CAMINO DE ALINEACIÓN.	34
FIG. I.9: RESTRICCIONES DEL CAMINO GLOBAL.	36
FIG. I.10: RESTRICCIONES DEL CAMINO LOCAL.....	37
FIG. I.11: ABSTRACCIÓN DE DATOS.....	38
FIG. I.12: MODELO DE URNA Y BOLAS CON N-ESTADOS.	40
FIG. I.13: FORMA DE UNA NEURONA.	47
FIG. I.14: EJEMPLO DE UNA RED NEURONAL ARTIFICIAL.....	48
FIG. I.15: SISTEMA NEURONAL ARTIFICIAL.	48
FIG. II.16: SENSOR ULTRASÓNICO DE LA SERIE HERIAN PROFFER MODELO HE-US23.....	56
FIG. II.17: BUMPER.....	57
FIG. II.18: SENSOR DE INFRARROJO TCRT500	57
FIG. II.19: SIMBOLOGÍA TRANSISTORES NPN Y PNP	58
FIG. II.20: DESCRIPCIÓN DE PINES DEL TRANSISTOR TIP 120.....	59
FIG. II.21: DIFERENTES TIPOS DE RELÉS Y ENCAPSULADOS.	59
FIG. II.22: DIAGRAMA DE FUNCIONAMIENTO DE UN RELÉ.	60
FIG. II.23: CIRCUITO PARA DISPARAR UN RELÉ.	60
FIG. II.24: MOTORES DE CORRIENTE CONTINUA	61

FIG. II.25: DIAGRAMA DE LOS BOBINADOS DE UN MOTOR UNIPOLAR.....	62
FIG. II.26: CONEXIÓN DE LOS BOBINADOS A LOS TRANSISTORES.	62
FIG. II.27: SISTEMA DE ENGRANAJES	63
FIG. III.28: DIAGRAMA DE BLOQUE DE UN MICROCONTROLADOR	67
FIG. III.29: DIAGRAMA DE BLOQUES PIC 16F877A.....	72
FIG. III.30: PIC 16F877A ROTULACIÓN DE PINES.....	73
FIG. III.31: DIAGRAMA DE BLOQUES RSC-4128.....	75
FIG. III.32: MODULO DE RECONOCIMIENTO DE VOZ VR STAMP™	76
FIG. III.33: DIAGRAMA DE BLOQUES RSC-4X RPM	78
FIG. IV.34: ETAPAS DE IMPLEMENTACIÓN DEL ROBOT	83
FIG. IV.35: POSICIÓN DE LAS RUEDAS EN CONFIGURACIÓN ACKERMAN	84
FIG. IV.36: DIAGRAMA DE BLOQUES DEL SISTEMA ELECTRÓNICO	85
FIG. IV.37 : DIAGRAMA DE BLOQUES DEL SISTEMA ELECTRÓNICO (RECEPCIÓN)	85
FIG. IV.38: PLAQUETA PRINCIPAL DE CONTROL.....	86
FIG. IV.39: CONTROL DE MOTORES DRIVER L293D	87
FIG. IV.40: TRANSMISIÓN INALÁMBRICA.....	88
FIG. IV.41: RECEPCIÓN INALÁMBRICA	88
FIG. IV.42: MANEJO MOTOR C.C.....	89
FIG. IV.43: ESQUEMA SENSOR DE OBSTÁCULOS	89
FIG. IV.44: DIVISIÓN DE COMANDOS.....	91
FIG. IV.45: EDICIÓN DEL ARCHIVO DE AUDIO	92
FIG. IV.46: ERROR EN EL FORMATO DE ARCHIVO DE AUDIO	92
FIG. IV.47: ETIQUETADO DE LOS ARCHIVOS DE AUDIO	93
FIG. IV.48: MENSAJE DE PRECAUCIÓN.....	96
FIG. IV.49: CREACIÓN DEL VOCABULARIO.	96
FIG. IV.50: SELECCIÓN DE LA PRONUNCIACIÓN.....	97
FIG. IV.51: CONFIGURACION DE PARAMETROS.....	98
FIG. IV.52: CORRECCIÓN DE PATHS.....	101

FIG. IV.53: MENSAJE DE ERROR	102
FIG. IV.54: MENSAJE DE COMPILACION CORRECTO	102
FIG. IV.55: DIAGRAMA DE FLUJO VR STAMP.....	103
FIG. IV.56: CUADRO DE DIALOGA PARA LA CARGA DEL ARCHIVO.HEX	109
FIG. IV.57: DIAGRAMA DE FLUJO LECTURA SERIAL.....	109
FIG. IV.58: SUBROUTINA ADELANTE.....	110
FIG. IV.59: SUBROUTINA ATRÁS	111
FIG. IV.60: SUBROUTINA IZQUIERDA.....	112
FIG. 61: SUBROUTINA DERECHA.....	112
FIG. IV.62: SUBROUTINA RESET	113

ÍNDICE DE TABLAS

TABLA I.I: CLASIFICACIÓN DE LAS CONSONANTES POR EL MODO Y LUGAR DE ARTICULACIÓN	24
TABLA I.II: CLASIFICACIÓN DE LAS VOCALAS SEGÚN LA POSICIÓN DE LA LENGUA.....	25
TABLA I.III: LOS FONEMAS DEL ALFABETO FONÉTICO INTERNACIONAL UTILIZADOS EN LA LENGUA CASTELLANA	26
TABLA I.IV: DISTRIBUCIÓN PORCENTUAL DE LA FRECUENCIA DE LOS FONEMAS EN ESPAÑOL	26
TABLA III.V: CARACTERÍSTICAS DEL PIC 16F877A.....	69
TABLA III.VI: DESCRIPCIÓN DE PINES PIC 16F877A	73
TABLA III.VII: DESCRIPCIÓN PINES VR STAMP™	79
TABLA IV.VII: TABLA DE SONIDOS	93
TABLA IV.IX: DEFINICIÓN PALABRA DE INICIO E ÍNDICE ASIGNADO	99
TABLA IV.X: ASIGNACIÓN DE ÍNDICES AL VOCABULARIO	99

INTRODUCCIÓN

Las investigaciones realizadas en busca de perfeccionar el área de la robótica son innumerables, mas aún si se trata de aplicaciones que requieren elementos adicionales como implementar el reconocimiento de voz en robots para que entienda ordenes dadas por el ser humano a través del habla. Una de las aspiraciones es la perfección de la ejecución de comandos de voz para que estos sean realizados con un nivel alto de confiabilidad especialmente en tareas que requieran gran precisión, en la EIE-TC se encuentran algunas investigaciones relacionadas con este tema, que son abordadas e implementadas con distintas tecnologías pero que la confiabilidad todavía esta muy lejos de perfeccionarse es por ello que en esta aplicación final de trabajo busco controlar a un robot a través de la voz con tecnología distinta a la ya estudiada, que lleve o alcance niveles de confiabilidad mas altos de los ya presentados en anteriores trabajos de investigación, para ello se centra como la parte modular de esta tesis; los microcontroladores de la empresa Sensory Inc. que cuanta con las características ideales para este propósito, para el reconocimiento de voz debe ser programado en Micro C, conjuntamente con dos Software adicionales que complementan las librerías que son incluidas para lograr armonizar a un robot que interactúe y sirva como compañero y control demótico del hogar a través de radiofrecuencia.

La parte mecánica del robot no es abordada con profundidad en este trabajo, sin embargo no se quita la relevancia que tiene este tema. Se detallará el modo de acoplamiento que tuvo cada una de sus partes, y se describe ciertos “secretos” por así llamarlos en la programación y en todo el desarrollo de la circuitería que se refiere en cada uno de los capítulos aquí expuestos.

CAPITULO I

NOCIONES TEÓRICAS SOBRE EL RECONOCIMIENTO DE VOZ

1.1 CONCEPTO

El reconocimiento de voz es una técnica mediante la cual, se interpreta el significado de nuestras palabras, con la finalidad de tomar decisiones o ejecutar una orden.

Reconocimiento de voz es una expresión que se emplea para incluir un conjunto de aplicaciones, entre las que podemos mencionar las siguientes:

Identificación de locutor, verificación del locutor, selección del locutor, reconocimiento del habla, reconocimiento de una palabra clave.

Con el objetivo de comprender mejor el análisis de la voz y su representación digital es conveniente estudiar como se produce la señal de voz en el aparato fonador humano.

1.2 Formación de la voz

La voz se puede definir como “un conjunto amplio de sonidos articulados por los seres humanos”. La voz es el producto acústico de movimientos voluntarios y bien controlados del aparato fonador. El aire se inhala en los pulmones al expandir la caja torácica y es conducido

por la cavidad nasal, el velo, y la tráquea. Se expelle al contraer la caja torácica e incrementar la presión pulmonar. Después de ser expelido por los pulmones, el aire pasa entre dos piezas simétricas de ligamentos y músculos localizados en la laringe, dentro de la tráquea: las cuerdas vocales. Y después sale al exterior a través del tracto vocal. La figura 1.1 muestra el proceso de formación de la voz y la comunicación de emisor a receptor.

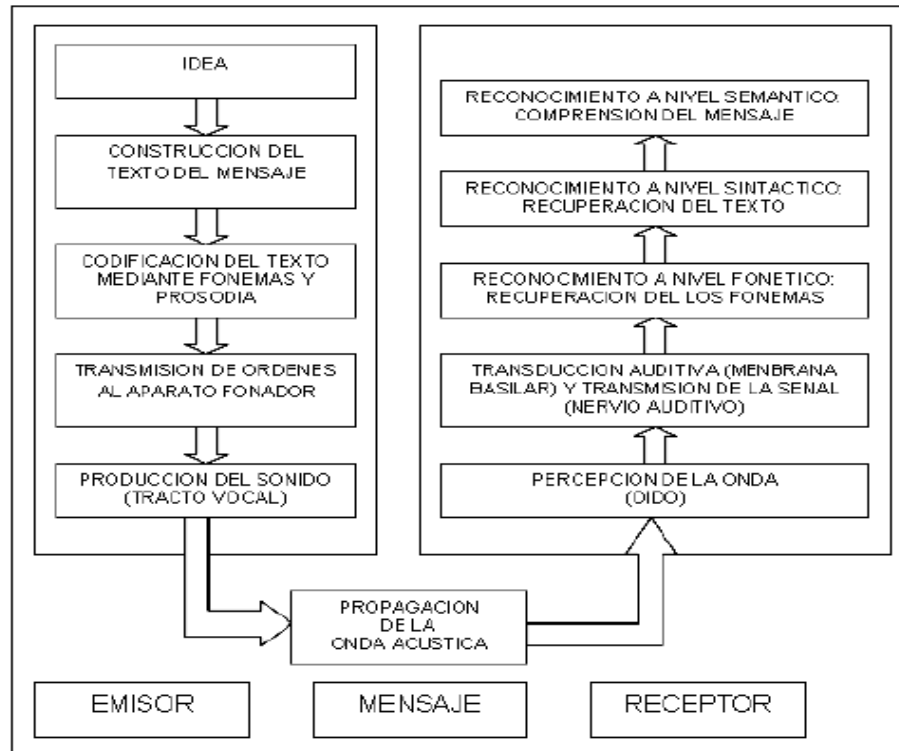


Fig. 1.1: Comunicación Oral

1.3 Fonología Y Fonética

La Fonología estudia los fonemas, es decir el modelo fónico convencional e ideal del lenguaje. La Fonética, en cambio, se refiere a los sonidos en el habla, incluyendo su producción acústica y los procesos físicos y fisiológicos de emisión y articulación involucrados. Así, la Fonología es el estudio de los sonidos de la lengua en cuanto a su carácter simbólico o de representación mental. Procede detectando regularidades o recurrencias en los sonidos del lenguaje hablado y sus combinaciones, y haciendo abstracción de las pequeñas diferencias debidas a la

individualidad de cada hablante y de características suprasegmentales como la entonación, el acento (tónico, es decir por aumento de la intensidad y agógico, por aumento de la duración), etc. Cada uno de los sonidos abstractos así identificados es un fonema. Uno de los objetivos de la fonología es reducir al máximo la cantidad de fonemas requeridos para representar cada idioma de una manera suficientemente precisa.

La Fonética estudia experimentalmente los mecanismos de producción y percepción de los sonidos utilizados en el habla a través del análisis acústico, articulatorio y perceptivo. Se ocupa, por consiguiente, de las realizaciones de los fonemas.

1.3.1 Elementos Del Aparato Fonatorio

La voz humana se produce por medio del aparato fonatorio.

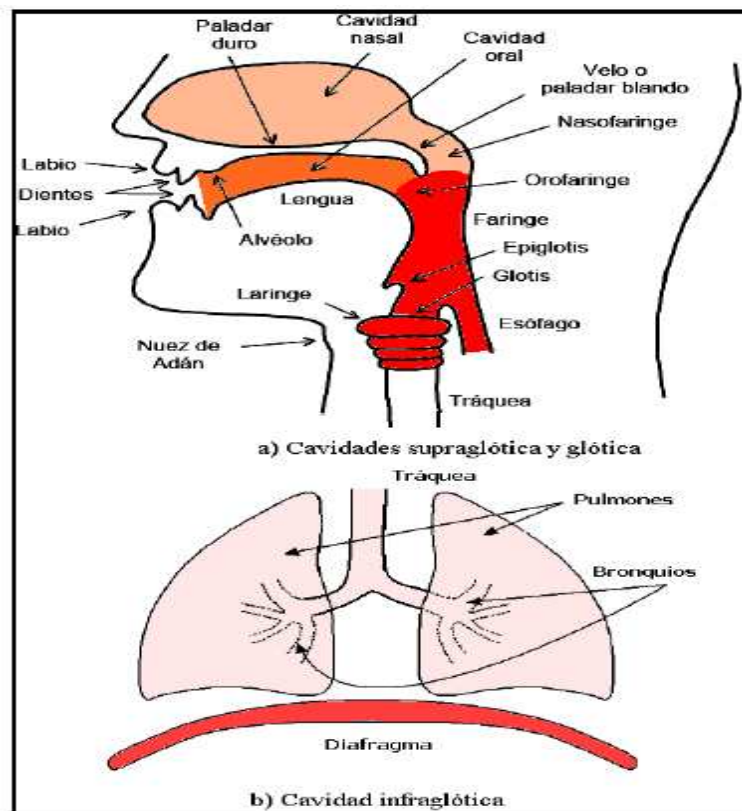


Fig. 1.2: Aparato fonatorio

Se forma por los siguientes órganos:

- Los pulmones que son la fuente de energía en forma de un flujo de aire.
- La laringe que contiene las cuerdas vocales
- La faringe
- Las cavidades oral (o bucal) y nasal.
- Elementos articulatorios: los labios, los dientes, el alvéolo, el paladar, el velo del paladar y la lengua.

En la Figura 1.2 se muestra una descripción del aparato fonatorio.

Las cuerdas vocales son dos membranas dentro de la laringe orientadas de adelante hacia atrás (Fig. 1.3). La abertura entre ambas cuerdas se denomina glotis.

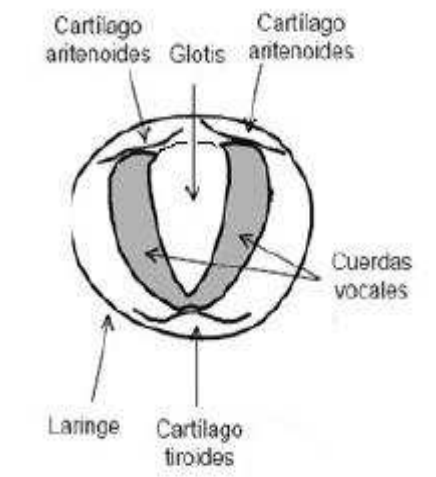


Fig. 1.3: Corte esquemático de la laringe en plano horizontal

La comunicación hablada se realiza a través del aparato respiratorio. El aparato fonatorio se divide en tres grandes partes: las cavidades infraglóticas, la cavidad glótica y las cavidades supraglóticas. La porción que incluye las cavidades faríngea, oral y nasal junto con los elementos articulatorios se denomina cavidad *supraglótica*, en tanto que los espacios por debajo de la laringe, es decir la tráquea, los bronquios, los pulmones y el diafragma se denominan cavidades *infraglóticas*.

1.3.2 Articulación de los sonidos vocales

Para la producción del habla se tienen los siguientes elementos:

- Una fuente de energía, proporcionada por el aire a presión que se expulsa en la espiración.
- Un órgano vibratorio: las cuerdas vocales.
- Una caja de resonancia: las fosas nasales, la cavidad bucal y la faringe.
- Un sistema de articulación del sonido: lengua, labios, dientes y alvéolo.

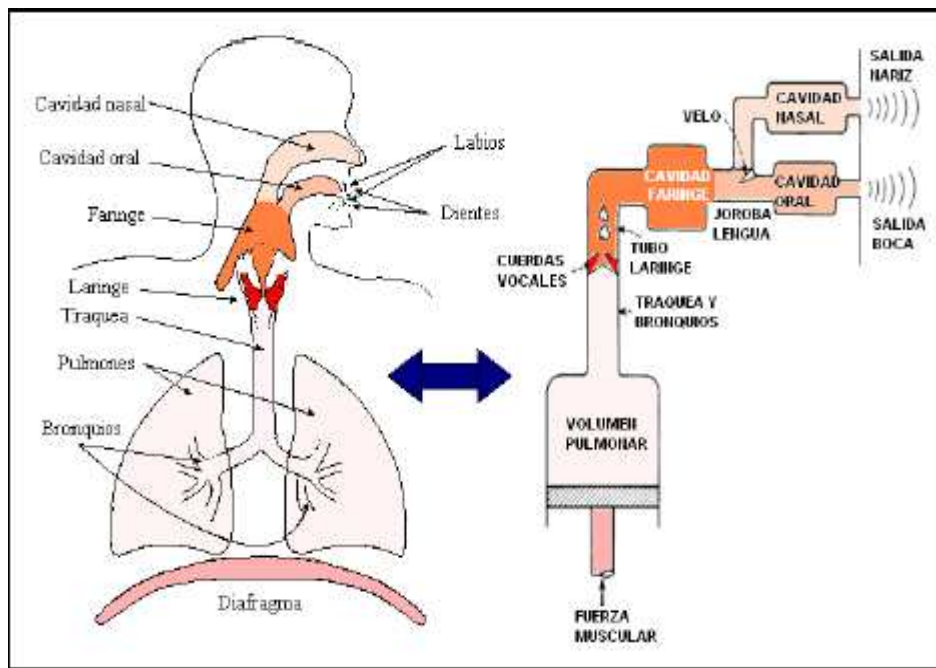


Fig. 1.4: *Fases de la representación del sonido*

Cuando las cuerdas vocales se encuentran separadas, la glotis adopta una forma triangular. El aire pasa libremente y prácticamente no se produce sonido como se observa en figura 1.4.

Es el caso de la respiración. Para producir sonido es necesario proporcionar corriente de aire, tarea desempeñada por los pulmones con ayuda del diafragma. En este proceso el aire es aspirado a los pulmones por medio de la contracción del diafragma; cuando éste se relaja se produce la espiración de aire contenido en los pulmones. Tanto los bronquios como la tráquea sirven como canales de transmisión del flujo aéreo. El aire asciende hasta llegar a la laringe, donde están las cuerdas vocales que son músculos y se oponen a manera de labios; la abertura

triangular que dejan entre sí es la glotis, por ella entra y sale el aire inspirado y espirado. Al *emitir voz*, las cuerdas vocales se juntan por contracción de los músculos adheridos en los cartílagos de la laringe y la glotis se cierra. Las cuerdas vocales comienzan a vibrar a modo de lengüetas, produciéndose un sonido tonal, es decir, periódico. La frecuencia de este sonido depende de varios factores, entre otros del tamaño y la masa de las cuerdas vocales, de la tensión que se les aplique y de la velocidad del flujo del aire proveniente de los pulmones. A mayor tamaño, menor frecuencia de vibración, lo cual explica por qué en los varones, cuya glotis es en promedio mayor que la de las mujeres, la voz es en general más grave. El tracto vocal se extiende desde la laringe hasta los labios y las ventanas nasales. Es muy similar a un tubo acústico, con una serie de resonancias, llamadas formantes y antiresonancias. La posición de los formantes queda determinada principalmente por la forma de articulación, sobre todo el movimiento de la lengua y de los labios. Las antiresonancias se producen al bajarse el velo del paladar y crearse una oclusión en el tracto bucal. El aire sale por las fosas nasales, y la cavidad bucal actúa como una cavidad acoplada. La manera como se articulan los distintos órganos permite establecer características especiales de cada individuo, como su timbre de voz. Varios de los elementos de la cavidad supraglótica se controlan a voluntad, permitiendo modificar dentro de márgenes muy amplios los sonidos producidos por las cuerdas vocales o agregar partes distintivas a los mismos, e inclusive producir sonidos propios. Todo esto se efectúa por dos mecanismos principales: el *filtrado* y la *articulación*.

El *filtrado* actúa modificando el espectro del sonido. Tiene lugar en las cuatro cavidades supraglóticas principales: la faringe, la cavidad nasal, la cavidad oral y la cavidad labial. Las mismas constituyen resonadores acústicos que enfatizan determinadas bandas frecuenciales del espectro generado por las cuerdas vocales, conduciendo al concepto de *formantes*, es decir una serie de picos de resonancia ubicados en frecuencias o bandas de frecuencia que son bastante específicas para cada tipo de sonido.

La *articulación* es una modificación principalmente a nivel temporal de los sonidos y está directamente relacionada con la emisión de los mismos y con los fenómenos transitorios que los acompañan. Está caracterizada por el tracto vocal en que tiene lugar, por los elementos que intervienen y por el modo en que se produce, factores que dan origen a una clasificación fonética. La disposición de todos los órganos se mantiene igual durante la realización de cada sonido que corresponde a un fonema, pero hay diferencias individuales. La señal mantiene ciertas características durante los intervalos correspondientes a cada sonido. Además se produce un proceso de coarticulación o acomodo hacia la articulación del siguiente sonido.

1.4 Características del sonido

Tres características que todo sonido emite: la intensidad, el tono y el timbre. Están relacionadas con diferentes propiedades de las ondas sonoras.

1.4.1 Intensidad

Se define como la energía que atraviesa por segundo una superficie unidad dispuesta perpendicularmente a la dirección de propagación. Equivale a una potencia por unidad de superficie y se expresa en W/m^2 . Para expresar intensidades sonoras se emplea una escala cuyas divisiones son potencias de diez y cuya unidad de medida es el decibelio (dB).

1.4.2 Tono

El tono es la cualidad del sonido relacionada con el oído, el cual asigna un lugar al sonido en la escala musical, permitiendo, por lo tanto, distinguir entre los sonidos graves y los agudos. La magnitud física que está asociada al tono es *la frecuencia*.

1.4.3 Timbre

El timbre es la cualidad del sonido que permite distinguir sonidos procedentes de diferentes instrumentos, aun cuando posean igual tono e intensidad. Es una cualidad para reconocer a una persona por su voz porque caracteriza a cada individuo.

1.5 Clasificación de los sonidos de voz

Existen varios criterios de clasificación que toman en cuenta aspectos del fenómeno de emisión del habla y resultan de tomar en cuenta:

- el carácter vocálico o consonántico,
- la oralidad o nasalidad,
- el carácter tonal (sonoro) o no tonal (sordo),
- el lugar de articulación,
- el modo de articulación,
- la posición de los órganos articulatorios y
- la duración

1.5.1 Vocales y consonantes

Las vocales son sonidos emitidos por la sola vibración de las cuerdas vocales sin ningún obstáculo entre la laringe y las aberturas oral y nasal. Son siempre sonidos de carácter tonal (cuasi periódicos), por consiguiente de espectro discreto. Las consonantes, por el contrario, se emiten interponiendo algún obstáculo formado por elementos articulatorios.

Los sonidos correspondientes a las consonantes pueden ser tonales o no dependiendo de la vibración de las cuerdas vocales. En el castellano las vocales pueden constituir palabras completas, no así las consonantes.

1.5.2 Oralidad Y Nasalidad

Los fonemas en los que el aire pasa por la cavidad nasal se denominan nasales, en tanto que aquéllos en los que sale por la boca se denominan orales.

En castellano son nasales sólo las consonantes “m”, “n”, “ñ”.

1.5.3 Tonalidad

Los fonemas en los que participa la vibración de las cuerdas vocales se denominan tonales o *sonoros*. La tonalidad lleva implícito un espectro cuasi periódico. Como se puntualizó anteriormente, todas las vocales son tonales, pero existen varias consonantes que también lo son: “b”, “d”, “v”, “m”, etc. Aquellos fonemas producidos sin vibraciones glotales se denominan *sordos*. Varios de ellos son el resultado de la turbulencia causada por el aire pasando a gran velocidad por un espacio reducido, como las consonantes “s”, “z”, “j”, “f”.

1.5.4 Lugar y modo de articulación (consonantes)

La articulación es el proceso mediante el cual se puede clasificar a las consonantes. Los órganos articulatorios son los labios, los dientes, las diferentes partes del paladar (alvéolo, paladar duro, paladar blando o velo), la lengua y la glotis. Salvo la glotis, que puede articular por sí misma, el resto de los órganos articula por oposición con otro.

Para cada punto de articulación ésta puede efectuarse de diferentes modos, dando lugar a fonemas:

Oclusivos: la salida del aire se cierra momentáneamente por completo.

Fricativos: el aire sale atravesando un espacio estrecho.

Africados: oclusión seguida por fricación.

Laterales: la lengua obstruye el centro de la boca y el aire sale por los lados.

Vibrantes: la lengua vibra cerrando el paso del aire intermitentemente.

Aproximantes: La obstrucción muy estrecha que no llega a producir turbulencia.

Los fonemas oclusivos también se denominan a veces explosivos o *plosivos*, debido a la liberación repentina de la presión presente inmediatamente antes de su emisión, ejemplos típicos son las letras “t” y “p”.

En la Tabla I.I se indican las consonantes clasificadas según el lugar y el modo de articulación, la sonoridad y la nasalidad. En algunos casos una misma consonante aparece en dos categorías diferentes, correspondientes a las diferencias observadas.

Tabla I.I: Clasificación de las consonantes por el modo y lugar de articulación

Lugar de articulación	Modo de articulación								
	Oral								Nasal
	Oclusiva		Fricativa		Africada	Lateral	Vibrante	Aproximante	Sonora
	Sorda	Sonora	Sorda	Sonora	Sorda	Sonora	Sonora	Sonora	
Bilabial	p	b, v		b, v				w	m
Labiodental			f						
Linguodental			z	d					
Alveolar	t	d	s	y	ch	l	r, rr		n
Palatal				(y)	(ch)	ll		ui	ñ
Velar	k	g	j	g					
Glotal			h						

1.5.5 Posición de los órganos articulatorios (vocales)

En el caso de las vocales, la articulación consiste en la modificación de la acción filtrante de los diversos resonadores, lo cual depende de las posiciones de la lengua (tanto en elevación como en profundidad o avance), de la mandíbula inferior, de los labios y del paladar blando. Estos órganos influyen sobre los formantes, permitiendo su control. Según la posición de la lengua se clasifican las vocales como se muestra en la Tabla I.II.

Es importante conocer la clasificación de las vocales de acuerdo a la posición de la lengua para detectar el sonido acústico de cada una de las vocales pronunciadas, cabe señalar que depende de las palabras formantes.

Tabla I.II: Clasificación de las vocales según la posición de la lengua

Posición vertical	Tipo de vocal	Posición horizontal (avance)		
		Anterior	Central	Posterior
Alta	Cerrada	i		u
Media	Media	e		o
Baja	Abierta		a	

Otra cualidad controlable es la *labialización*, es decir el hecho de que se haga participar activamente los labios. Las vocales labializadas, también definidas como *redondeadas*, son las que redondean los labios hacia adelante, incrementando la longitud efectiva del tracto vocal. La única vocal labializada en el castellano es la “u”.

1.5.6 Duración

La duración de los sonidos, especialmente de las vocales tiene importancia a nivel expresivo, a través del énfasis o acentuación a través de la duración. En el idioma inglés, en cambio, la duración de una vocal puede cambiar completamente el significado de la palabra que la contiene.

1.5.7 Alfabeto fonético internacional

El Alfabeto Fonético Internacional (International Phonetic Alphabet, IPA) contiene una gran cantidad de fonemas de diversos idiomas; permite representar de una manera inequívoca los fonemas independientemente del idioma. En la Tabla I.III se indica el subconjunto correspondiente al idioma castellano.

Los fonemas son esenciales para la correcta pronunciación de un determinado idioma, así por ejemplo cuando aprendemos el idioma inglés, no sabemos si la pronunciación realizada es la correcta, para ello se han estudiado los fonemas, de esta forma se puede verificar si la pronunciación fue la correcta o no.

Tabla I.III: Los fonemas del alfabeto fonético internacional utilizados en la lengua castellana

Fonemas castellanos					
Sonido	Ejemplo	Sonido	Ejemplo	Sonido	Ejemplo
[p]	paso	[θ]	zorzal, lápiz	[ɲ]	mañana, ñoño
[b]	base, vena	[s]	solo, cosa	[dʒ]	yo, Yapeyú
[β]	labor, lavar	[x]	giro, jarabe	[j]	bien, biólogo
[t]	tres, canto	[tʃ]	hecho, Chubut	[w]	hueso, buitre
[d]	dama, andar	[r]	arder, jarabe	Vocales	
[ð]	cedro, verdad	[rr]	perro, rojo	[a]	cama
[k]	caso, disco	[l]	loable, fiel	[e]	espera, ver
[g]	gula, goma	[λ]	llanto, calle	[i]	vine, iris
[ɣ]	agua, negro	[m]	mamá, ámbar	[o]	loro, pos
[f]	fino, tifón	[n]	nene, joven	[u]	burla, huracán

1.5.8 Frecuencia de utilización de los fonemas en el castellano

El castellano es un idioma cuya escritura es eminentemente fonética, ya que salvo pocos casos, hay correspondencia entre grafema y fonema. No todos los idiomas tienen esta característica.

Desde el punto de vista de la teoría fonológica, Emilio Alarcos Llorach expone: “La frecuencia con que los sonidos se repiten en la cadena hablada caracteriza, sin duda, fonéticamente una lengua dada. Igualmente, la frecuencia con que los fonemas aparecen en la elocución con valor distintivo contribuye a caracterizar fonológicamente tal lengua, mostrando el rendimiento funcional de cada uno de ellos”. En un estudio realizado por el mismo autor, se presentan los resultados en la Tabla I.IV. Cabe resaltar que este estudio se realizó con muestras del español peninsular.

Tabla I.IV: Distribución porcentual de la frecuencia de los fonemas en español

/a/	13,7	/f/	1
/e/	12,6	/s/	8
/i/	8,6	/x/	0,7
/o/	10,3	/z/	1,7
/u/	2,1	/ch/	0,4
/p/	2,1	/y/	0,4
/t/	4,6	/m/	2,5
/k/	3,8	/n/	2,7
/b/	2	/ñ/	0,2
/d/	4	/r/	0,5
/g/	1	/rr/	2,5
		/l/	4,7

En este análisis, las vocales representan un 46.23% del total de los fonemas y las consonantes el 53.77% restante, o sea, siempre un poco menos de la mitad del universo para las vocales.

Para el español latinoamericano no se presentan variaciones de importancia, salvo que se manifiesta un porcentaje mayor de utilización en la vocal /e/, los demás datos guardan estrecha similitud.

La vocal, y a la vez el fonema, más frecuente, es la /a/; La más frecuente de las consonantes obstruyentes, e incluso de todas las consonantes, es la /s/.

1.5.9 Características acústicas de la identidad del locutor

Haciendo mención al tema, H. Kuwabara manifiesta: “Los factores que son relevantes para caracterizar la identidad del locutor se pueden categorizar en términos socio-psicológicos y en términos fisiológicos”. Las propiedades anatómicas de los órganos de fonación son las responsables de la frecuencia fundamental y de las características espectrales de la voz que corresponden a un análisis de *bajo nivel*. Nuestro entorno, educación o estado de ánimo caracterizan el acento, la curva entonativa, la duración de las palabras, ritmo, pausado, niveles de potencia, contribuyen a “modelar” la individualidad e identidad del locutor en un análisis de *alto nivel*.

Las investigaciones sobre la identidad del locutor abarcan estudios psicológicos y fonéticos que demuestran las relaciones entre los parámetros acústicos y las características del locutor como son: edad, sexo, altura, peso y otras propiedades físicas, hasta los estudios más recientes que intentan aclarar qué parámetros acústicos son relevantes en lo que a la identidad del locutor se refiere. Existen investigaciones sobre la contribución del tono, la frecuencia de los formantes, la envolvente espectral y otros parámetros acústicos para varios locutores pronunciando vocales, concluyendo que, en cuanto a la identidad del locutor se refiere, el parámetro más importante era el tono o seguido de la frecuencia de los formantes, la

fluctuación y la inclinación espectral de la fuente glotal. Dentro de un *mismo género* los parámetros más significativos para distinguir locutores son los relacionados con el tracto vocal. Los estudios a los cuales H. Kuwabara en hace mención determinaron que los parámetros más influyentes en la identidad del locutor son:

Fuente vocal:

- El tono medio
- El patrón tiempo-frecuencia del tono (curva de tono).
- La fluctuación de la frecuencia fundamental.
- La fluctuación de la energía de la fuente glotal.
- La forma de onda glotal.

Tracto vocal:

- La longitud del tracto vocal.
- La forma de la envolvente espectral y la inclinación espectral.
- Las diferencias relativas en la amplitud de los formantes.
- Los valores absolutos de las frecuencias de los formantes.
- El patrón tiempo-frecuencia de las frecuencias de los formantes (trayectorias de los formantes).
- El ancho de banda de los formantes.

Las cualidades de la voz, en resumen, están representadas en varios parámetros cuyo orden de importancia puede variar de hablante a hablante e incluso dependiendo del tipo de habla que se esté analizando. Las características relativas a la *fente vocal* poseen una gran variación incluso para el mismo locutor, donde se manifiestan los efectos en su estado de ánimo; en cambio son menos variables las características del *tracto vocal*.

1.6 Digitalización de la señal de voz

Para realizar la digitalización de la señal de voz, primero se debe realizar la conversión de la señal acústica a una señal de tipo eléctrico, esto se logra mediante un transductor, que para nuestro caso, es un micrófono, y finalmente se realiza la conversión de una señal analógica a una digital. Para recuperar la señal inicial a partir de la señal digital, se debe realizar la conversión de la señal digital a señal analógica, y la conversión de señal analógica a señal acústica mediante un parlante.

La conversión análogo-digital requiere de un muestreo de la señal cada cierto intervalo de tiempo, este intervalo de tiempo se lo conoce como periodo de muestreo, y su inverso como frecuencia de muestreo. Es importante tomar una frecuencia de muestreo adecuada, para evitar fenómenos de aliasing, es decir, componentes de alta frecuencia presentes en la banda de interés. El teorema de Nyquist indica que para no tener problemas con fenómenos de aliasing, se debe muestrear como mínimo al doble del ancho de banda de la señal.

Es recomendable, antes de muestrear la señal pasarla por un filtro pasa bajos antialiasing para atenuar las componentes de alta frecuencia de la señal o el ruido, de manera que limite el ancho de banda de la señal analógica. También se debe muestrear la señal filtrada a una frecuencia ligeramente mayor al doble del ancho de banda de la señal.

1.6.1 Codificación de Voz

La codificación de voz busca reducir la cantidad de bits que se necesitan para almacenar o transmitir una señal de voz, en otras palabras se trata de eliminar la redundancia de la señal, de manera que la señal comprimida con respecto a la señal original sea lo más parecida posible. Es decir, la inteligibilidad del mensaje no debe perderse, además debe existir un compromiso entre la calidad de la señal y la tasa de comprensión. Los codificadores de voz se pueden clasificar en tres grupos: codificadores de onda, codificadores paramétricos y codificadores híbridos.

En la figura 1.5, se puede apreciar la clasificación de los codificadores de onda y los paramétricos.

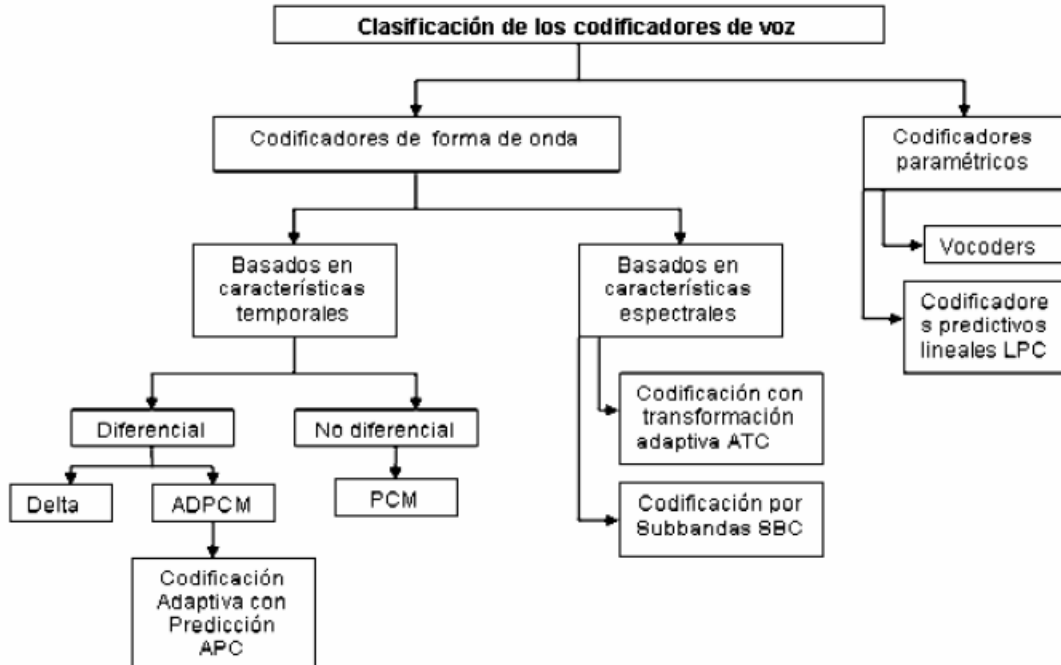


Fig. 1.5: Clasificación de los codificadores de voz.

1.6.1.1 Codificadores de forma de onda

Los codificadores de forma de onda intentan reproducir la forma de la onda de la señal de entrada. Generalmente se diseñan para ser independientes de la señal, de tal forma que pueden ser usados para codificar una gran variedad de señales.

Presentan una degradación aceptable en presencia de ruido y errores de transmisión. Sin embargo, para que sean efectivos, sólo se deben usar a tasas de muestreo medias. La codificación se puede llevar a cabo tanto en el dominio del tiempo como de la frecuencia.

1.6.1.2 Codificadores en el dominio del tiempo

Los codificadores en el dominio del tiempo se caracterizan por ser: sencillos de implementar; presentan bajo retardo; no utilizan un modelo específico de la señal de voz y por tanto son

muy generales y trabajan a regímenes binarios en torno a 32 kbps. Dentro de este grupo tenemos los siguientes codificadores: PCM¹, DPCM² y ADPCM³.

1.6.1.3 Codificación en el dominio de la frecuencia

Este tipo de codificadores dividen la señal en distintas componentes en frecuencia y codifican cada una de éstas de forma independiente. El número de bits usados para codificar cada componente en frecuencia puede variar dinámicamente.

1.6.1.4 Codificadores Paramétricos

Los codificadores de forma de onda no tienen en cuenta la naturaleza de la señal a codificar. Sin embargo, si codificamos una señal de voz, podemos aprovechar sus características intrínsecas para que la codificación se realice de forma más eficiente. Así, los vocoders⁴ (Voice coders) suponen un modelo digital de producción de voz.

Los vocoders intentan producir una señal que suene como la voz original, independientemente de si la forma de onda se parece o no. En el transmisor se analiza la voz y se extraen los parámetros del modelo y la excitación. Esta información se envía al receptor donde se sintetiza la voz. El resultado es que se produce voz inteligible a muy baja velocidad de muestreo, pero tiene el problema de que no suena natural.

1.6.1.5 Codificadores Híbridos

En la codificación híbrida se combinan las técnicas de los codificadores de forma de onda con las de los vocoders con el propósito de obtener una alta calidad de voz a baja tasa de muestreo

¹ PCM: Pulse Code Modulation

² DPCM: Differential Pulse Code Modulation

³ ADPCM: Adaptive Differential Pulse Code Modulation

⁴ Vocoder: Voice Coder

(inferior a 8 Kbps). En estos codificadores, las muestras de la señal de entrada se dividen en bloques de muestras (vectores) que son procesados como si fueran uno solo. Llevan a cabo una representación paramétrica de la señal de voz para tratar de que la señal sintética se parezca lo más posible a la original. También se les conoce como codificadores de análisis por síntesis. En el emisor se lleva a cabo un análisis que obtiene los parámetros de la señal para luego sintetizarla y conseguir el mayor parecido a la original.

1.7 Técnicas para el reconocimiento

Hasta ahora hemos estudiado como se produce la voz y en base a estos conocimientos se realiza un estudio de las técnicas utilizadas para el reconocimiento de voz, cada una de las cuales presenta sus ventajas y desventajas, aquí se presentan algunas de ellas que son utilizadas en esta tesis.

1.7.1 Alineamiento Temporal Dinámico O Dynamic Time Warping (DTW)

El alineamiento temporal dinámico, es un método que permite alinear temporalmente, las características de la palabra que se quiere reconocer con las características de la palabra de referencia, mediante programación dinámica. Esto se debe a que la duración temporal de la pronunciación de una palabra, no siempre es la misma. Por este motivo, antes de realizar la comparación, se debe alinear previamente las características de la palabra de prueba, con los modelos de palabras obtenidos en la fase de entrenamiento.

Para aplicar DTW, se debe dividir a cada palabra de referencia en tramas, y aplicar a éstas un análisis como por ejemplo LPC (Lineal Predictive Coding), de esta manera podemos modelar cada palabra como una secuencia de vectores de características. Igual procedimiento se debe aplicar a la palabra de prueba, en las mismas condiciones que se realizaron para las palabras de referencia. Luego se hace coincidir la secuencia de vectores de características, de la palabra

de prueba con la palabra de referencia, de modo que éstas queden alineadas en tiempo lo mejor que se pueda lograr.

1.7.1.1 Descripción del Alineamiento Temporal Dinámico.

Una medida de la distancia entre dos vectores de características, permite determinar la similitud o diferencia entre los dos vectores. La distancia euclideana, es una medida que se puede utilizar para este efecto.

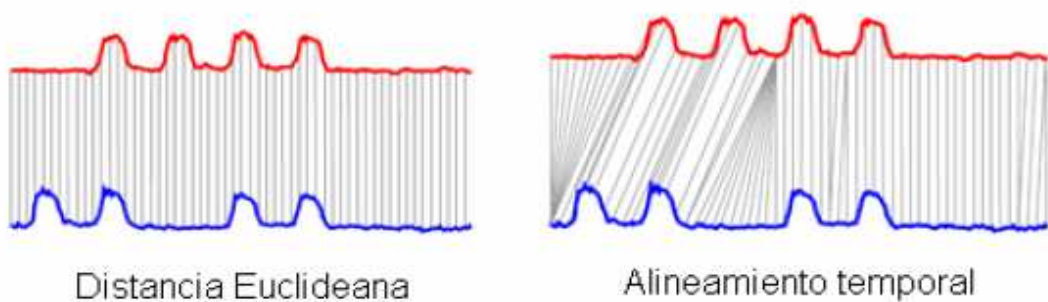


Fig. 1.6: Comparación entre la distancia euclideana y el alineamiento temporal.

La limitación de utilizar la distancia Euclideana, es por ejemplo, en el caso de tener dos vectores de características idénticas, pero uno desplazado ligeramente en el eje del tiempo del otro, la distancia Euclideana, establecería que un vector es diferente del otro, esto se debe a que la comparación se realiza punto por punto. El alineamiento temporal permite vencer la limitación de la distancia euclideana. En la figura 1.6, se puede apreciar la comparación entre la distancia Euclideana y el alineamiento temporal.

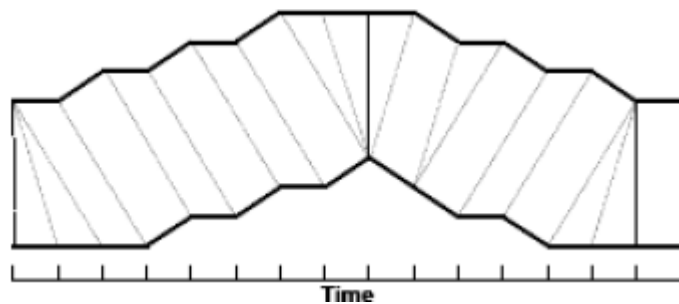


Fig. 1.7: Alineamiento temporal entre dos series temporales.

Supongamos que tenemos dos vectores de características X y Y (donde el un vector corresponde a las características de la palabra de prueba y el otro vector a las características de la palabra de referencia) cada una de longitud m y n respectivamente, en donde.

$$X = x_1, x_2, \dots, x_i, \dots, x_m \quad Y = y_1, y_2, \dots, y_j, \dots, y_n$$

Para alinear las dos secuencias usando el alineamiento temporal dinámico, debemos construir una matriz de dimensiones n por m , donde el (i, j) elemento de la matriz contiene la distancia $d(x_i, y_j)$ entre los puntos x_i y y_j (puede ser la distancia euclídeana). Cada elemento de la matriz (i, j) corresponde a la alineación entre el punto x_i y y_j . Esto se puede apreciar en la figura 1.8. Un camino de alineación W , es un conjunto continuo de elementos de la matriz que definen una trayectoria entre X y Y . El k mo elemento es definido como $W_k = (i, j)$, en donde se

$$W = w_1, w_2, \dots, w_K \quad \max(m, n) \leq K < m + n - 1$$

tiene:

El camino de alineación está sujeto a varias restricciones.

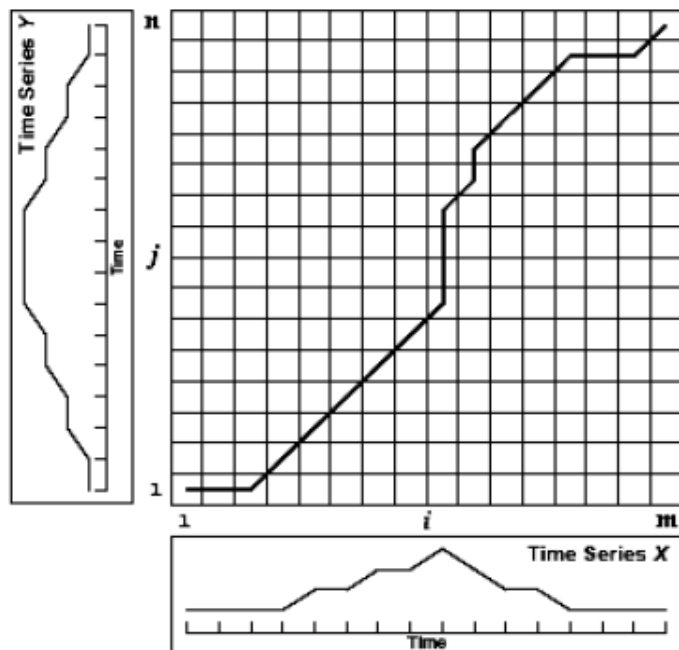


Fig. 1.8: Matriz de costo y distancia mínima para el camino de alineación.

El camino de alineación debe empezar en el inicio de cada vector en $w_1 = (1,1)$ y termina en el final de los dos vectores en $w_k = (m, n)$. Esto exige que el camino de alineación empiece y termine diagonalmente entre las dos esquinas opuestas de la matriz.

Dado el elemento $w_k = (a,b)$ entonces $w_{k-1} = (a',b')$ donde $a-a' \leq 1$ y $b-b' \leq 1$. Esto restringe los pasos permitidos a las celdas adyacentes para el camino de alineación.

Dado $w_k = (a,b)$ entonces $w_{k-1} = (a',b')$ donde $a-a' \geq 0$ y $b-b' \geq 0$. Esto obliga a que los puntos en W estén monótonamente espaciados en el tiempo.

Existen gran cantidad de caminos de alineación que cumplen las condiciones mencionadas anteriormente. Sin embargo, el camino de interés es aquel que minimiza el costo del alineamiento. Este camino puede ser encontrado usando la programación dinámica, este método permite evaluar la siguiente iteración la cual define la distancia acumulada $\gamma(i,j)$ como la distancia $d(i,j)$ encontrada en la actual celda y la mínima de las distancias acumuladas para los elementos adyacentes.

$$\gamma(i, j) = d(x_i, y_j) + \min [\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)]$$

Si los vectores de características X y Y , son idénticos, el camino de alineación que pasa por la matriz es una línea recta diagonal.

1.7.1.2 Complejidad del Alineamiento Temporal Dinámico

La complejidad en tiempo y espacio del alineamiento temporal es fácil de determinar. Cada celda perteneciente a la matriz m por n , se llena exactamente una vez, y cada celda se llena a un tiempo constante. El rendimiento a causa de la complejidad en tiempo y espacio, es $O(n^2)$, si $m=n$. La complejidad espacial de tipo cuadrático, es particularmente prohibitiva porque los requerimientos de memoria están en el orden de los terabytes, para series temporales que contienen solo 177,000 medidas. Una complejidad espacial de tipo lineal, es posible

implementar en el algoritmo del alineamiento temporal, solo si se guarda la columna actual y la anterior en la memoria, tal como la matriz de costo se llena de izquierda a derecha. Solo reteniendo dos columnas en cualquier tiempo, se puede determinar la mejor distancia de alineamiento entre dos series temporales. Sin embargo, no es posible reconstruir el camino de alineamiento entre las dos series temporales porque la información requerida para calcular el camino de alineamiento es echado fuera al descartar las columnas. Por este motivo, no es posible implementar en el algoritmo de alineamiento temporal, una complejidad de tipo lineal en aplicaciones que se requiere conocer el camino de alineamiento.

1.7.1.3 Métodos para mejorar el rendimiento del alineamiento temporal dinámico.

La complejidad del alineamiento temporal, crea la necesidad de encontrar métodos que permitan acelerar el algoritmo de alineamiento temporal. Los métodos para hacer el algoritmo de alineamiento temporal más rápido son los siguientes:

1.7.1.3.1 Restricciones

Las restricciones son ampliamente utilizadas para acelerar el alineamiento temporal. Dos de las restricciones que se usan comúnmente son la banda de Sakoe-Chuba y el paralelogramo de Itakura, los cuales se muestran en la figura I.9.

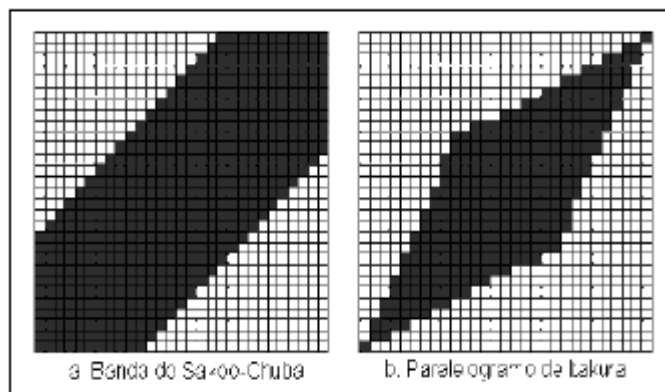


Fig. I.9: Restricciones del camino global.

Las restricciones globales limitan la cantidad máxima de expansión o comprensión, de manera que no todos los puntos de la matriz están permitidos, de esta forma se reduce el número de puntos a buscar. Esto permite reducir la complejidad computacional. Las áreas sombreadas de la figura 1.9, son las celdas de la matriz de costo que son llenadas por el algoritmo de alineamiento temporal para cada tipo de restricción.

Cuando se usan las restricciones, el algoritmo de alineamiento temporal encuentra el mejor camino de alineamiento, en la ventana. Sin embargo, el camino de alineamiento global, no será encontrado si éste no está enteramente dentro de la ventana. Las restricciones trabajan bien en dominios donde el óptimo camino de alineación se asemeja a un alineamiento lineal, y pasa por la matriz diagonalmente en una relativa línea recta. Las restricciones trabajan pobremente, si las series temporales son de eventos que empiezan y terminan en tiempos radicalmente diferentes, porque el camino de alineación puede desviarse muy lejos de una alineación lineal y casi toda la matriz de costo, debe evaluarse para encontrar el óptimo camino de alineamiento.

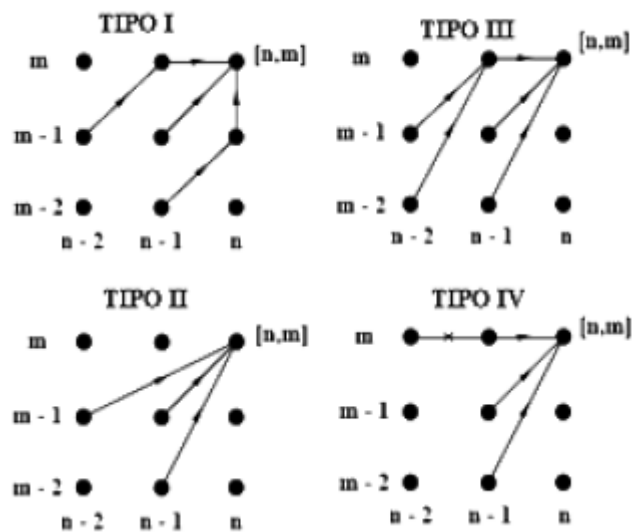


Fig. 1.10: Restricciones del camino local.

Las restricciones locales permiten limitar, el camino en el vecindario de un nodo, de la matriz de costo. En la figura I.10, se indica algunos ejemplos de restricción del camino local.

1.7.1.3.2 Abstracción de datos

La abstracción de datos permite acelerar el algoritmo de alineamiento temporal, el cual es ejecutado sobre una representación de datos reducida. En el lado izquierdo de la figura I.11, se puede observar una matriz completa para la cual se debe encontrar la mínima distancia del camino de alineamiento. En lugar de ejecutar el algoritmo de alineamiento temporal sobre toda la matriz, las series temporales son reducidas en tamaño, para hacer el número de celdas de la matriz más manejables. El camino de alineamiento es encontrado para series temporales de menor resolución, y es trazado para la matriz completa.

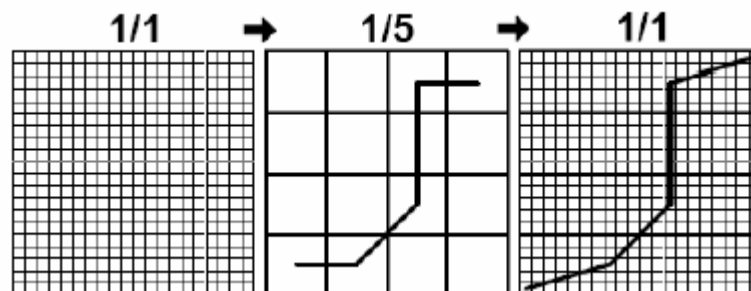


Fig. I.11: Abstracción de datos.

Al aplicar la abstracción de datos, el algoritmo de alineamiento temporal es acelerado por un factor constante grande, pero la complejidad en tiempo y espacio se mantiene. Además la distancia del camino de alineamiento que es calculado entre las dos series temporales incrementa su inexactitud, cuando el nivel de abstracción de los datos se incrementa. Proyectar el camino de alineación de baja resolución en una matriz de resolución completa, crea un camino de alineación que está lejos del óptimo, aún si el óptimo camino de alineación realmente pasa por las celdas de baja resolución, la proyección del camino de alineación en resoluciones superiores, ignora las variaciones locales del camino de alineamiento.

Esto puede ser muy significativo en la abstracción de de datos.

1.7.2 Introducción a los Modelos Ocultos De Markov (MOM)

Los Modelos Ocultos de Markov fueron introducidos y estudiados a finales de 1960 y principios de 1970. Las bases teóricas fueron publicadas por L.E.Baum y sus colegas, y fueron implementadas para aplicaciones de procesamiento de la voz por Baker en CMU, y por Jelinek y sus colegas en IBM alrededor de 1970.

Sin embargo, los MOM, son ampliamente comprendidos y aplicados en aplicaciones del procesamiento de voz varios años más tarde. La razón por la cual los MOM son ampliamente utilizados, se debe a que los modelos poseen una rica estructura matemática y permiten formar la base teórica para utilizarse en una variedad de aplicaciones, además cuando son usados en la práctica trabajan muy bien.

1.7.2.1 Modelos Ocultos de Markov (MOM)

A continuación se estudiará un modelo de Markov, en el cual, la observación es una función probabilística del estado, por ejemplo, el modelo resultante (conocido como Modelo Oculto de Markov) es un proceso doblemente estocástico, con un proceso estocástico no observable (es oculto), pero solo es observado por otro conjunto de procesos estocásticos que producen la secuencia de observaciones.

Considere el siguiente modelo que corresponde a una urna y bolas.

a. Modelo de una Urna y bolas

Para comprender el funcionamiento de los MOM, consideremos el sistema de urnas con bolas, indicado en la figura I.12, el cual ilustra el caso general de un MOM de símbolos discretos. Nosotros asumimos que tenemos N urnas de vidrio en un cuarto. Dentro de cada urna tenemos un gran número de bolas de diferentes colores.

El proceso físico para obtener las observaciones es el siguiente. Una persona dentro de un cuarto, y de acuerdo a algún proceso aleatorio, escoge una urna. De esta urna, una bola es escogida en forma aleatoria, y el color es tomado como la observación. La bola puede ser entonces reemplazada en la urna de la cual fue seleccionada. Una nueva urna es entonces seleccionada de acuerdo al proceso aleatorio asociado a la actual urna.

Es claro para este simple MOM que corresponde al sistema formado por las urnas y las bolas, que cada estado corresponde a una urna específica, para la cual, un posible color (una bola) está definido para cada estado. La selección de la urna está gobernada por la matriz de transición de estados del MOM.

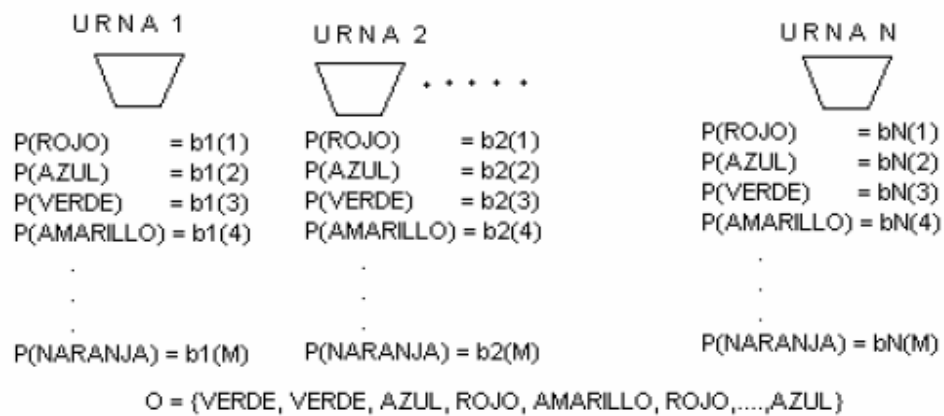


Fig. 1.12: Modelo de urna y bolas con n -estados.

1.7.2.1.1 Elementos de un MOM

1) N , es el número de estados en el modelo. Aunque los estados son ocultos, para muchas aplicaciones, frecuentemente hay algún significado físico vinculado con el estado o el conjunto de estados del modelo. Por ejemplo, en el modelo de la urna y bolas, los estados corresponden a las urnas. Generalmente los estados están interconectados, en tal manera, que cualquier estado puede ser alcanzado, desde cualquier otro estado (por ejemplo un modelo ergódico). Nosotros representaremos los estados como $S = \{S_1, S_2, \dots, S_N\}$, y el estado en el tiempo t como q_t . Así tenemos formada una sucesión.

2) M , es el número de símbolos distintos observados por estado, por ejemplo, el tamaño de un alfabeto discreto. Los símbolos observados corresponden a la salida física del sistema que está siendo modelado. Para el ejemplo del modelo de urnas y bolas, los símbolos observados fueron los colores de las bolas seleccionadas de la urna. Nosotros denotamos los símbolos de manera individual como $V = \{V_1, V_2, \dots, V_M\}$.

3) La distribución de probabilidad de transición de estado se define como $A = \{a_{ij}\}$, donde

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i] \quad 1 \leq i, j \leq N$$

Para el caso especial donde cualquier estado puede alcanzar a cualquier otro estado en un simple paso, nosotros tenemos $a_{ij} > 0$ para todo i, j . Para otros tipos de MOM, nosotros tendríamos $a_{ij} = 0$ para una o más parejas (i, j) . 4) La distribución de probabilidad de observación de símbolo en cada estado es $B = \{b_j(k)\}$, donde

$$b_j(k) = P[V_k \text{ en } t | q_t = S_j], \quad 1 \leq j \leq N \\ 1 \leq k \leq M$$

5) La distribución de probabilidad del estado inicial se define como $\pi = \{\pi_i\}$ donde

$$\pi_i = P[q_1 = S_i] \quad 1 \leq i \leq N$$

Fijados valores apropiados para N , M , A , B y π , los MOM pueden usarse como un generador para dar una secuencia de observaciones (donde cada observación O_t es uno de los símbolos de V , y T es el número de observaciones en la secuencia).

$$O = O_1 O_2 \dots O_T$$

Este tipo de observaciones pueden acrecentar para dar mayor precisión.

La simulación de un MOM implica los pasos que siguen:

- Escoger un estado inicial $q_1 = S_i$ de acuerdo a la distribución de estado inicial π .
- Poner $t = 1$
- Escoger $O_t = V_k$ de acuerdo a la distribución de probabilidad de símbolo en el estado S_i , por ejemplo $b_i(k)$.
- Pasar a un nuevo estado $q_{t+1} = S_j$, según la distribución de probabilidad de transición para el estado S_i , por ejemplo, a_{ij} .
- Poner $t = t+1$, regresar al paso 3, si $t < T$, caso contrario se termina el proceso.

El procedimiento descrito puede ser utilizado como un generador de observaciones, o también como un modelo para el cual una secuencia de observación dada fue generada por un apropiado MOM.

Una completa especificación de un MOM requiere especificar dos parámetros del modelo (N y M), especificación de los símbolos observados y la especificación de tres medidas probabilísticas A , B , y π . Por conveniencia, usamos la notación compacta, para indicar todos los parámetros del modelo.

$$\lambda = (A, B, \pi)$$

1.7.2.1.2 Problemas básicos para un MOM

Existen tres problemas de interés en los MOM, que deben resolverse para que sean útiles en aplicaciones del mundo real. Estos problemas son los siguientes: Problema 1: Dado la secuencia de observación $O = O_1, O_2, \dots, O_T$, y el modelo $\lambda = (A, B, \pi)$, ¿cómo hacer eficiente en tiempo de cómputo, la probabilidad de la secuencia de observación dado el modelo $P(O | \lambda)$?

Este primer paso, es la evaluación del problema: dado un modelo y una secuencia de observaciones, como calcular la probabilidad que la secuencia observada fue producida por el

modelo. También podemos ver el problema como una calificación, de que bien un modelo dado se ajusta a las secuencias observadas.

El último punto de vista es sumamente útil. Por ejemplo, en el caso, en el cual nosotros estamos tratando de escoger un modelo, de una serie de modelos, la solución al problema 1 nos permite escoger el modelo que mejor se ajuste a las observaciones. El mejor procedimiento para resolver este problema es el algoritmo de avance – retroceso conocido en la terminología inglesa como algoritmo Forward – Backward principalmente porque reduce significativamente el coste computacional.

Problema 2: Dada la secuencia de observación $O = O_1, O_2, \dots, O_T$, y el modelo λ , ¿ cómo encontrar la secuencia de estados $Q = q_1 q_2 \dots q_T$, que mejor explique las observaciones ?

Al resolver el problema 2, nosotros intentamos descubrir la parte oculta del modelo, por ejemplo, escoger la correcta secuencia de estados. Para situaciones prácticas, utilizamos un criterio que sea óptimo para resolver el problema como mejor sea posible. Desafortunadamente, hay varios criterios, que pueden ser adoptados, la selección del criterio depende del uso pensado para la secuencia de estados descubierta. Un método se basa en métodos de programación dinámica, y se conoce con el nombre de Algoritmo de Viterbi.

Problema 3: ¿Cómo ajustar los parámetros del modelo $\lambda = (A, B, \pi)$ para maximizar $P(O | \lambda)$?

Al resolver el problema 3, intentamos optimizar los parámetros del modelo. La secuencia de observaciones usada para ajustar los parámetros del modelo es llamada secuencia de entrenamiento, puesto que esta es usada para entrenar el MOM. El entrenamiento es un problema crucial en una variedad de aplicaciones de MOM, puesto que esto permite adaptar los parámetros del modelo óptimamente a los datos de entrenamiento observados, por ejemplo, crear el mejor modelo para un fonema real. No se conoce una solución analítica para el modelo, que maximice la probabilidad de la secuencia de observación. De hecho, dada

cualquier secuencia de observación finita como dato de entrenamiento, no hay una manera óptima de estimar los parámetros del modelo. Sin embargo, se puede escoger $\lambda = (A, B, \pi)$ tal que $P(O/\lambda)$ sea maximizado localmente usando procedimientos iterativos como el método de Baum – Welch, o usando el método del gradiente.

1.7.3 Redes Neuronales Artificiales

Una Red Neuronal Artificial es una técnica de procesamiento de datos que se basa en el sistema nervioso. Tanto animales como humanos somos capaces de reaccionar adaptándonos a cambios en nuestro ambiente externo o interno, para realizar dichos cambios utilizamos el sistema nervioso. El objetivo de las redes neuronales artificiales es emular el comportamiento del sistema nervioso, para realizar tareas como reconocimiento de imágenes, interpretación de sonidos, en general tareas de procesos de reconocimiento y aprendizaje.

1.7.3.1 Definición de una Red Neuronal

No existe una sola definición para describir lo que es una red neuronal, por esta razón a continuación mencionamos algunas: Técnica de procesamiento de datos inspirada en el sistema nervioso animal (McCulloch y Pitts, 1943)

Redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptivos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico.

Una red neuronal es una estructura de procesamiento de información paralela y distribuida, que intenta emular las funciones computacionales elementales de la red nerviosa del cerebro humano, en base a la interconexión de multitud de elementos de procesamiento, cada uno de los cuales presenta un comportamiento completamente local.

1.7.3.1.1 Ventajas de las Redes Neuronales.

Aprendizaje: consiste en la habilidad de aprender mediante una etapa de aprendizaje que consiste en proporcionar a la RNA (Redes Neuronales Artificiales) datos como entrada a su vez que se le indica cuál es la salida (respuesta) esperada.

Auto organización: crea su propia representación de la información en su interior. Sin necesidad de una “programación explícita”.

Tolerancia a fallos. Almacena la información de forma redundante, ésta puede seguir respondiendo aceptablemente aún si se daña parcialmente.

Flexibilidad: puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada, por ejemplo, si la información de entrada es la imagen de un objeto, la respuesta correspondiente es acertada, incluso si la imagen tiene parámetros de luz ligeramente distintos o el objeto cambia ligeramente de posición
Tiempo real: es paralela, si se implementa con computadoras o en dispositivos electrónicos que utilicen dicha paralelización se pueden obtener respuestas en tiempo real, de la misma manera que el cerebro es capaz de procesar cantidades ingentes de información en paralelo sin un esfuerzo aparente: comprensión de imágenes, comprensión de sonidos, comprensión de parámetros táctiles, de calor y movimiento.

1.7.3.1.2 Comparación entre las Redes Neuronales y los Computadores digitales.

Las Redes Neuronales no aplican criterios de circuitos lógicos o digitales. Los computadores digitales, tienen un razonamiento deductivo, se aplica reglas conocidas en la entrada de datos para producir la salida. Las redes neuronales, tienen un razonamiento inductivo, y dados los datos de entrada y salida, se construyen las reglas.

Las redes neuronales cuentan con mayor velocidad de proceso dadas algunas características:

En un computador el cómputo es centralizado síncrono y serial.

La computadora cuenta con una complejidad relativamente pequeña con respecto R.N.

En una red neuronal el cómputo es colectivo, asincrónico y paralelo.

Un computador digital es exacto. Una red neuronal es inexacta.

Un computador digital tiene conectividad estática. En una red neuronal la conectividad es dinámica.

Un computador digital no es tolerante a fallas. Una red neuronal es un sistema redundante y que comparte responsabilidades, por esta razón es tolerante a fallas.

En un computador digital la memoria es centralizada. En una red neuronal la memoria es distribuida.

Un computador digital es aplicable en el caso de tener bien claras las reglas como los datos de entrada. Una red neuronal se puede adaptar, en el caso de tener reglas desconocidas o complicadas, o si los datos de entrada tienen ruido.

1.7.3.1.3 Aplicaciones de las Redes Neuronales.

Medicina. Las aplicaciones en medicina encuentran especialmente en problemas de diagnóstico médico. Por ejemplo: diagnóstico de cardiopatías, detección de tumores cancerígenos, caracterización de la dinámica en la variabilidad cardiaca, compresión de señales electrocardiográfica, predicción de enfermedades degenerativas.

Procesamiento de la señal. En esta área las redes neuronales han encontrado un amplio campo de aplicación. Algunos problemas de clasificación donde se aplican las redes neuronales serían: Ecuación de canales de comunicación, reconocimiento de patrones en imágenes, reconocimiento de voz, eliminación activa de ruido, control.

Economía. En este campo, donde hay que tomar decisiones entre varias alternativas, las redes neuronales son directamente aplicables. Así algunas de estas aplicaciones serían: concesión de créditos, detección de posibles fraudes en tarjetas de crédito, determinación de la posibilidad

de quiebra de un banco, predicción del gasto eléctrico de empresas y centrales, tendencias a corto y medio plazo en bolsas de valores, predicción de existencias.

Medio Ambiente. El ambiente es dinámico y no lineal, cualquier método aplicado a este campo necesariamente debe tener en cuenta estas características.

Tenemos, pues, otro campo importante de aplicación de las redes neuronales.

Algunas aplicaciones de éstas serían: predicción de irradiación solar, predicción de niveles tóxicos de ozono en zonas urbanas y rurales, previsión del tiempo.

1.7.3.2 Fundamentos de las Redes Neuronales

Los elementos básicos de un sistema neuronal biológico son las neuronas, que se agrupan en conjuntos compuestos por millones de ellas organizadas en capas, constituyendo un sistema con funcionalidad propia. Un conjunto de estos subsistemas da lugar a un sistema global.

Las neuronas a diferencia del resto de células tienen la capacidad de comunicarse. Las dendritas y el cuerpo celular reciben las señales de entrada, el cuerpo celular las combina e integra y emite señales de salida. El axón transporta esas señales a los terminales axónicos. En la figura I.13, se puede apreciar las partes de una neurona biológica.



Fig. I.13: Forma de una neurona.

En la figura I.14 podemos ver una RNA como una colección de procesadores paralelos conectados entre ellos en forma de un grafo dirigido, organizados de tal manera que la estructura de la red nos lleva por sí misma a considerarla como una característica a tener en cuenta a la hora de crear una RNA. Podemos representar de manera sistemática cada elemento de proceso (unidad) de la red como un nodo, con conexiones entre unidades representadas mediante arcos, además estos arcos nos indican la dirección en la que fluye la información.

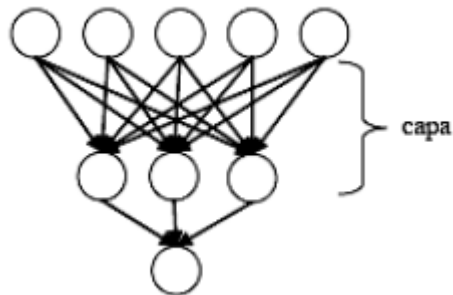


Fig. I.14: Ejemplo de una red neuronal artificial.

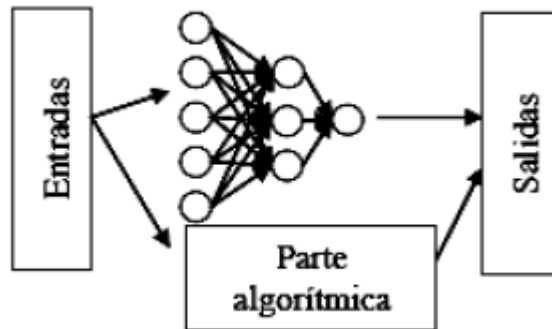


Fig. I.15: Sistema Neuronal Artificial.

CAPITULO II

ROBÓTICA

2.1 Concepto de Robot

“Máquina controlada por ordenador y programada para moverse, manipular objetos y realizar trabajos a la vez que interacciona con su entorno”.

“Un robot puede ser descompuesto en un conjunto de subsistemas funcionales: procesos, planeación, control, sensores, sistemas eléctricos, y sistemas mecánicos”.

Para mi caso en particular me apego al segundo concepto, recordemos que el robot que fue implementado sirve como un robot compañero que recibe ordenes a través de la voz, sin realizar trabajo duro alguno pero interacciona con el medio a través de sus sensores.

El subsistema de Software no tiene referencia dentro porque es una parte implícita de los subsistemas de sensores, planeación, y control; que integra todos los subsistemas como un todo.

Los robots son capaces de realizar tareas repetitivas de forma más rápida, barata y precisa que los seres humanos. El término procede de la palabra checa robot, que significa 'trabajo obligatorio'; fue empleado por primera vez en la obra teatral de 1921 R.U.R. (Robots Universales de Rossum) por el novelista y dramaturgo checo Karel Čapek. Desde entonces se

ha empleado la palabra robot para referirse a una máquina que realiza trabajos para ayudar a las personas o efectúa tareas difíciles o desagradables para los humanos.

2.2 Clasificación de los robots

La potencia del software en el controlador determina la utilidad y flexibilidad del robot dentro de las limitantes del diseño mecánico y la capacidad de los sensores. Los robots han sido clasificados de acuerdo a su generación, a su nivel de inteligencia, a su nivel de control, y a su nivel de lenguaje de programación. Estas clasificaciones reflejan la potencia del software en el controlador, en particular, la sofisticada interacción de los sensores. La generación de un robot se determina por el orden histórico de desarrollos en la robótica. Cinco generaciones son normalmente asignadas a los robots. La tercera generación es utilizada en la industria, la cuarta se desarrolla en los laboratorios de investigación, y la quinta generación es un gran sueño.

2.2.1. Robots Play-back.

Los cuales regeneran una secuencia de instrucciones grabadas, como un robot utilizado en recubrimiento por spray o soldadura por arco. Estos robots comúnmente tienen un control de lazo abierto.

2.2.2. Robots controlados por sensores

Estos tienen un control en lazo cerrado de movimientos manipulados, y hacen decisiones basados en datos obtenidos por sensores.

2.2.3. Robots controlados por visión

Donde los robots pueden manipular un objeto al utilizar información un sistema de visión.

2.2.4. Robots controlados adaptablemente

Donde los robots pueden automáticamente reprogramar sus acciones sobre la base de los datos obtenidos por los sensores.

2.2.5. Robots con inteligencia artificial

Donde los robots utilizan las técnicas de inteligencia artificial para tomar sus propias decisiones y resolver problemas. Existen robots que pueden actuar en base a aprendizaje e interactuar con el usuario con sus criterios.

La Asociación de Robots Japonesa (JIRA) ha clasificado a los robots dentro de seis clases sobre la base de su nivel de inteligencia:

- Dispositivos de manejo manual, controlados por una persona.
- Robots de secuencia arreglada.
- Robots de secuencia variable, donde un operador puede modificar la secuencia fácilmente.
- Robots regeneradores, donde el operador humano conduce el robot a través de la tarea.
- Robots de control numérico, donde el operador alimenta la programación del movimiento, hasta que se enseñe manualmente la tarea.
- Robots inteligentes, los cuales pueden entender e interactuar con cambios en el medio ambiente.

Los programas en el controlador del robot pueden ser agrupados de acuerdo al nivel de control que realizan.

Nivel de inteligencia artificial, donde el programa aceptará un comando como "levantar el producto" y descomponerlo dentro de una secuencia de comandos de bajo nivel basados en un modelo estratégico de las tareas.

Nivel de modo de control, donde los movimientos del sistema son modelados, para lo que se incluye la interacción dinámica entre los diferentes mecanismos, trayectorias planeadas, y los puntos de asignación seleccionados.

Niveles de servosistemas, donde los actuadores controlan los parámetros de los mecanismos con el uso de una retroalimentación interna de los datos obtenidos por los sensores, y la ruta es modificada sobre la base de los datos que se obtienen de sensores externos. Todas las detecciones de fallas y mecanismos de corrección son implementados en este nivel.

En la clasificación final se considerara el nivel del lenguaje de programación. La clave para una aplicación efectiva de los robots para una amplia variedad de tareas, es el desarrollo de lenguajes de alto nivel. Existen muchos sistemas de programación de robots, aunque la mayoría del software más avanzado se encuentra en los laboratorios de investigación. Los sistemas de programación de robots caen dentro de tres clases:

Sistemas guiados, en el cual el usuario conduce el robot a través de los movimientos a ser realizados.

Sistemas de programación de nivel-robot, en los cuales el usuario escribe un programa de computadora al especificar el movimiento y el sensado.

Sistemas de programación de nivel-tarea, en el cual el usuario especifica la operación por sus acciones sobre los objetos que el robot manipula.

Subsistemas del Robot.

En la actualidad, muchas de las funciones llevadas a cabo por los subsistemas son realizadas manualmente, o de una forma off-line, pero en un futuro las investigaciones en estos campos permitirán la automatización de dichas tareas.

El Subsistema de Procesos incluye las tareas que lleva a cabo el robot, el medio ambiente en el cual es colocado, y la interacción entre este y el robot. Este es el dominio de la ingeniería aplicada. Antes de que un robot pueda realizar una tarea, ésta debe ser buscada dentro de una

secuencia de pasos que el robot pueda ejecutar. La tarea de búsqueda es llevada a cabo por el Subsistema de Planeación, el cual incluye los modelos de procesos inteligentes, percepción y planeación. En el modelo de procesos, los datos que se obtienen de una variedad de sensores son fusionados (Integración Sensorial) con modelos matemáticos de las tareas para formar un modelo del mundo. Al usar este modelo de mundo, el proceso de percepción selecciona la estrategia para ejecutar la tarea. Estas estrategias son convertidas dentro de los programas de control del robot durante el proceso de planeación.

Estos programas son ejecutados por el Subsistema de Control; en este subsistema, los comandos de alto nivel son convertidos en referencias para actuadores físicos, los valores retroalimentados son comparados contra estas referencias, y los algoritmos de control estabilizan el movimiento de los elementos físicos.

Al realizar esta tarea los mecanismos son modelados, el proceso es modelado, la ganancia de lazo cerrado puede ser adaptada, y los valores medidos son utilizados para actualizar los procesos y los modelos de los mecanismos.

Desde el subsistema de control se alimentan las referencias de los actuadores al Subsistema Eléctrico el cual incluye todos los controles eléctricos de los actuadores. Los actuadores hidráulicos y neumáticos son usualmente manejados por electroválvulas controladas. También, este subsistema contiene computadoras, interfaces, y fuentes de alimentación. Los actuadores manejan los mecanismos en el Subsistema Mecánico para operar en el medio ambiente, esto es, realizar una tarea determinada. Los parámetros dentro del robot y del medio ambiente son monitoreados por el Subsistema de Sensores; esta información sensorial se utiliza como retroalimentación en las ganancias de lazo cerrado para detectar potencialmente las situaciones peligrosas, para verificar que las tareas se realizan correctamente, y para construir un modelo del mundo.

2.4 Robots con vehículos de rodado "Rovers"

Entre los innumerables tipos de movilidad y diseño centro la atención en este tipo de robots porque el diseño presentado en esta tesis se base en una plataforma de ruedas, mientras la gente y la mayoría de los animales se desplazan sobre extremidades, la mayoría de las máquinas móviles utilizan ruedas. Las ruedas son más simples de controlar, tienen pocos problemas de estabilidad, usan menos energía por unidad de distancia de movimiento y son más veloces que las extremidades. La estabilidad se mantiene al fijar el centro de gravedad del vehículo en triangulación de los puntos que tocan tierra. Sin embargo, las ruedas solamente pueden utilizarse sobre terrenos relativamente lisos y sólidos. Si se quiere utilizar el robot en terrenos rugosos las ruedas tienen que tener un tamaño mayor que los obstáculos encontrados. El arreglo más familiar para las ruedas de un vehículo es el utilizado por los automóviles. Cuatro ruedas son colocadas en las esquinas de un rectángulo. La mayoría de estos vehículos tiene maniobrabilidad limitada debido a que tienen que avanzar para poder dar vuelta. También se requiere de un sistema de suspensión para asegurar que las ruedas estén en contacto con la superficie durante todo el tiempo. Cuando el robot se desplaza en línea recta las cuatro ruedas tienen que girar a la misma velocidad, en cambio al momento de dar vuelta las ruedas interiores giran más lento que las ruedas exteriores.

En un robot móvil, estos requerimientos son alcanzados por un buen diseño mecánico y mediante el control de la velocidad de las ruedas de dirección independiente. Sin embargo las imprecisiones que se presentan para alcanzar una trayectoria definida son causadas por factores mecánicos, deslizamiento de las ruedas, dobleces en los ejes de dirección, y desalineamiento de las ruedas.

2.5 Componentes de los robots

2.5.1 Sensores

Haciendo una analogía con los seres humanos podríamos llamarlos los sentidos del robot. Sentar no es percibir; los sensores son meramente transductores que convierten un fenómeno

físico en señales eléctricas que un micro puede leer, esto puede hacerse por medio de un convertidor analógico digital (ADC), se carga un valor de un puerto de entrada/salida ó se usa una interrupción externa; comúnmente se necesita alguna interface electrónica entre el sensor y la computadora (o micro) para acondicionar y/o amplificar la señal. Existen actualmente sensores de gran alcance pero con un costo elevado, razón por la cual se implemento con un sensor infrarrojo de bajo costo TCRT5000.

Los sensores que fueron implementados en este proyecto son infrarrojos, y sensor mecánico conocido como Bumpers, que se describe su funcionamiento en la parte inferior además otras definiciones de sensores que son también útiles en la implementación de robots.

2.5.1.1 Sensores Ultrasónicos

Los sensores ultrasónicos sirven para que un robot pueda percibir información a cierta distancia del terreno en que se desenvuelve y así ser capaz de movilizarse.

Los sensores ultrasónicos emplean el fenómeno de la piezoelectricidad, esto es, cuando se deforman algunos materiales sólidos generan dentro de ellos una carga eléctrica. Este efecto es reversible en el sentido de que si se aplica una carga, el material se deformará mecánicamente como respuesta. Este principio electromagnético de conversión de energía se aplica con utilidad en ambas direcciones. El caso en que se aplica energía eléctrica y se obtiene energía mecánica se emplea en pequeños vibradores, en los sistemas sonar para la detección acústica y detección de objetos bajo el agua, en equipos ultrasónicos de pruebas industriales, y muchos otros, al usar los dos casos de aplicación de energía de uno y otro tipo para obtener la consecuente energía es como se obtienen distancias. El efecto piezoeléctrico puede responder a (o producir) deformaciones mecánicas del material en muchas formas diferentes, como dilatación en el espesor, etc. El efecto piezoeléctrico es sensible a la dirección, porque la

ensión produce una polaridad definida en el voltaje, mientras que la compresión produce una opuesta.

Así pues, si al transductor piezoeléctrico de un sensor ultrasónico, con los cortes requeridos, se le aplica en sus extremos (electrodos) un voltaje, el cristal sufre cambios en sus dimensiones, lo que ocasiona un cambio de presión en el medio que lo rodea (el aire es el medio en el que se trabaja), y viceversa, al ser sometido el cristal a un cambio de presión aparecen cargas eléctricas en sus extremos, donde se crea una diferencia de potencial. Por lo que este tipo de transductor puede funcionar como emisor ó receptor ultrasónico. En figura II.16. se observa un sensor de ultrasonido de modelo HE-US23



Fig. II.16: Sensor Ultrasónico de la serie Herian Proffer modelo HE-US23

2.5.1.2 Sensores Mecánicos

Este tipo de sensor se usa para detección de obstáculos por contacto directo. No son adecuados para robots de alta velocidad ya que cuando detectan el obstáculo ya están encima y no da tiempo a frenar el robot.

Bumpers: es un conmutador de 2 posiciones con muelle de retorno a la posición de reposo y con una palanca de accionamiento más o menos larga según el modelo elegido. En estado de reposo la patita común (C) y la de reposo (R) están en contacto permanente hasta que la presión aplicada a la palanca del bumper hace saltar la pequeña pletina acerada interior y entonces el contacto pasa de la posición de reposo a la de activo (A), ver Figura II.17, se puede

escuchar cuando el bumper cambia de estado, porque se oye un pequeño clic, esto sucede casi al final del recorrido de la palanca.

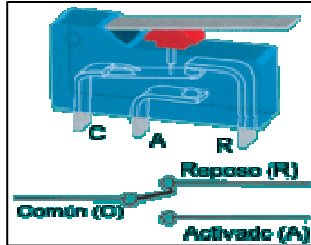


Fig. II.17: Bumper

2.5.1.3 Sensores Infrarrojos

Este tipo de sensores emplea un emisor de luz con su respectivo receptor, sus principales desventajas son la interferencia de la luz ambiental y la dependencia del color y propiedades de las superficies de los objetos interceptados por la señal, en la figura II.18 se puede observar sensores infrarrojos que se utilizó en la implementación del robot, se trata del sensor infrarrojo TCRT500.

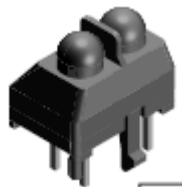


Fig. II.18: Sensor de infrarrojo TCRT500

2.5.2 Interfaces Físicas.

Las interfaces de potencia son dispositivos intermedios entre nuestro microcontrolador y aparatos que requieran cantidades de corriente mayores a los que pueden manejar nuestro microcontrolador (por lo general estamos hablando de 20 miliamperios como máximo por pin), motores de paso, DC, servomotores, lámparas, reflectores, grupos de leds son ejemplos de dispositivos que podríamos a llegar a controlar desde el microcontrolador. Por ejemplo es un grave error tratar de conectar un motor directamente a los pines del microcontrolador. Nos

valdremos de transistores, relés, puentes-H o interfaces electrónicas de control, para construir nuestras interfaces de potencia.

2.5.3 Transistores.

Los transistores pueden funcionar como amplificadores o interruptores, si los utilizamos como interruptores al igual que los relés pueden manejar corrientes altas, controlados por corrientes bajas. Los transistores son dispositivos de tres terminales (patas) y en el caso de los bipolares sus terminales se llaman emisor base y colector, al poner una corriente pequeña en la base, una corriente alta puede pasar del colector al emisor. Entre los transistores bipolares podemos diferenciar dos tipos NPN y PNP.

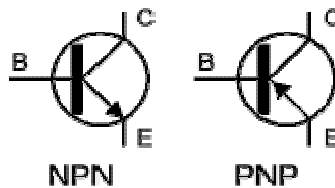


Fig. II.19: simbología transistores npn y pnp

En un transistor NPN:

La base controla el transistor. El emisor es la conexión a tierra. El colector conectará la parte de la carga a tierra.

Postulados.

La conexión base / emisor es como un diodo (Un LED que no emite luz) donde siempre habrán 0.7 volts. Si hay cero voltios o menos de 0.7 voltios en la base del transistor el diodo no conducirá y no habrá conducción entre colector y emisor.

Si el diodo conduce el transistor se prende y la corriente podrá pasar del colector al emisor. Necesario para polarizarse y trabajar en la sección de corte y saturación.

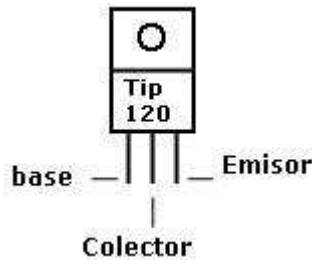


Fig. II.20: Descripción de pines del transistor Tip 120.

Transistores PNP.

Trabaja igual que un transistor NPN con la diferencia que el diodo base emisor está al revés.

Debemos tomar en cuenta que el transistor puede trabajar en tres formas.

Esto quiere decir que para funcionar la base debe estar 0.7 volts, por debajo del emisor, lo cual quiere decir que tendríamos que poner un voltaje negativo para hacerlo funcionar

2.5.4 RELÉS.

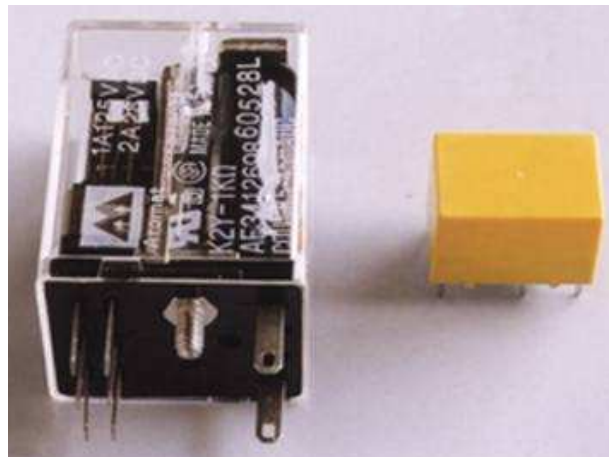


Fig. II.21: Diferentes tipos de Relés y encapsulados.

Son interruptores controlados por una pequeña corriente eléctrica. Según el relé estos pueden ser energizados con una corriente muy pequeña, por lo que pueden ser disparados directamente por el pic. (Asegurarse que no se sobrepasen los 8 miliamperios y que está conectado el diodo de protección). El relé conectará una fuente de alimentación separada del circuito pic entregando la corriente necesaria.

Al ser un interruptor mecánico puede ser bastante lento, tarda un par de milisegundos para cerrarse, si queremos switchear algo muy rápido el relé no será el dispositivo mas efectivo. Para este tipo de trabajos utilizaremos un transistor.

La figura II.22 muestra los contactos que se disparan cuando se induce corriente, en la figura II.23, se muestra un circuito adicional para que la corriente sea la suficiente para disparar.

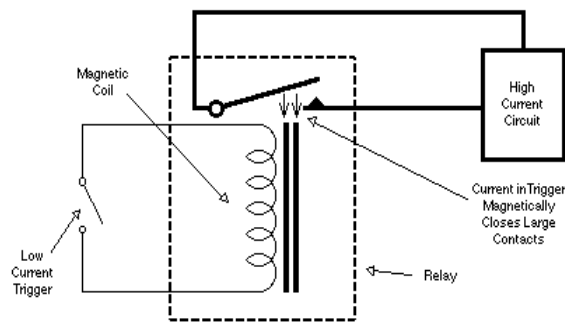


Fig. II.22: Diagrama de funcionamiento de un Relé.

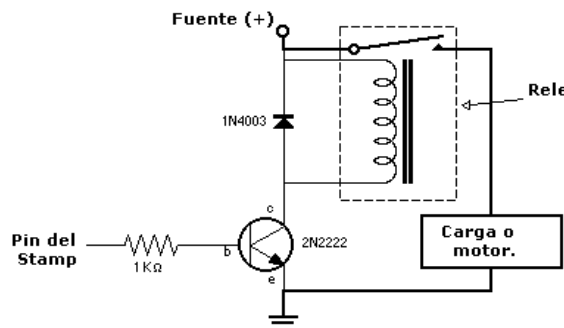


Fig. II.23: Circuito para disparar un Relé.

2.5.5 Movimiento.

Habíamos comentado que los sensores eran para el sistema, como los sentidos son a los humanos. Ahora veremos unos dispositivos que sirven para generar movimiento, los músculos del sistema electrónico.

2.5.5.1 Solenoides. Son dispositivos electromecánicos. Un solenoide típico consistiría en una bobina con un cilindro de metal en el centro, cuando se energiza la bobina se crea un campo

magnético que sube o baja el cilindro metálico. Un solenoide rotacional es un solenoide que en vez de generar un movimiento lineal produce un movimiento rotacional.

2.5.5.2 Motores DC.

Los motores DC pueden ser utilizados para el movimiento y locomoción. Por lo general tienen muchas revoluciones por minuto RPM (velocidad), pero bajo torque (fuerza), pero esto se soluciona poniendo engranajes reductores que bajarán la velocidad pero aumentarán la fuerza. Son los que se utilizan en juguetes, walkmans, coches de escape, ventiladores, etc. Al aplicar un voltaje DC el motor empieza a girar, al quitarlo se detiene, si se quiere cambiar la dirección del motor, se debe cambiar la polaridad.



Fig. II.24: Motores de corriente continua

2.5.5.3 Motores de Paso.

Pueden ser utilizados para locomoción, movimiento, dirección y posicionamiento. Estos motores se encuentran en muchísimos equipos industriales y computacionales, los escaners, las disketeras e impresoras utilizan este tipo de motores. Son especiales ya que se puede controlar digitalmente (enviando pulsos de 0's y 1s).

El grado de rotación de su eje, permitiéndonos posicionarlos con extremada precisión.

El stepper funciona con pulsos si la secuencia de pulsos es la correcta el motor dará un paso, cada motor tiene una especificación de grados por paso que recorren.

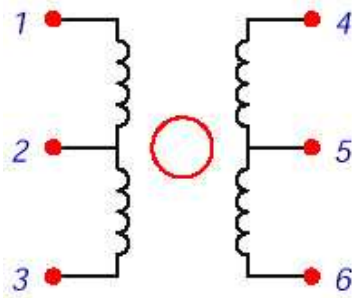


Fig. II.25: Diagrama de los bobinados de un motor unipolar.

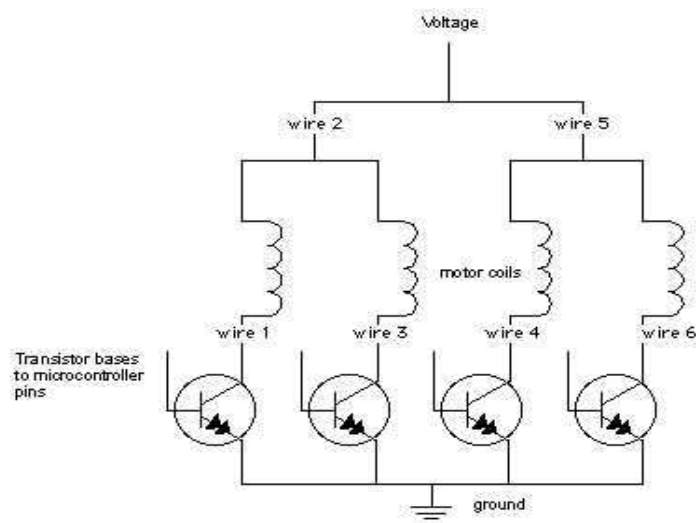


Fig. II.26: Conexión de los bobinados a los transistores.

2.5.5.4 Servos.

Son motores DC con engranajes y un sistema de retroalimentación de posición, son usados para el control de posición. Su eje puede rotar 180 grados y se utilizan mucho en aparatos de Hobby R/C para controlar coches, aviones, barcos y helicópteros.

Se controlan a través de tres cables, dos son para voltaje y tierra y el tercero para una señal de posición de control que es un pulso. Una posición central sería un pulso de 1.5 milisegundos, que es enviado 50 veces por segundo al motor. Un pulso de 1 ms rotará el eje totalmente a la izquierda y un pulso de 2 ms rotará totalmente el eje a la derecha. Cualquier valor intermedio hará posicionar el eje entre los +/-90 grados, con respecto al centro (1.5 ms).

2.5.5 Engranajes

Los engranajes son sistemas mecánicos que transmiten el movimiento de rotación desde un eje hasta otro mediante el contacto sucesivo de pequeñas levas denominadas dientes. Los dientes de una rueda dentada pueden ser cilíndricos o helicoidales.

Los engranajes cilíndricos rectos poseen dientes paralelos al eje de rotación de la rueda y pueden transmitir potencia solamente entre ejes paralelos.

El engranaje pequeño se conoce como piñón mientras que el engranaje permite reducir la velocidad como muestra la figura II.27.

Existen diferentes tipos de engranajes, pero el utilizado en esta tesis es de configuración y ranura simple.



Fig. II.27: Sistema de engranajes

La **relación de transmisión** (i) es una relación entre las velocidades de rotación de algunos engranajes conectados entre sí. Esta relación se debe a la diferencia de diámetros de las dos ruedas, que implica una diferencia entre las velocidades de rotación de ambos ejes, esto se puede verificar mediante el concepto de velocidad angular. Otro punto que se debe considerar es que al cambiar la relación de transmisión se cambia el par de fuerza aplicado, por lo que

debe realizarse un análisis para saber si este nuevo par será capaz de vencer la inercia del engranaje y otras fuerzas externas y comenzar el movimiento o por otro lado si el engranaje será capaz de soportar un par muy grande sin fallar.

Matemáticamente, la relación de transmisión entre dos engranajes circulares con un determinado número de dientes Z se puede expresar de la siguiente manera:

$$\frac{\omega_1}{\omega_2} = -\frac{Z_2}{Z_1}$$

Donde:

- ω_1 es la velocidad angular de entrada
- ω_2 es la velocidad angular de salida transmitida
- Z_1 es el número de dientes del engranaje de entrada.
- Z_2 es el número de dientes del engranaje de salida.
- El signo menos indica que se invierte el sentido del giro.

Según la expresión anterior, la velocidad angular transmitida es inversamente proporcional al número de dientes del engranaje al que se transmite la velocidad.

Si no existe disipación de calor en la transmisión del movimiento entonces podemos expresar la relación de velocidades angulares equivalente a la relación inversa de momentos:

$$\frac{\omega_1}{\omega_2} = -\frac{Z_2}{Z_1} = \frac{M_2}{M_1}$$

- M_1 es el momento transmitido a ω_1
- M_2 es el momento que sale del engranaje 2 a ω_2 .

Si uno de los engranajes es helicoidal y si se pone como entrada en la conversión de velocidad angular, entonces la velocidad de salida del engranaje circular es Z_2 veces más pequeña que la velocidad del engranaje helicoidal. Existen trenes epicicloidales donde las relaciones de transmisión se obtienen mediante la fórmula de Willis y en la intervienen engranajes

intercalados en el tren y que tienen un movimiento relativo entre el engranaje conductor y conducido. Estos mecanismos son muy comunes en los sistemas de transmisión automática de automóviles.

CAPITULO III

MICROCONTROLADORES Y MÓDULO DE RECONOCIMIENTO DE VOZ

3.1 ¿Que es un microcontrolador?

Es un circuito integrado programable que acepta un listado de instrucciones y contiene todos los componentes de un computador. Se utilizan para realizar determinadas tareas o para gobernar dispositivos, debido a su reducido tamaño, suele ir incorporado en el propio dispositivo que gobierna.

El microcontrolador es un dispositivo dedicado. En su memoria solo reside un programa destinado a gobernar una aplicación determinada, sus líneas de entradas y salidas (I/O) permiten la conexión de sensores y relay. Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea asignada.

Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para Contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM/EEPROM & FLASH.

- Líneas de (entrada / salida) para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertos Serie y Paralelo, A/D y D/A, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Evidentemente, el corazón del microcontrolador es un microprocesador, pero cabe recordar que el microcontrolador es para una aplicación concreta y no es universal como el microprocesador.

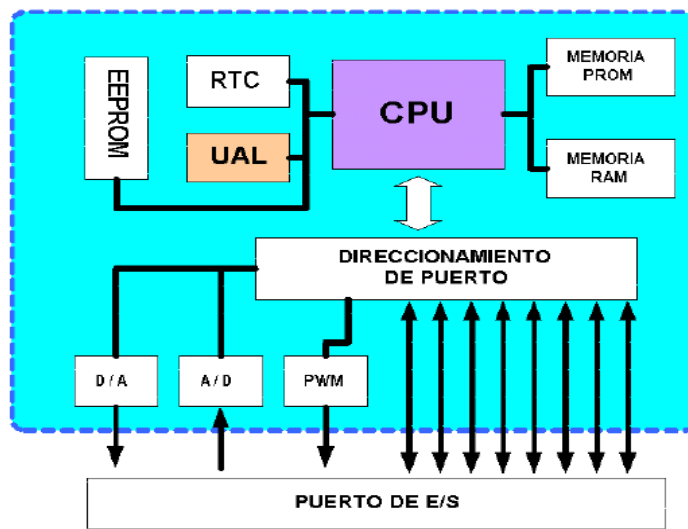


Fig. III.28: Diagrama de bloque de un microcontrolador

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna.

En este caso el controlador recibe el nombre de controlador empotrado (embedded controller).

3.1.2 ¿Diferencia entre microprocesadores y microcontroladores?

El microprocesador es un circuito integrado que contiene la Unidad Central de Proceso (CPU), también llamado procesador, de un computador.

El CPU está formado por la Unidad de Control, que interpreta las instrucciones, y el BUS de Datos, que los ejecuta.

Los pines de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control, para permitir conectarle con la Memoria y los Módulos de (ENTRADA / SALIDA) E/S y configurar un computador implementado por varios circuitos integrados.

Se dice que un microprocesador es un sistema abierto porque su configuración es variable de acuerdo con la aplicación a la que se destine.

El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos. Se podría pensar que las características de un sistema cerrado representan una desventaja con relación a los Microprocesadores, pero en la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es difícil no encontrar uno que se adapte a nuestros requerimientos del momento.

Es posible seleccionar la capacidad de las memorias, el número de líneas de (ENTRADA / SALIDA) E/S, la cantidad y potencia de los elementos auxiliares, la velocidad de funcionamiento, etc. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar.

Podemos concluir con que la diferencia fundamental entre un Microprocesador y un Microcontrolador: es que el Microprocesador es un sistema abierto con el que se puede construirse un computador con las características que se desee, acoplándole los módulos necesarios. Un Microcontrolador es un sistema cerrado que contiene un computador completo y de presentaciones limitadas que no se pueden modificar.

3.1.3 CARACTERÍSTICAS DE PIC'S UTILIZADOS

En la tabla III.V se observa las características del PIC 16F877A que fue utilizado en esta tesis:

Tabla III.V: Características del pic 16f877A

CARACTERÍSTICAS	16F877
Frecuencia máxima	DX-20MHz
Memoria de programa flash palabra de 14 bits	8KB
Posiciones RAM de datos	368
Posiciones EEPROM de datos	256
Puertos E/S	A,B,C,D,E
Número de pines	40
Interrupciones	14
Timers	3
Módulos CCP	2
Comunicaciones Serie	MSSP, USART
Comunicaciones paralelo	PSP
Líneas de entrada de CAD de 10 bits	8
Juego de instrucciones	35 Instrucciones
Longitud de la instrucción	14 bits
Arquitectura	Harvard
CPU	Risc
Canales Pwm	2
Pila Hardware	-
Ejecución En 1 Ciclo Máquina	-

Descripción de los puertos:

Puerto A:

- Puerto de e/s de 6 pines
- RA0 è RA0 y AN0
- RA1 è RA1 y AN1
- RA2 è RA2, AN2 y Vref-
- RA3 è RA3, AN3 y Vref+
- RA4 è RA4 (Salida en colector abierto) y T0CKI(Entrada de reloj del módulo Timer0)
- RA5 è RA5, AN4 y SS (Selección esclavo para el puerto serie síncrono)

Puerto B:

- Puerto e/s 8 pines
- Resistencias pull-up programables
- RB0 Interrupción externa
- RB4-7 Interrupción por cambio de flanco
- RB5-RB7 y RB3 programación y debugger in circuit

Puerto C:

- Puerto e/s de 8 pines
- RC0 - RC0, T1OSO (Timer1 salida oscilador) y T1CKI (Entrada de reloj del módulo Timer1).
- RC1-RC2 è PWM/COMP/CAPT
- RC1 è T1OSI (entrada osc timer1)
- RC3-4 è IIC
- RC3-5 è SPI
- RC6-7 è USART

Puerto D:

- Puerto e/s de 8 pines

- Bus de datos en PPS (Puerto paralelo esclavo)
- Puerto E:
- Puerto de e/s de 3 pines
- RE0 è RE0 y AN5 y Read de PPS
- RE1 è RE1 y AN6 y Write de PPS
- RE2 è RE2 y AN7 y CS de PPS

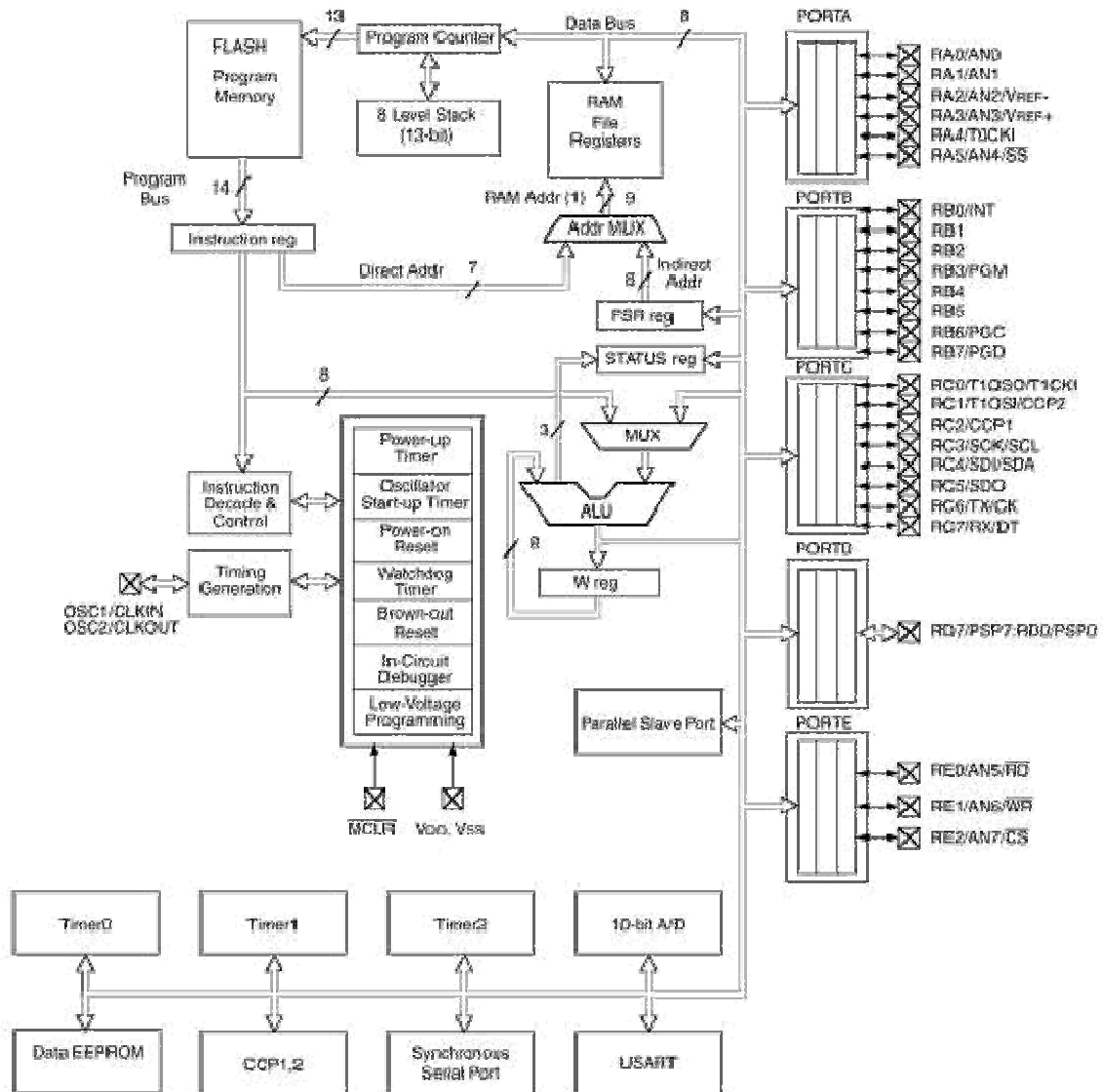
3.1.3.1 Dispositivos periféricos:

- Timer0: Temporizador-contador de 8 bits con preescaler de 8 bits
- Timer1: Temporizador-contador de 16 bits con preescaler que puede incrementarse en modo sleep de forma externa por un cristal/clock.
- Timer2: Temporizador-contador de 8 bits con preescaler y postescaler.
- Dos módulos de Captura, Comparación, PWM (Modulación de Anchura de Impulsos).
- Conversor A/D de 10 bits.
- Puerto Serie Síncrono Master (MSSP) con SPI e I²C (Master/Slave).
- USART/SCI (Universal Synchronous Asynchronous Receiver Transmitter) con 9 bit.
- Puerta Paralela Esclava (PSP) solo en encapsulados con 40 pines

3.1.3.2 Diagrama De Bloques

El pic 16f877a, utilizado en esta tesis cuenta con memoria de datos de 368 Bytes y una memoria EEPROM de 256 Bytes, con una memoria flash de 8K, cuenta con 5 puertos, este microcontrolador por su gran potencial permite realizar proyectos que necesiten gran consumo de memoria, para la realización de esta tesis este microcontrolador es el complemento para el módulo VR Stamp.

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

Fig. III.29: Diagrama de bloques pic 16F877A

3.1.3.3 Descripción De Pines

La figura III.3 muestra la rotulación o configuración de los pines, distribución de puertos, interrupciones, y los pines que pueden ser utilizados de dos funciones de acuerdo a las necesidades que se requieran.

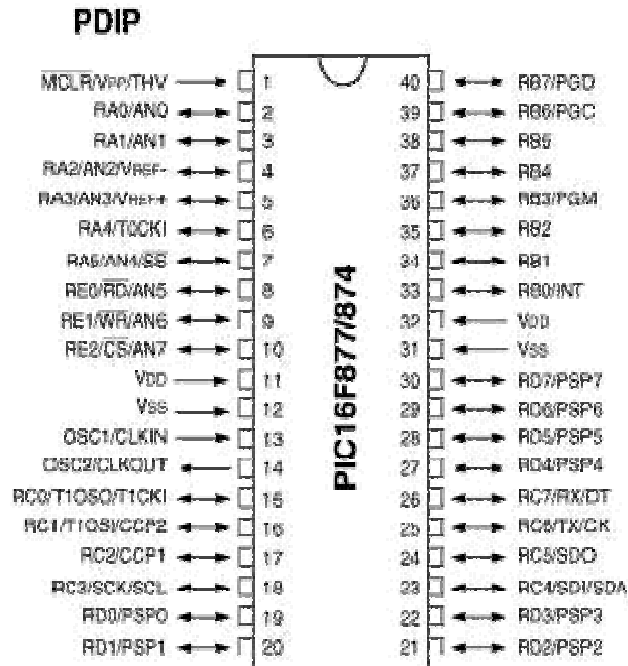


Fig. III.30: Pic 16F877A Rotulación de pines

Cada uno de los pines son descritos en la tabla III.VI.

Tabla III.VI: Descripción de pines pic 16f877A

RB0/INT	33	I/O	TTL/ST	PORTB es un puerto I/O bidireccional. Puede ser programado todo como entradas RB0 puede ser pin de interrupción externo.
RB1	34	I/O	TTL	
RB2	35	I/O	TTL	
RB3/PGM	36	I/O	TTL	RB3: puede ser la entada de programación de bajo voltaje

RB4	37	I/O	TTL	Pin de interrupción
RB5	38	I/O	TTL	Pin de interrupción
RB6/PGC	39	I/O	TTL/ST	Pin de interrupción. Reloj de
RB7/PGD	40	I/O	TTL/ST	programación serial

Tabla III.II: Descripción de pines pic 16f877A (continuación)

3.1.4 Microcontrolador RSC-4128

Este microcontrolador que esta dentro del módulo de reconocimiento de voz y que se describe mas adelante es objeto de estudio el cual presenta las siguientes características:

El microcontrolador RSC-4128 es un C.I especial para el reconocimiento de voz con entradas y salidas mezclando y procesando estas señales.

Cuenta con 8 bits con mejoras en las instrucciones de interrupción de control, con un registro superior y arquitectura independiente, un filtro digital, un acelerador matemático.

El RSC-4128 tiene 20 direcciones y 8 bit de bus de datos para interfaz con memoria externa. Esto incluye un –XM (Program Run) como entrada para ejecutar el programa, opciones de habilitar o deshabilitar la memoria externa ROM.

Tres puertos bidireccionales con 24 líneas configurables de propósito general para control de dispositivos y variedad de fuentes.

Cuatro de estas líneas pueden ser programados como comparadores analógicos de entrada. 16 pueden ser utilizadas como entradas y salidas I/O (input autput) wakeup.

El RSC-4128 tiene una frecuencia alta de 14.32 MHz de reloj así como una frecuencia de 32.768 Hz de reloj. El procesador de reloj puede ser seleccionado de otra fuente.

El diagrama de bloques muestra todos los componentes del módulo VR Stamp; convertidor análogo digital, filtros, pre amplificadores, pulsos de modulación. Ver fig. III.31

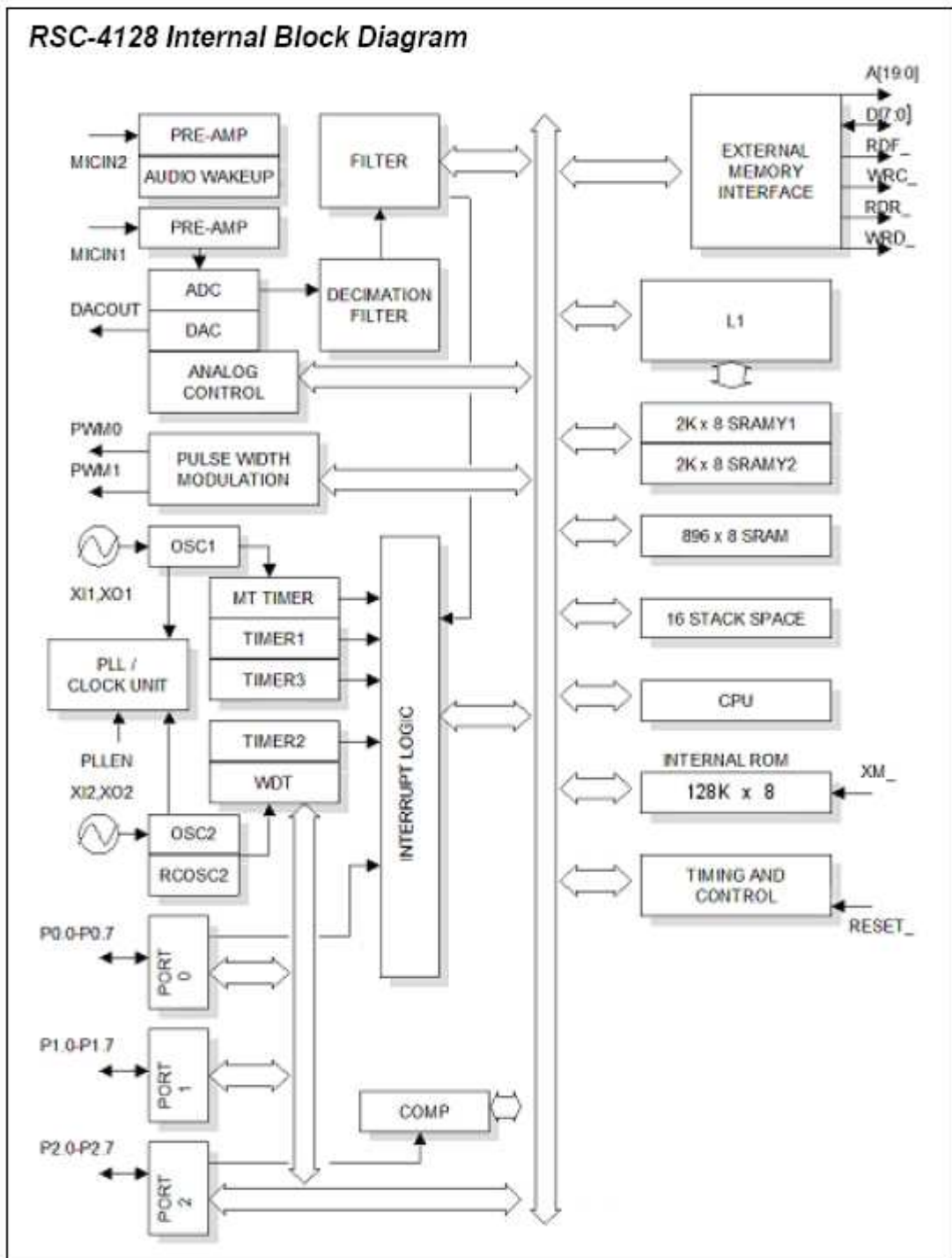


Fig. III.31: Diagrama de bloques RSC-4128

3.2 Estudio Módulo de Reconocimiento de Voz

Se ha estudiado cual es el funcionamiento y técnicas de reconocimiento de voz en el capítulo 1, lo cual permite centrarnos en el módulo de reconocimiento de voz adecuado para este

proyecto, recordemos que el objetivo de este trabajo final es el diseño de un prototipo de robot compañero controlado por comandos de voz, por lo que deberá presentar un reconocimiento de voz Independiente del hablante y sea amigable en su manejo, para lo cual se investigó algunos de estos módulos siendo el apropiado el módulo de reconocimiento de voz RSC-4x RPM de la empresa Sensory, por sus características de desempeño que a continuación se describe.

3.2.1 Módulo de Reconocimiento de Voz VR Stamp.

3.2.1.1 Descripción RSC-4x RPM (Rapid Prototyping Module)

El módulo de reconocimiento de voz RSC-4x RPM (Rapid Prototyping Module) o conocido como VR Stamp™, de prototipo simplificado ofrece tecnologías del habla integradas y codificadas en un integrado en un módulo de 40 pines DIP. El RSC-4x RPM's tiene una memoria Flash de 32 Mbit. Este módulo es de la empresa Sensory Inc., puede ser utilizada para grabar y reproducir varios archivos de voz. El RPM es un prototipo de herramientas designado para desarrollar varios productos (ver figura III.32).

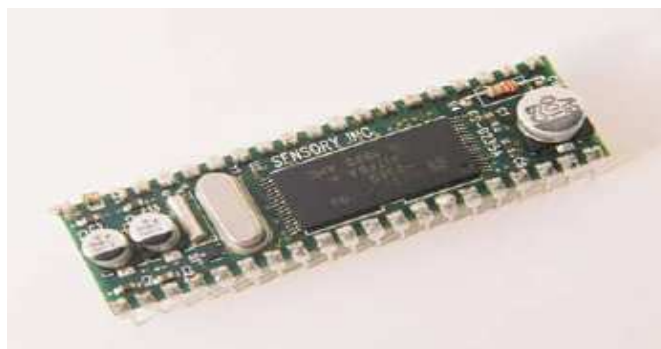


Fig. III.32: Módulo de reconocimiento de voz VR Stamp™

Con su módulo de programación MPB, (Module Programming Board), se puede descargar aplicaciones de programas en el módulo RSC-4x RPM. El módulo puede ser reprogramado al igual que cualquier microcontrolador.

3.2.1.2 CARACTERÍSTICAS

3.2.1.2.1 Capacidades

Ruido robusto independiente del habla (SI) Speaker Independent lo que quiere decir que el reconocimiento de palabras es independiente del locutor. Implica palabras fijas ya modelizadas y no necesita aprendizaje, y reconocimiento dependiente del habla SD (Speaker Dependent) el reconocimiento de palabras es dependiente del locutor. Implica modelizar o entrenar palabras mediante aprendizaje.

Reconocimiento de patrones independientes incluso para varios lenguajes, para uso internacional.

Alta calidad, 2.4 a 7.8 Kbps de síntesis del habla y efectos de sonido si incluye con el sensory "SX", que son herramientas de software adicionales de este módulo.

3.2.1.2.2 Compresión del habla con "SXL".

Señal de la palabra con Verificación del habla, Speaker Verification Word Spot (SVWS) – con seguridad al ruido.

8 voces MIDI-compatible con síntesis de música.

Audio wakeup from sleep. Despertar al detectar audio.

Toque de Tono de salida (DTMF).

Graba y reproduce (RP).

3.2.1.2.3 Soluciones Integradas

RSC-4128, procesos del habla y 4 Mbit de memoria Flash.

32 Mbit de memoria flash para datos seriales.

14 Mhz (principal) y 32 khz (time keeping), de reloj.

24 líneas de entrada y salida.

Amplificador de micrófono.

Modulación con ancho de pulso (PWM) para el habla y salida opcional de un convertidor análogo digital DAC.

3.2.1.2.4 Bajos requerimientos de potencia

VDD = 3.0V – 3.6V

I_{ACT} = 26mA @ 3V (typ. for complete module)

I_{SLEEP} = <20uA @ 3V (typ. for complete module)

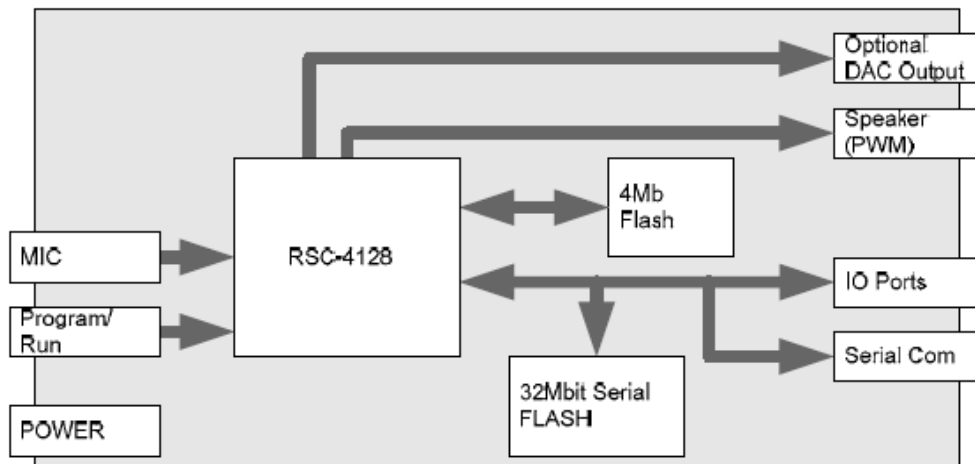


Fig. III.33: Diagrama de bloques RSC-4X RPM

Especificaciones DC

3.2.1.3 Componentes principales

3.2.1.3.1 U3 RSC-4128 IC

Un circuito integrado rsc-4128 es el procesador principal.

3.2.1.3.2 Y1 Cristal

Un cristal de 3.57 Mhz controla a un 14.3 Mhz de fase de bloqueo de bucle phase lock loop (PLL), el RSC-4128 se crea un sistema de reloj. El timer 1 es usado para todas las funciones de reconocimientos de voz y solo debe ser usado por el programador de la aplicación.

3.2.1.3.3 Y2 Cristal

Un cristal de 32.768 KHz establece la frecuencia del timer 2 sobre el RSC-4128 IC. El timer 2 es usado para cronometro desde aplicaciones que no están interrumpidas, diferente al timer 1.

3.2.1.3.4 R4 fuente de Micrófono resistor (ganancia)

Una resistencia de 1.8 k se calibra para uso del micrófono con -44 db de sensibilidad con 2.2 k esto satisface la ganancia de sistema de -46 db.

3.2.1.3.5 U2 4Mbit Memoria Flash serial

Esta memoria flash serial de 32 M es designada para almacenar plantillas de reconocimiento de voz, graba y reproduce datos, y aplicaciones de datos.

3.2.1.4 Especificaciones pines RSC-4X RPM

Tabla III.VII: Descripción pines VR Stamp™

Pin	Description	Pin	Description
1	Microphone Return (Analog GND)	40	GND
2	Microphone VDD	39	VDD
3	Microphone Input	38	P0.7
4	Microphone Power	37	P0.6 (Serial Flash SO)
5	DAC Output	36	P0.5 (Serial Flash SI)
6	P2.0	35	P0.4 (Serial Flash Clock)
7	P2.1	34	P0.3 (Serial Flash -CE)
8	P2.2	33	P0.2
9	P2.3	32	P0.1 (Serial Com TXD)
10	P2.4	31	P0.0 (Serial Com RXD)
11	P2.5	30	P1.7
12	P2.6	29	P1.6
13	P2.7	28	P1.5
14	P1.0	27	Power Down (PDN)
15	P1.1	26	-Reset
16	P1.2	25	Program/-Run (-XM)
17	P1.3	24	PWM 1 (Speaker Output)
18	P1.4	23	PWM 0 (Speaker Output)
19	VDD	22	VDD
20	GND	21	GND

Nota: P0.3, P0.4, P0.5 y P0.6 podría ser reservado para acceder a SD/SV (Single-word SD Wordspotting) Reconocimiento de una palabra, dependiente del locutor, dentro de una frase. Implica modelizar palabras mediante aprendizaje, SV (Speaker Verification) Reconocimiento del locutor. Implica aprendizaje del locutor. Reconocimiento graba y reproducir datos almacenados en la memoria flash serial.

3.2.1.4.1 (-Reset)

Entrada: Un pulso negativo resetea el RSC-4x RPM.

3.2.1.4.2 Programa de ejecución (-XM)

Entrada: ejecuta la aplicación, este abandona la ejecución si el pin es desconectado. Descarga aplicaciones binarias, arranca este pin en alto con una resistencia de 100 Ohms.

3.2.1.4.3 Potencia baja (PDN)

Salida: la potencia baja es activada si el pin esta con salida en alto, y es controlada por la aplicación firmware.

3.2.1.4.4 Propósito general I/O

Bidireccional: el RSC-4x RPM tiene 24 pines de propósito general. Cada línea puede ser programada como una entrada o salida.

3.2.1.4.5 Puertos de micrófono.

Entrada: el RSC-4x RPM es designado así que puede usarse un micrófono con una resistencia de 1.2 K o un micrófono externo. El micrófono dependiendo de las resistencia que se configure puede proporcionar (1.2K) a -49dB ganancia del sistema.

Para modo Sleep la corriente debe ser de 100 μ A (typ.) con un micrófono.

3.2.1.4.6 Salida de parlante (PWM0, PWM1)

Salida: puede ser directamente conectado al parlante.

3.2.1.4.7 Dac Salida

Salida: Convertidor análogo digital es opcional para la salida del parlante. Para usar DAC como salida se necesita proveer un amplificador externo.

3.2.1.5 Aplicaciones RSC-4x

Se puede desarrollar con el RSC-4x RPM:

- Reconocimiento independiente de voz
- Reconocimiento dependiente de voz
- Verificación de voz
- Grabar y reproducir voz.
- Compresión de voz SXL

Creando archivos independientes de voz

Para crear archivos independientes de reconocimiento se usa "Quick T2SI (Pat. Pending)" "QuickSynthesis 4".

CAPITULO IV

APLICACIÓN Y DESARROLLO

4.1 Funcionamiento General

El robot cuenta con tres aspectos fundamentales para el funcionamiento; el primero es la elección del hardware del robot, el segundo es la circuitería para el control de los motores y el tercer aspecto importante es la programación del módulo de reconocimiento de voz RSC-4128 RPM, junto con el microcontrolador 16f877A que se encarga de ejecutar las ordenes binarias que provienen del VR Stamp, lo descrito en estas tres etapas viene implícito el resto de componentes adicionales que componen en su conjunto el robot.

En los capítulos anteriores se ha estudiado los conceptos y cada uno de los componentes del robot. Conoscamos como debe funcionar el robot, tomando en cuenta que éste debe ser capaz de ejecutar ordenes a través de la voz. Las ordenes de movilidad son: bailar, adelante, atrás, izquierda, derecha que son comandos que requiere el funcionamiento de los motores, además se programó comandos que son para la distracción y ayuda para quien envía las ordenes, estos comandos son: bienvenida, saludo, encender, equivocaste, cantar, televisión, radio, estos dos últimos comandos sirve para el control demótico del hogar realizado a través

de radio frecuencia, esta etapa será descrita a mayor detalle mas adelante en este mismo capitulo.

El robot deberá desplazarse hacia adelante hasta que el sensor detecte obstáculos que permitirá retroceder y viceversa para el comando hacia atrás.

La figura IV.34 muestra las tres etapas que serán analizadas y descritas individualmente para la implantación de este proyecto.

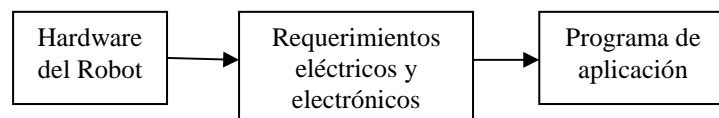


Fig. IV.34: Etapas de implementación del robot

4.2 Hardware del robot

El hardware del robot no es diseñado sino que es adaptado a las necesidades que se requiere para la implementación del robot, se realiza un estudio del funcionamiento de engranajes que es la parte más importante para el sistema mecánico ver capítulo 2 (engranajes). Para el hardware del robot se implementó con dos juguetes; uno que representa el cuerpo del robot y está incluido un motor de C.C incrustado en un sistema de engranajes que permite la movilidad de las manos, este tiene el aspecto de un androide y para la parte del desplazamiento del robot se cuenta con una carrocería de configuración Ackerman, es decir dos ruedas traseras con un sistema de engranajes con un motor de C.C que sirve como tracción, y dos ruedas delanteras que sirven como dirección de igual forma con engranajes y un motor C.C. al realizar el giro estas dos ruedas se mueven fácilmente hacia la izquierda y derecha, dependiendo la polaridad que tenga el motor.

Exite una relación entre los ángulos de las ruedas que dan la dirección del robot como se muestra en la figura IV.35, viene dada por la ecuación 4.1:

$$\cot \theta_i - \cot \theta_0 = \frac{d}{l} \quad \text{Ec: (4.1)}$$

Siendo:

θ_i = Angulo relativo a la rueda interior.

θ_0 = Angulo relativo a la rueda exterior.

d= Separación lateral entre ruedas.

l= Separación longitudinal entre ruedas.

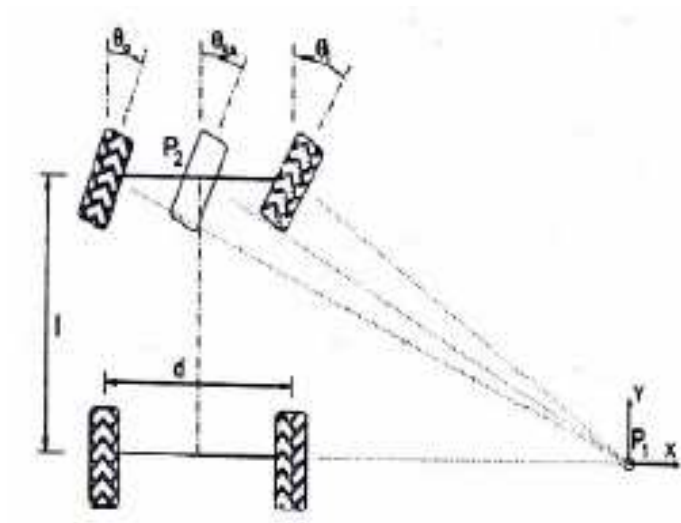


Fig. IV.35: posición de las ruedas en configuración Ackerman

4.2.1 Sistema de engranajes:

Los sistemas de engranajes utilizados en el robot aumentan el torque y disminuye la velocidad, utiliza como piñón el engranaje del motor para generar fuerza al contacto con el engranaje de mayor tamaño.

El cuerpo del robot cuenca con mecanismos que son un conjunto de elementos rígidos, móviles unos respecto de otros, unidos entre sí mediante diferentes tipos de uniones, llamados pares cinemáticas (pernos, uniones de contacto, pasadores, etc.), cuyo propósito es la transmisión de movimientos y fuerzas.

4.3 Requerimientos eléctricos y electrónicos

En la figura IV.36 se observa el diagrama de bloques del sistema eléctrico y electrónico del robot a ser implementado. Para el control domótico se cuenta con la transmisión inalámbrica el mismo que es receptado y procesado como se observa en la figura IV.37.

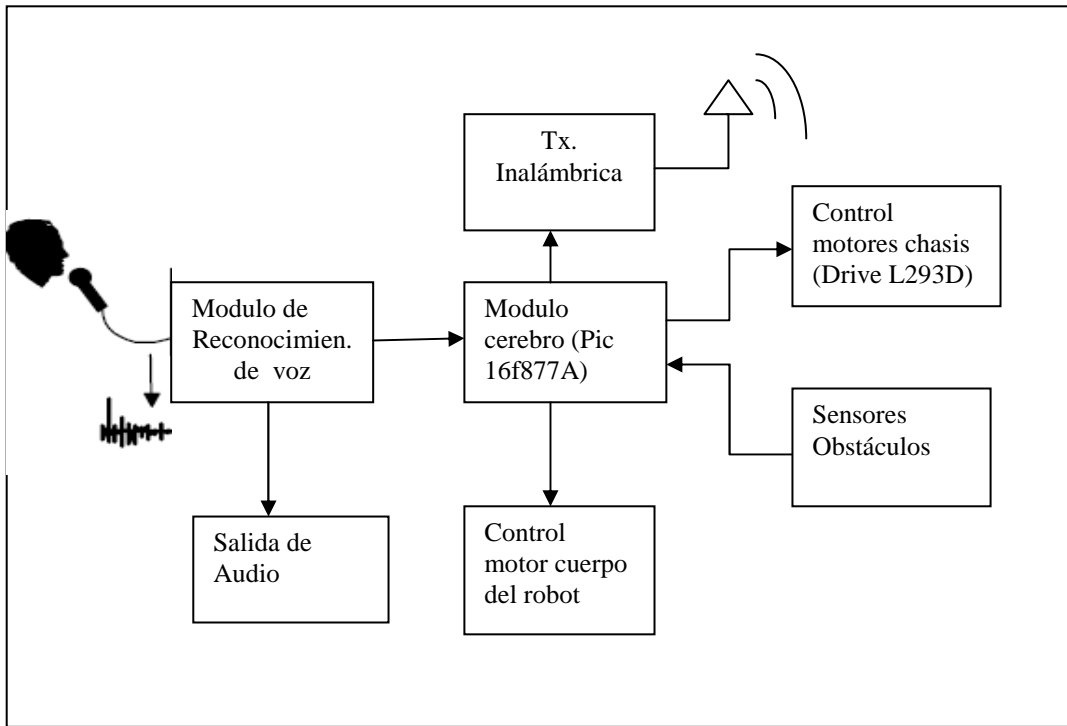


Fig. IV.36: Diagrama de bloques del sistema electrónico

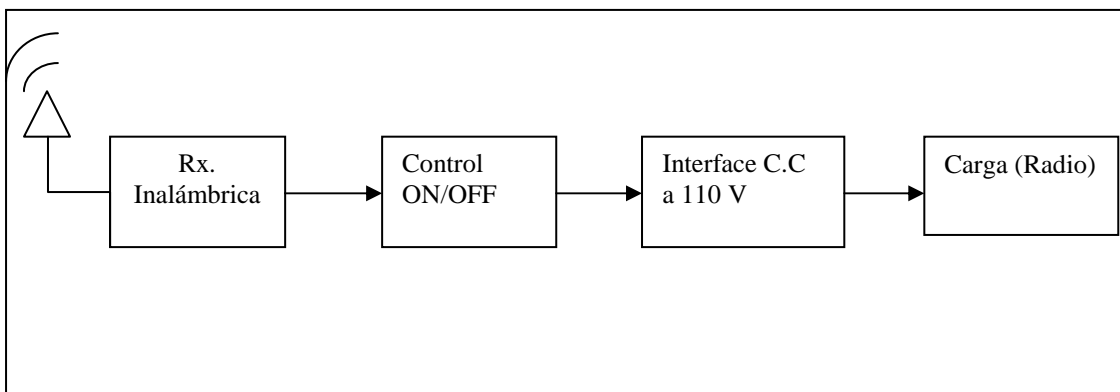


Fig. IV.37 : Diagrama de bloques del sistema electrónico (Recepción)

4.3.1 Diseño de la placa principal de control

El módulo de reconocimiento de voz (VR Stamp), y el módulo cerebro (PIC 16F877A), constituyen la placa principal. Al ingresar las señales de voz a través del micrófono el módulo de reconocimiento de voz procesa la señal ingresada por programación éste a su vez envía pulsos seriales con una transmisión asíncrona hacia el módulo cerebro que es la que verifica que orden fue ingresada, (mayores detalles en la etapa de programación), físicamente la placa esta diseñada como se muestra la figura IV.38, que comunica con el resto de módulos a través de conectores, de igual modo para la recepción de información de los sensores que son enviados al módulo cerebro donde procesa la información, así también envía combinaciones de pulsos digitales hacia el driver L293D para el control de los motores del chasis del robot.

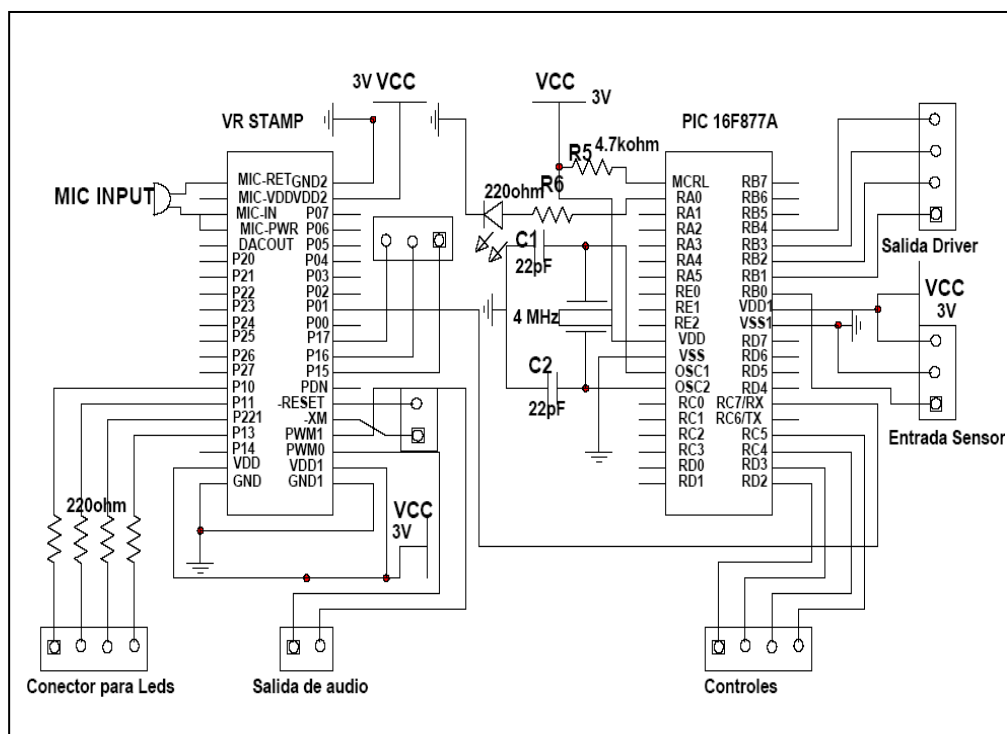


Fig. IV.38: Plaqueta principal de control

4.3.2 Módulo Control Motores

Para el control del módulo de los motores se usa el C.I L293D, para los motores del

chasis, el C.I L293D, hace la función de un circuito puente es un driver con cuatro entradas de señales digitales, alimentación independiente para cada motor, capaz de controlar motores de hasta 24 V CC, consume voltajes desde 4.5 hasta 36 Vcc, controla corrientes bidireccionales desde 600-mA, proporciona corriente de salida hasta 1A, tiene configuración Darlington por lo que el consumo de corriente es alto es la principal desventaja de este circuito integrado especialmente cuando el consumo de corriente esta en su capacidad máxima de la fuente de alimentación. El esquema de montaje para el control de los motores se muestra en la figura IV.39, que recibe las señales combinadas desde el pic 16F877A para controlar la polaridad de los motores.

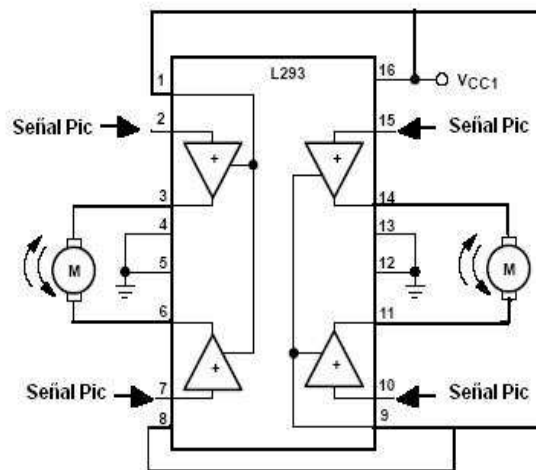


Fig. IV.39: Control de motores Driver L293D

4.3.3 Transmisión y recepción inalámbrica

La transmisión inalámbrica se realiza con elementos reciclados de un juguete controlado por radio frecuencia, el C.I trabaja a 47 MHz, esta pequeña plaqueta realiza la etapa de modulación y transmisión, funciona con 3 V.

En la figura IV.40 se observa la entrada de la señal del pic hacia el bloque de transmisión de la señal, únicamente se genera un solo pulso para el encendido y apagado del dispositivo, por medio de programación se realiza la detección del estado que se encuentra el módulo receptor

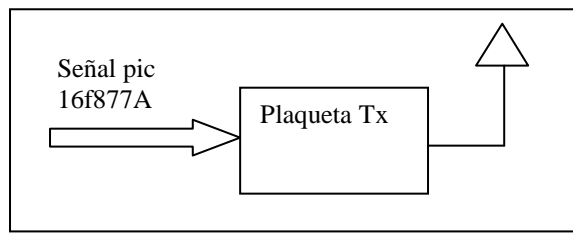


Fig. IV.40: transmisión Inalámbrica

La recepción inalámbrica se realiza con la misma frecuencia y demodulación descrita en la parte de transmisión, se ha utilizado el pic 16f628A para que se encargue de mantener encendida o apagada la carga, tomando en cuenta que solo se tiene un solo pulso para apagar o encender, ver figura IV.41.

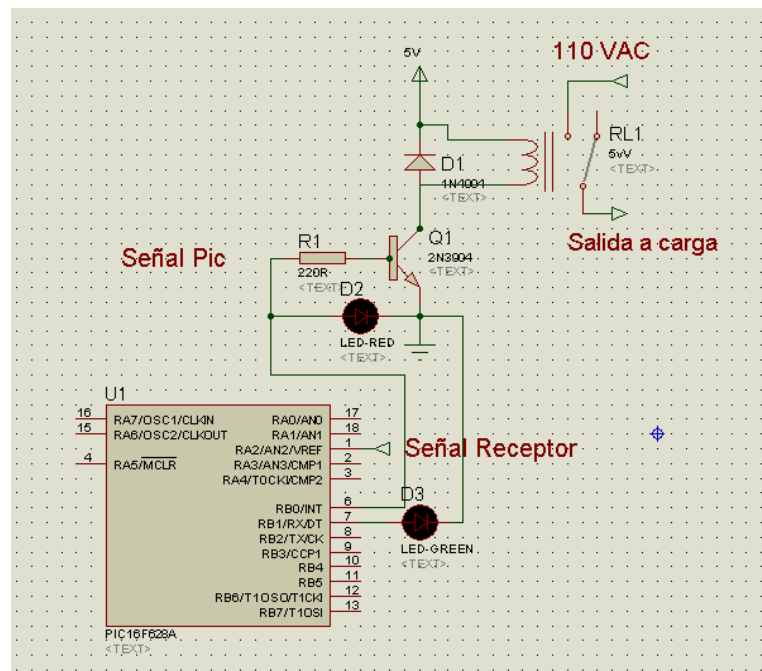


Fig. IV.41: Recepción Inalámbrica

4.3.4 Circuito para el control del cuerpo del robot

El diseño de esta sencilla plaqueta se realizó con el fin de controlar el movimiento de los brazos del robot, como sugerencia para el lector solo se puede utilizar un TIP 110 en

reemplazo del relé y sus componentes, si el objetivo es mover en un solo sentido el motor C.C, el relé utilizado (ver figura IV.42) es a 3V razón por la cual se gana en potencia.

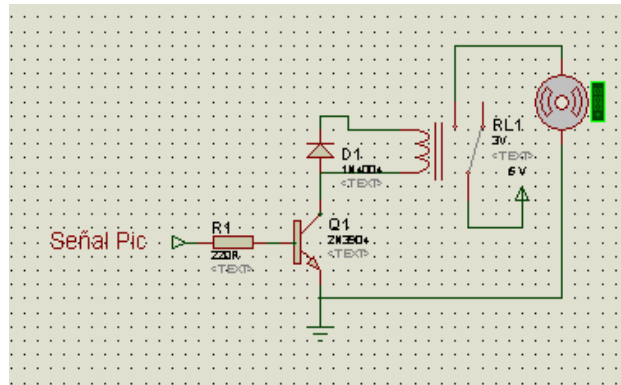


Fig. IV.42: Manejo Motor C.C

4.3.4 Sensores de Obstáculo

El sensor TCRT5000 se encarga de la detección de obstáculos, se trata de un sensor infrarrojo como se estudio en el capítulo 2, este envía un pulso que es ingresado al Pic a través del puerto RB0. La figura IV.43 presenta el esquema del montaje del sensor de obstáculos.

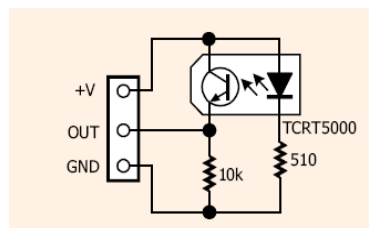


Fig. IV.43: Esquema Sensor de obstáculos

4.4 Programa de aplicación

Esta sección considero una de las partes esenciales en el funcionamiento del robot, porque constituye el proceso a seguir para cumplir ordenes especificas a través de la programación del módulo de reconocimiento de voz y el pic 16f877A.

4.4.1 Requerimientos para programar VR Stamp

La técnica de reconocimiento de voz que se utilizó para la programación del módulo es el reconocimiento independiente del habla (S.I), porque permite interactuar con cualquier usuario, tomando en cuenta que se trata de un robot compañero. Cabe señalar que el kit de programación incluye un CD de la empresa Phyton, con una llave de seguridad como licencia máxima para 60 días, o si queremos comprar la licencia definitiva debemos desembolsar una considerable cantidad de dinero, en este paquete incluye QuickSynthesis4™, que se encarga del tratamiento de audio, y para crear el vocabulario que será reconocido por el robot tenemos la herramienta Quick T2SI.

4.4.1.1 Comando de Inicio y ordenes

El comando de inicio es una frase o palabra que el reconocedor siempre está esperando, y una vez detectada da paso a la selección de un comando. Se debe escoger una palabra o frase con varias sílabas, porque esto permite al reconocedor tener mayor información fonética para comparar con voces o ruido de fondo que está escuchando. El comando de inicio es: Robotrax, los comandos están divididos en tres grupos de ejecución, los primeros realizan la movilidad del robot, los segundos se encargan del control domótico, y los terceros de dialogo, en la figura IV.44 se aprecia la división de estos comandos.

Todos los comandos deben ser anteceditos de la palabra Robotrax, y posteriormente pronunciar el comando de cualquiera de estos grupos, tomando en cuenta que los comandos de movilidad debe considerarse la distancia que recorra el robot, para aumentar el tono de voz y fuerte, y para los comandos 'bailar', 'cantar', 'atrás', son comandos con una fonética parecida por ello se debe pronunciar en forma clara cada orden. Se debe tomar en cuenta que los comandos de control domótico tienen un alcance limitado para encender la televisión o radio a través de radio frecuencia, a una distancia no mayor a los 3 metros.

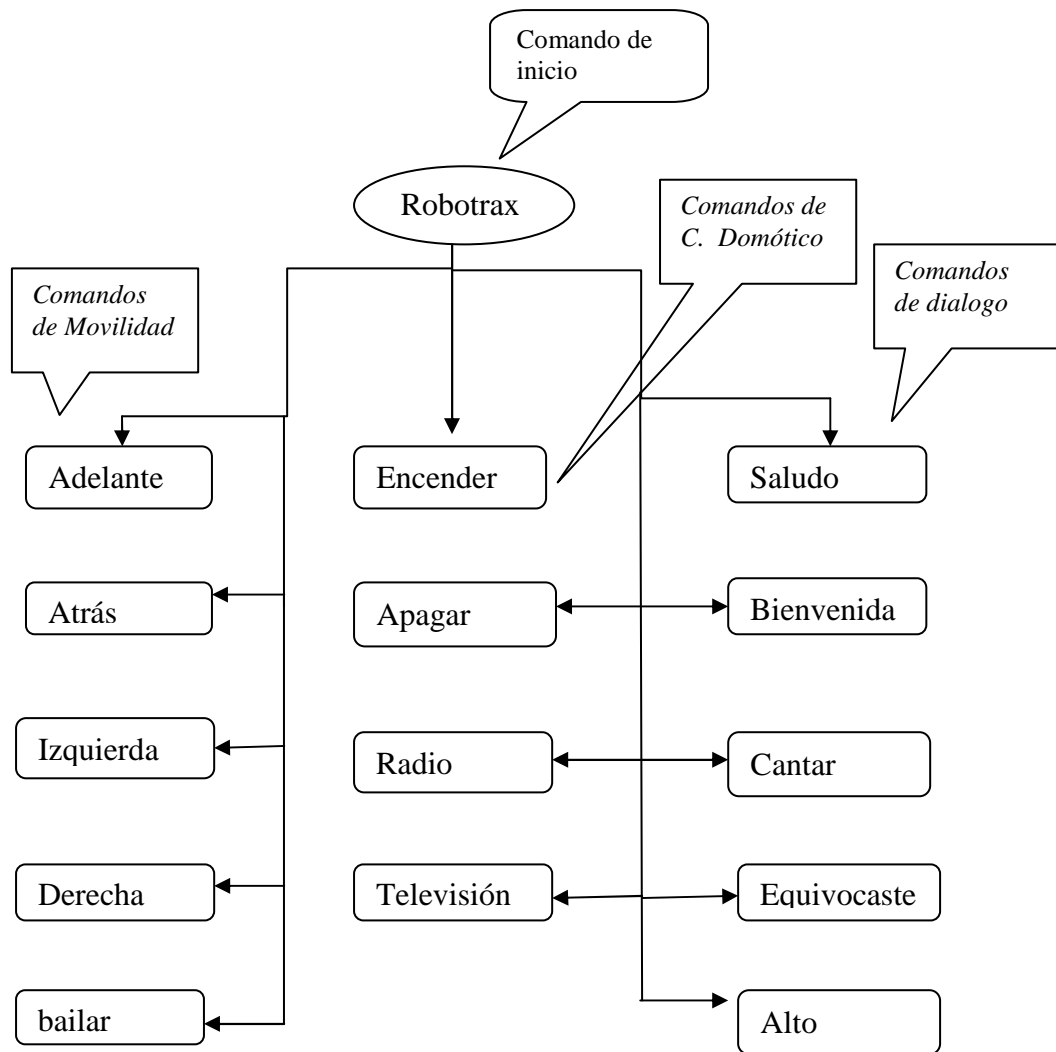


Fig. IV.44: División de comandos

4.4.1.2 Creación de mensajes de voz

Necesitamos generar archivos de audio para hacer al robot mas amigable, de esta manera se interactúa con el robot como si estuviéramos hablando con él, para ello grabamos la voz con algunas especificaciones como por ejemplo en formato wav, a 22 o 50 muestras por segundo, 16 bits, monoaural, y editamos con Nero WaweEditor con esto obtenemos un efecto robótico como se muestra en la figura IV.45.

Se puede utilizar cualquier editor de audio, que cumpla las especificaciones descritas.

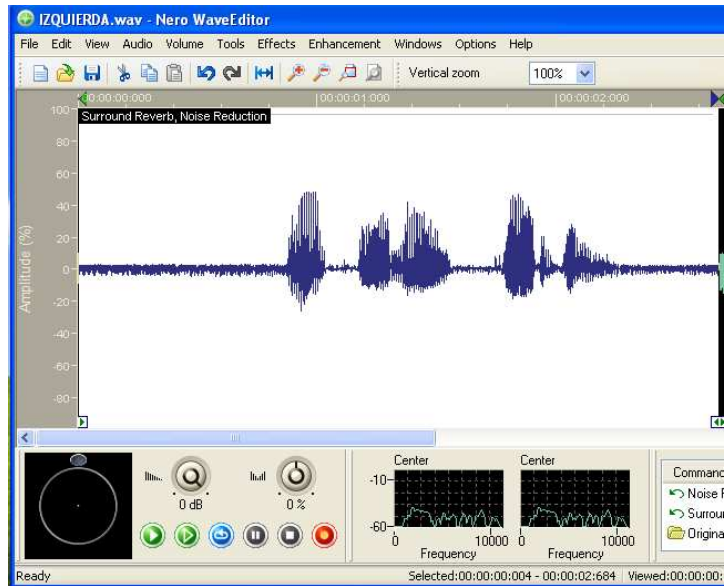


Fig. IV.45: Edición del archivo de audio

Al ejecutar QuickSynthesis4™ en la barra de herramientas opción añadir archivos de audio buscamos los mensajes de voz que hemos creado y editado con anterioridad, si las especificaciones no se cumplen se genera el siguiente error como muestra la figura IV.46.

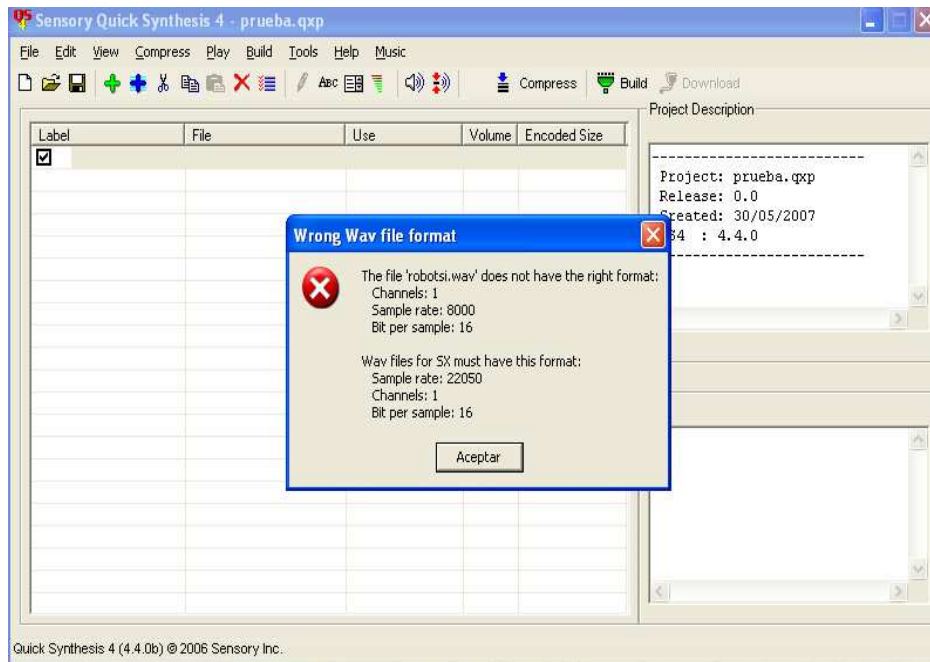


Fig. IV.46: Error en el formato de archivo de audio

Luego de corregir el error cumpliendo con los parámetros establecidos se realiza la compresión del audio a través del icono "Compress", y como ultimo paso presionamos el icono "build" que genera las librerías necesarias que luego llamaremos en la programación principal en Micro C de Phytton; y como comentario adicional se pueden cambiar los nombres de los archivos de audio o dejarlos tal como hemos ingresado originalmente, la figura IV.47 muestra el cuadro de dialogo. Ok individualmente a cada archivo para etiquetarlo o 'ok all' para dejarlo con sus nombres originales.

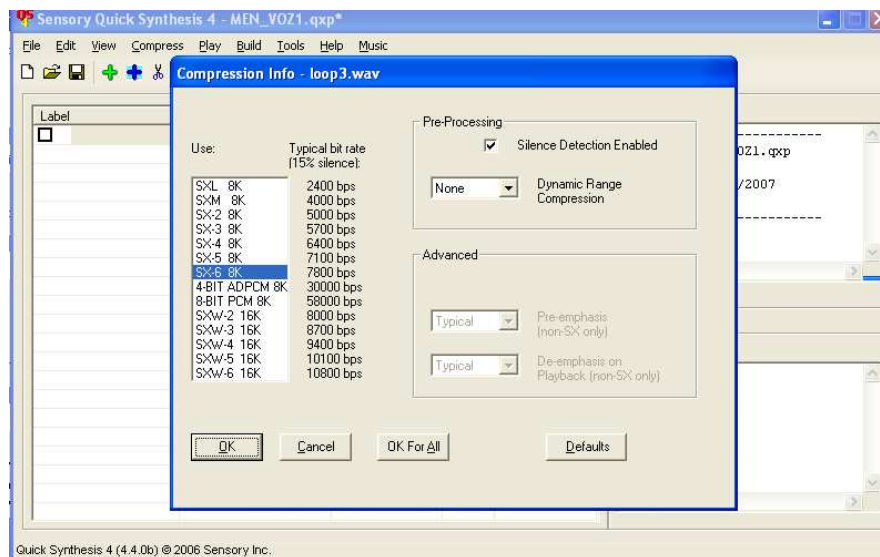


Fig. IV.47: Etiquetado de los archivos de audio

Posteriormente a la construccion de los archivos se generan una tabla de mensajes de voz con un sub indice numerado, significa que podemos llamar al archivo de audio con el numero generado o con el nombre que se etiquetó, la tabla IV.VIII muestra la tabla de sonidos generados.

Tabla IV.VIII: Tabla de sonidos

Nombre Generado	Sub índice	Descripción

Tabla IV.VIII: Tabla de sonidos (Continuación)		
#define SND_ADELANTE	1	Hacia adelante
#define SND_AHORRO	2	Ahorro de energía ejecutando
#define SND_ALTO	3	Alto
#define SND_APAGAR	4	Apagar dispositivo
#define SND_ATRAS	5	Orden recibida hacia atrás
#define SND_AYUDA	6	Diga como palabra de inicio Robotrax luego un comando por ejemplo adelante
#define SND_BAILAR	7	Me gusta bailar
#define SND_BIENVENIDA	8	Bienvenidos
#define SND_CANCION	9	Canta una canción pre editada
#define SND_CANTAR	10	Me gusta cantar
#define SND_COMANDO	11	Diga un comando
#define SND_DERECHA	12	Hacia la derecha
#define SND_DIGA	13	Diga
#define SND_EMPEZAR	14	De una orden
#define SND_ENCENDER	15	Encender dispositivo
#define SND_ERROR	16	Error
#define SND_ERROR_DESC	17	Error desconocido
#define SND_FIE	18	Escuela de Ing. Electrónica
#define SND_GRACIAS	19	Gracias
#define SND_GRACIAS2	20	Es un placer servirle
#define SND_IZQUIERDA	21	Hacia la izquierda
#define SND_NO	22	No

#define SND_PERDON	23	Perdón
#define SND_RADIO	24	Encender radio
#define SND_REPETIR	25	Por favor repita
#define SND_ROBOTRAX	26	Robotrax esta listo para empezar
#define SND_SI	27	Si
#define SND_SPOCH	28	Espoch 2008
#define SND_TV	29	Te Ve
#define SND_VERSION_MALA	30	Versión desconocida
#define SND_pdown	31	Power down
#define SND_00	32	Cero
#define SND_01	33	Uno
#define SND_02	34	Dos
#define SND_03	35	Tres
#define SND_04	36	Cuatro

La tabla IV.1 muestra el siguiente formato: SND_nombre_del_mensaje e índice asignado a cada archivo de audio, se crea un archivo .h, este archivo es una librería utilizada en aplicaciones de lenguaje C, con estos nombres o índices llamaremos un sonido dentro del programa.

4.4.1.3 Creación de comandos a través de Quick T2SI

La herramienta Quick T2SI permite crear vocabularios de reconocimiento de voz independiente, ingresando cada palabra a través del teclado, cuando ejecutamos Quick T2SI se genera un mensaje que indica que solo se puede utilizar el software hasta 180 días o se puede

generar hasta 50 archivos.h lo que suceda primero en la figura IV.48 muestra el mensaje.



Fig. IV.48: mensaje de precaución.

Seleccionamos vocabulario y escribimos el comando de inicio en mi caso Robotrax, luego las palabras a ser reconocidas, en esta misma herramienta se realiza un test con un micrófono conectado a la PC, pronunciamos la palabra de inicio, una línea verde es la que nos indica si la palabra fue reconocida, después pronunciamos un comando y de igual manera una línea verde es la que indica que la palabra fue reconocida tal como se muestra el la figura IV.49.

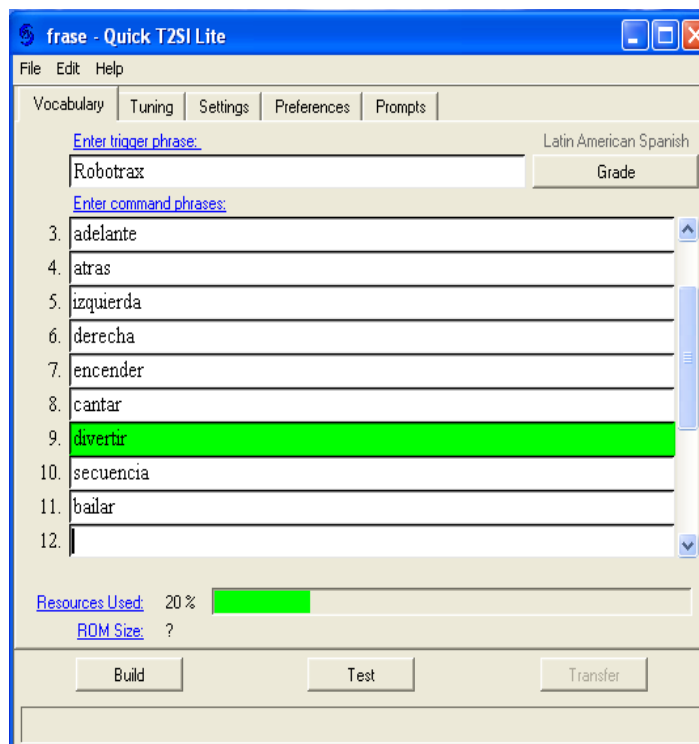


Fig. IV.49: Creación del vocabulario.

A continuación de crear el vocabulario, se realiza la sintonía de palabras, esto quiere decir buscar los fonemas adecuados de manera que la pronunciación del usuario concuerde con la del modelo que se establece con este programa. Con la opción 'select Word' escogemos la palabra que deseamos sintonizar, con la opción 'select pronunciations' se puede escoger la pronunciación sugerida o también se puede editar la pronunciación escogiendo manualmente los fonemas de la palabra seleccionada tal como se puede apreciar en la figura IV.50.

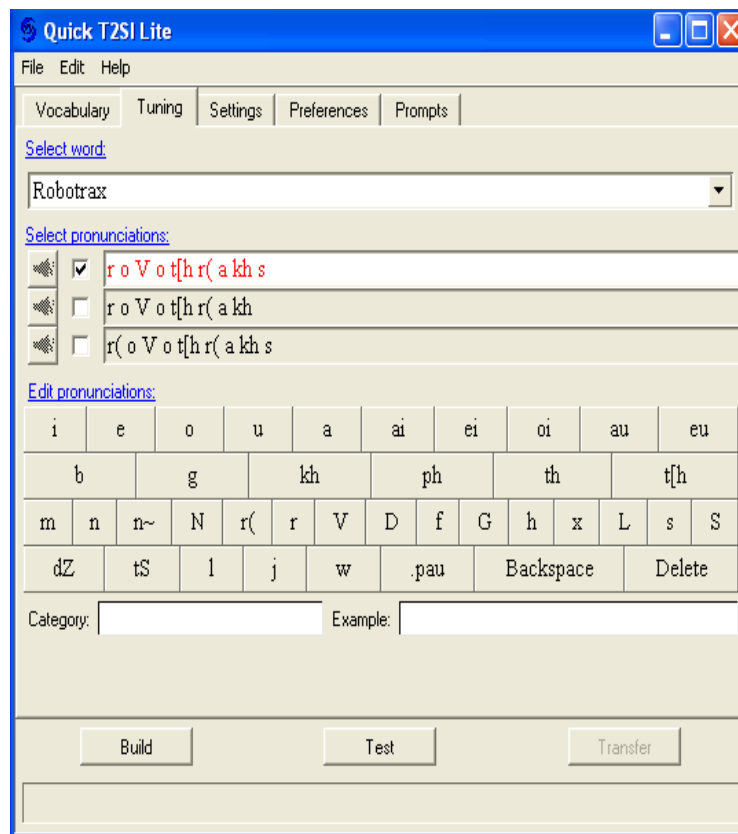


Fig. IV.50: Selección de la pronunciación

Hace no mucho tiempo el VR Stamp solo reconocía vocabulario en Inglés, actualmente esta diseñado para los idiomas de: Inglés, alemán, italiano, japonés y español.

Por defecto el reconocimiento es en inglés para cambiar a español latinoamericano nos valemos de la opción 'settings' en 'Lenguaje' tal como muestra la figura IV.51.

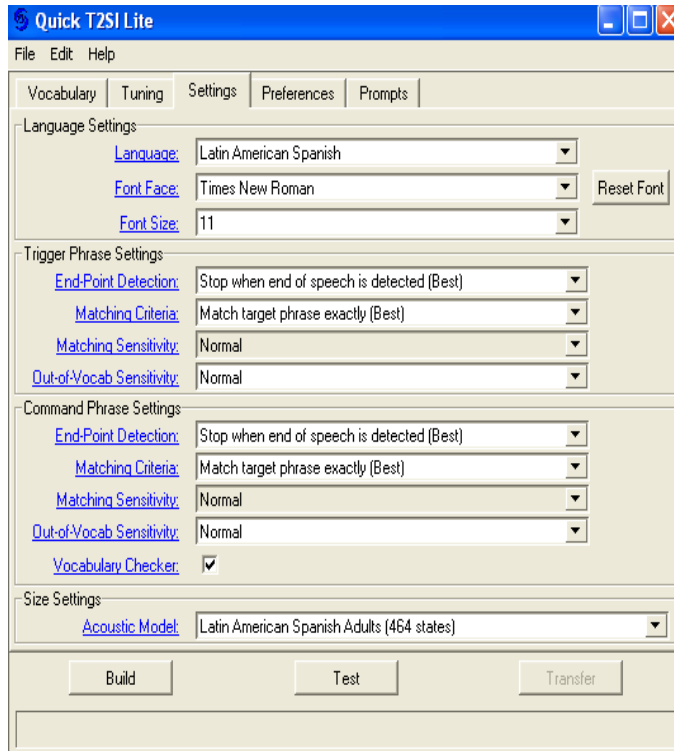


Fig. IV.51: Configuración de parámetros

La frase de inicio y de los comandos permite establecer como se espera reconocer la palabra, existen dos configuraciones. La primera opción espera por un determinado tiempo de silencio, para concluir que el usuario terminó de hablar y espera el resultado del reconocimiento. La segunda opción se detiene inmediatamente cuando la frase es detectada. La principal diferencia entre estos dos criterios es que el primero selecciona el mejor resultado de las más altas calificaciones de cada frase, mientras que el segundo simplemente toma la más alta calificación encontrada. La primera opción es mucho más confiable, puesto que ofrece mejores resultados.

La siguiente opción es el criterio para comparar la frase detectada con la del modelo. Por ejemplo, el comando bailar, entonces dentro de este contexto, se podría pronunciar ‘por favor bailar’, teniendo la posibilidad de que el reconocedor encuentre el comando dentro de la frase. En nuestro caso, siempre vamos a esperar la frase o palabra que se va a reconocer (sin esperar palabras adicionales).

La mejor opción es la de reconocer palabras sueltas garantizando mayor probabilidad de reconocer una palabra.

Existen otras opciones que la dejamos por defecto porque el mismo programa sugiere cual es la mejor. Otra opción que se puede modificar es la edad para quien va dirigido el reconocimiento de voz, en mi caso esta hecho para personas adultas con 460 estados, este modelo acústico es mucho más preciso.

Una vez probado y configurado correctamente presionamos 'Buil' para generar las librerías. En la tabla IV.IX se muestra el comando de inicio y la tabla IV.X muestra el vocabulario a ser reconocido.

Tabla IV.II: Definición palabra de inicio e índice asignado

Definición del vocabulario de inicio	Índice asignado
#define G_trig_frase_SILENCE	(0)
#define G_trig_frase_ROBOTRAX	(1)
#define G_trig_frase_nota	(2)

Tabla IV.X: Asignación de índices al vocabulario

Definición de los comandos	Índice asignado
#define G_comm_frase_SILENCE	(0)
#define G_comm_frase_SALUDO	(1)
#define G_comm_frase_BIENVENIDA	(2)
#define G_comm_frase_ADELANTE	(3)
#define G_comm_frase_ATRAS	(4)
#define G_comm_frase_IZQUIERDA	(5)
#define G_comm_frase_DERECHA	(6)
#define G_comm_frase_ENCENDER	(7)

Tabla IV.X: Asignación de índices al vocabulario(Continuación)	
#define G_comm_frase_CANTAR	(8)
#define G_comm_frase_DIVERTIR	(9)
#define G_comm_frase_SECUENCIA	(10)
#define G_comm_frase_BAILAR	(11)
#define G_comm_frase_TELEVISION	(12)
#define G_comm_frase_RADIO	(13)
#define G_comm_frase_EQUIVOVASTE	(14)
#define G_comm_frase_ALTO	(15)
#define G_comm_frase_APAGAR	(16)
#define G_comm_frase_nota	(17)

4.4.1.4 Programación VR Stamp

Hasta este punto se ha generado las librerías necesarias para programar el módulo de reconocimiento de voz VR Stamp, y se cuenta con todas las herramientas necesarias para generar el archivo.hex, este archivo se obtiene compilando en micro C, de la empresa Phytton, se puede generar el archivo.hex en C o ensamblador con otros compiladores pero no tenemos la oportunidad de manejar comandos que son específicos para tratar con el reconocimiento de palabras.

Si estamos fuera de la carpeta de sensory donde vienen adjuntas las librerías necesarias para compilar el software necesitamos corregir los Paths.

Si deseamos crear nuestra propia carpeta para generar el archivo.hex debemos tomar en cuenta las siguientes sugerencias:

Se debe borrar las librerías erróneas y añadir desde C:\sensory\FC2_0_14, esta carpeta es la que viene en fluent chip, y tiene que ser anexada conjuntamente a phyton, claro depende de la versión en mi caso particular se encuentra en la carpeta FC2_0_14.

Otra sugerencia es ir a la barra de herramientas en la opción 'project' – Project options – en el ítem folders corregir los paths. La figura IV.52 muestra los cuadros de dialogo necesarios para cambiar los paths, el cambio debe ser por:

C:\Phyton\Proj-se\1_10_09\MCCSE\INC;c:\sensory\FC2_0_14\include

C:\Phyton\Proj-se\1_10_09\MCCSE\LIB;c:\sensory\fc2_0_14\lib

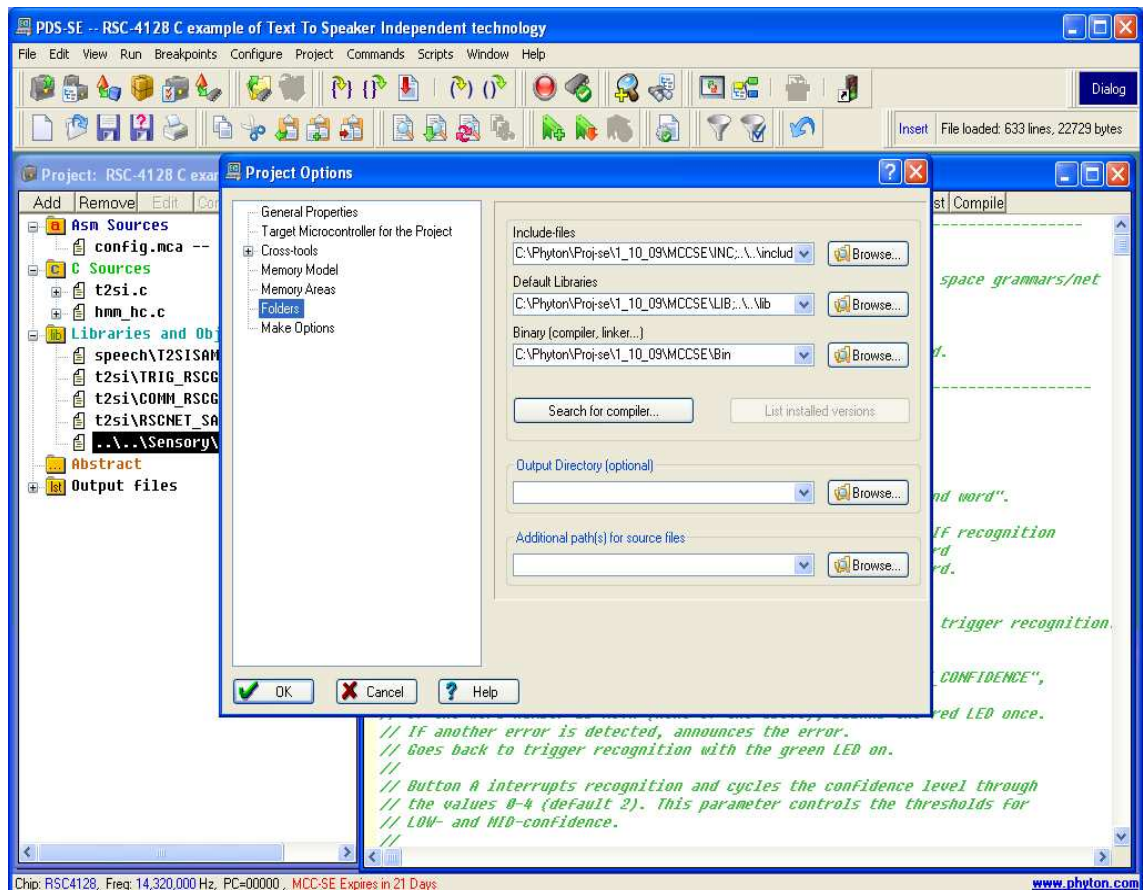


Fig. IV.52: Corrección de paths

Cuando estas librerías no son incluidas se genera el error que muestra la figura IV.53. Para que la compilación tenga éxito corregimos el path.

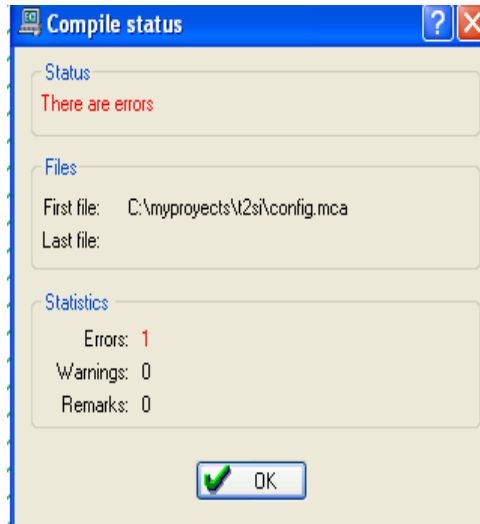


Fig. IV.53: Mensaje de error

Como ultimo paso añadimos las librerías que hemos creado con la herramienta QuickSynthesis4™ y Quick T2SI, con solo dar click derecho en librerías y añadir. La figura IV.54 muestra la compilación correcta y satisfactoria para generar el archivo.hex.



Fig. IV.54: Mensaje de compilación correcto

4.4.1.4.1 Código fuente VR stamp

Tenemos listo el editor para ingresar el código fuente, pero antes observemos la lógica de programación de la figura IV.55.

El diagrama de flujo muestra el funcionamiento lógico que debe tener el programa.

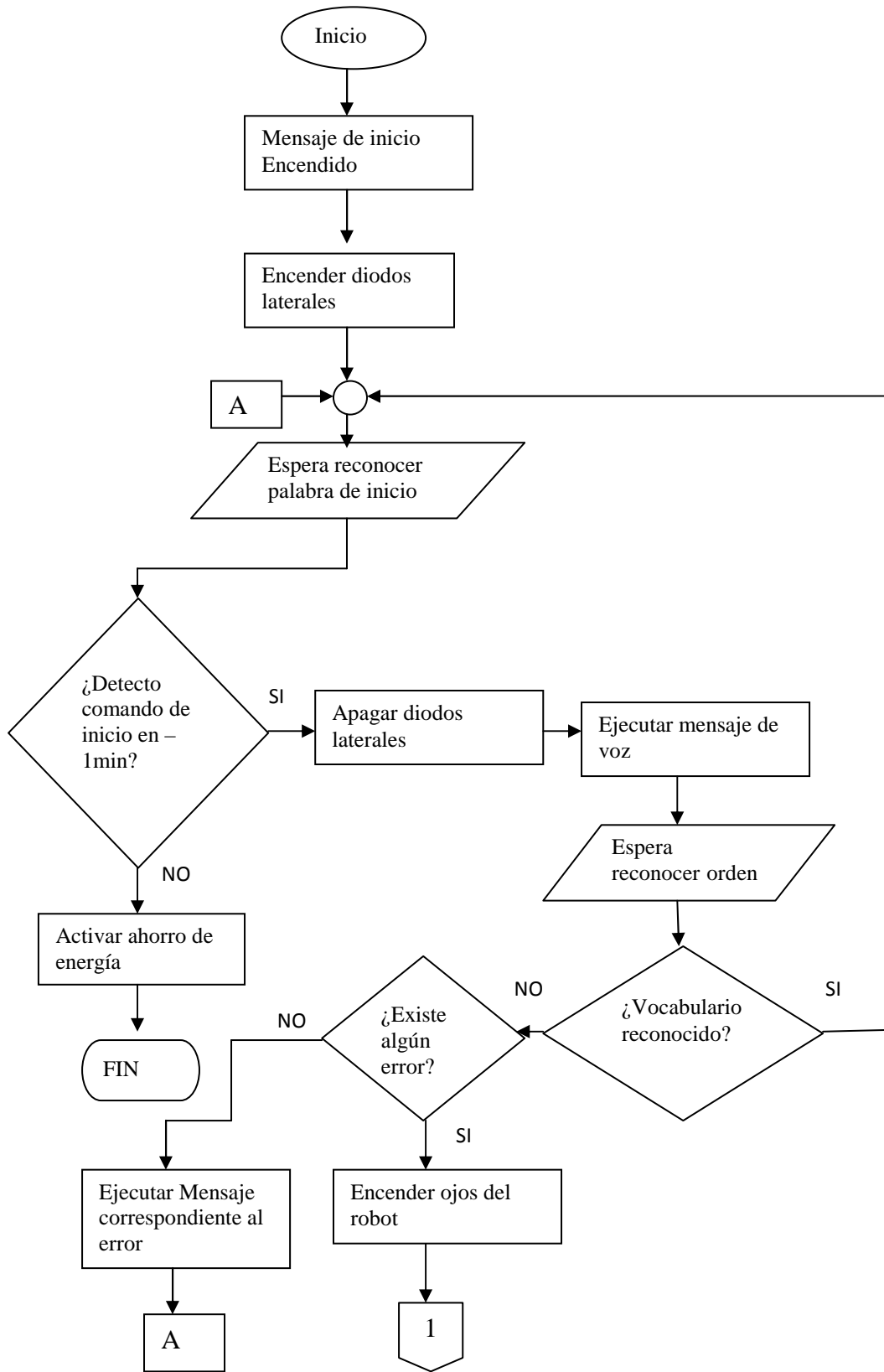


Fig. IV.55: Diagrama de flujo VR Stamp

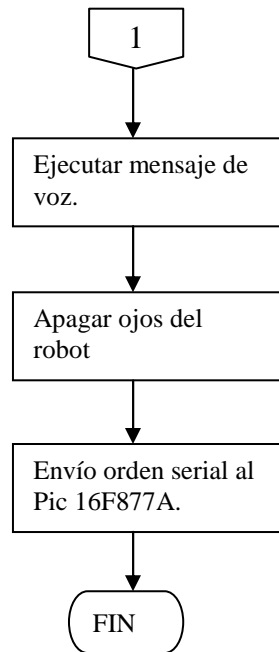


Fig. IV.55: Diagrama de flujo VR Stamp (Continuación)

El código fuente correspondiente al módulo de reconocimiento de voz VR Stamp se basa en micro C de phyton.

4.4.1.4.2 Análisis de los parámetros utilizados en el código fuente

La descripción de algunas palabras reservadas y funciones y sintaxis de Micro C de la empresa Phyton se describe brevemente a continuación.

`_PlaySnd`: Esta función permite escuchar sonidos que fueron creados con la herramienta Quick Synthesis™.

Sintaxis: `uchar _PlaySnd(uint soundIndex, long soundTableAddr, uint volume);`

Parámetros:

`uint soundIndex`: corresponde al índice del sonido perteneciente a la tabla de sonidos.

`long soundTableAddr`: corresponden a 32 bits, que indican la dirección en memoria de la tabla de sonido. Los bits desde el 21 al 32, deben estar puestos a cero. Si el bit 20, se encuentra en

alto, indica que la tabla se encuentra en memoria externa (Data Space), caso contrario está en memoria interna (Const space).

uint volume: permite seleccionar el volumen, el rango esta entre 1 y 256, donde 256 es máximo volumen.

Valores de Retorno:

ERR_OK: el mensaje se completo exitosamente

ERR_IN_SYNT: el usuario interrumpió el mensaje

ERR_SYNT_BAD_VERSIO: la tabla es incompatible con la versión

ERR_SYNT_BAD_MSG: error en el archivo de voz

_T2SI

Propósito: Reconocimiento del habla, dependiente del vocabulario, independiente del locutor, utilizando los Modelos Ocultos de Markov.

Sintaxis: uchar _T2SI(long acousticModel, long grammarModel, uchar knob, char timeout, uchar trailing, PARAMETERPASS *pRes)

Parámetros:

AcousticModel: corresponde a la dirección del modelo acústico.

GrammarModel: corresponde a la dirección del modelo gramatical

Knob: corresponde al nivel de confiabilidad para el reconocimiento de voz, el nivel se puede escoger entre 0 a 4, siendo 0 el nivel más pobre, 2 el nivel típico y 4 el nivel más alto. Este parámetro no es considerado en el reconocimiento de un comando de inicio.

Timeout: corresponde al tiempo que se espera por un comando, el valor seleccionado puede estar entre 1 a 255, donde cada unidad representa un segundo. El tiempo de espera recomendado para un comando es 3 segundos, y 60 segundos para un comando de inicio.

Trailing: corresponde al tiempo mínimo en silencio que se espera para concluir que se terminó de hablar. Este parámetro se puede seleccionar desde 4 a 36, donde cada unidad representa

0.025 segundos. El valor típico es 15, equivale a 400ms, el valor mínimo es 100ms y el valor máximo es 900ms.

PRes: corresponde un puntero a la estructura de resultados Valores de Retorno:

ERR_OK: significa reconocimiento con alta confiabilidad ERR_RECOG_LOW_CONF: significa que el resultado del reconocimiento tiene baja confiabilidad.

ERR_RECOG_MID_CONF: significa que el reconocimiento tiene un nivel aceptable de confiabilidad.

ERR_DATACOL_TIMEOUT: significa que no hubo reconocimiento dentro del tiempo esperado.

ERR_INT_BLOCK: significa que el usuario interrumpió el reconocimiento Results.pp_a: corresponde a un error.

Results.pp_b: contiene un índice que representa la palabra reconocida (class ID) o una palabra reconocida pero que no concuerda con ninguna del modelo (NOTA), es decir, una palabra fuera del vocabulario.

`_Init232`

Propósito: Habilita o deshabilita la transmisión serial

Sintaxis: `void _Init232()`

Parámetros: ninguno

Valor de Retorno: ninguno

`_SetBaudRate`

Propósito: Permite seleccionar la velocidad de transmisión

Sintaxis: `_SetBaudRate(velocidad)`

Parámetros:

Velocidad: representa el valor de velocidad deseado, por ejemplo, BAUD9600

Valor de Retorno: ninguno

`_Put232`

Valor de Retorno: ninguno

`_Put232`

Propósito: envía un byte para transmitir de manera serial

Sintaxis: `_Put232(uchar val)`

Parámetros:

Val, byte a ser enviado

Valor de Retorno: ninguno

`_Wait232`

Propósito: espera recibir un byte

Sintaxis: `uchar _Wait232()`

Parámetros: ninguno

Valor de Retorno: Byte recibido.

`_Wait10MilliSecs`

Propósito: espera por un número determinado de intervalos de 10 ms

Sintaxis: `_Wait10MilliSecs(uchar val)`

Parámetros:

Val: número de intervalos de 10 ms, donde el valor mínimo es 0 y el valor máximo es 255

Valor de retorno: ninguno

`_SleepIO`

Propósito: Permite enviar a modo de bajo consumo, o regresar del modo de bajo consumo.

Sintaxis: `uchar _SleepIO (uchar port, uchar bits, uchar states)`

Parámetros:

Port: puerto utilizado para regresar del modo de bajo consumo, puede ser 0 o 1

Bits: bit del puerto seleccionado para regresar del modo de bajo consumo

Este tipo de formato es de gran utilidad para el ahorro de energía.

States: estado del bit para regresar del modo de bajo consumo, por ejemplo, si se selecciona 0, entonces, se espera un estado bajo para regresar del modo de bajo consumo

Valor de Retorno:

ERR_OK = se envió o regresó del modo de bajo consumo sin problemas.

ERR_NO_OK = existe un error de configuración o ya se regreso del modo de bajo consumo.

La librería <demo_io.h> permite controlar el estado de los bits del 4 al 7 del puerto dos, también permite manejar interrupciones mediante los bits del 0 al 4 del puerto uno.

La librería <MEN_VOZ1.h>, es creada por el programa Quick Synthesis™, este archivo contiene la lista de sonidos, que se indica en la tabla IV.I. Este archivo se debe añadir al programa para poder seleccionar cualquier sonido que se desee emitir.

La librería <trig_rscApp_frase.h>, es creada por el programa Quick T2SI y permite incorporar el comando de inicio en el programa. La librería <comm_rscApp_frase.h> son creadas por el programa Quick T2SI y permiten incorporar los vocabularios dentro del programa.

4.4.1.4.3 Descarga del archivo.hex

El archivo generado al compilar en C, debemos descargar en el C.I VR Stamp, para ello nos valemos de Sensory Loader es otro programa que permite cargar archivos.HEX o archivos.BIN, para este trabajo la descarga se realizó en .hex, antes configuramos el puerto donde se conecta el usb, en el momento de la instalacion debemos tomar en cuenta en que puerto se conecto el cable debe ser conectado y configurado exactamente donde se instalo caso contrario genera un error.

Existe un limite en la velocidad de carga, para no tener inconvenientes se estableció en 19200 Baudios, en File es el path del archivo.hex y presionamos download, esperamos el mensaje 'descarga completada' tal como se muestra en la figura IV.56 y esta listo para ser utilizado, este programa tambien genera un archivo.bin que sirve igual que el .hex.

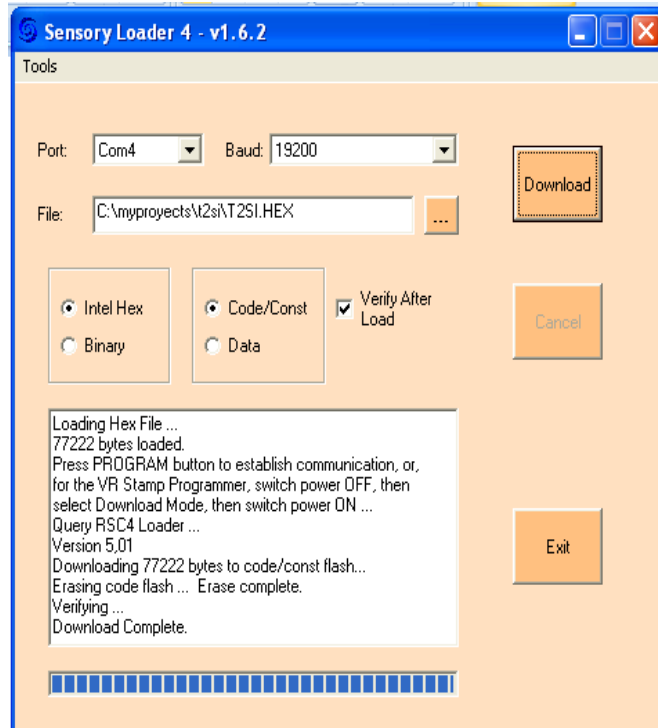


Fig. IV.56: Cuadro de dialogo para la carga del archivo.hex

4.4.2 Programación pic16f877A

Para la programación de este microcontrolador se utilizó microcode, el código fuente se puede observar en el anexo III, la lógica de programación utilizada se muestra en la figura IV.57.

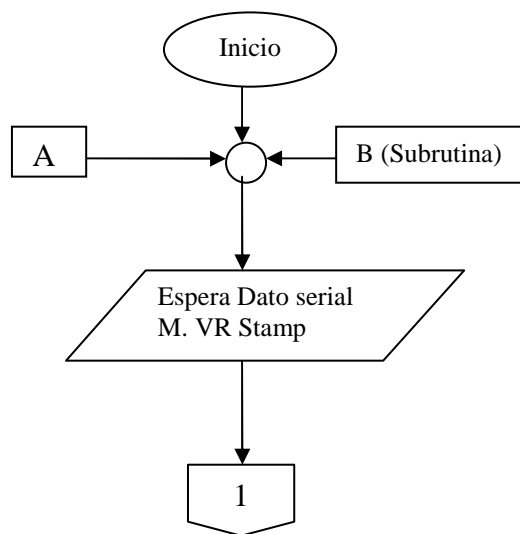


Fig. IV.57: Diagrama de flujo Lectura serial

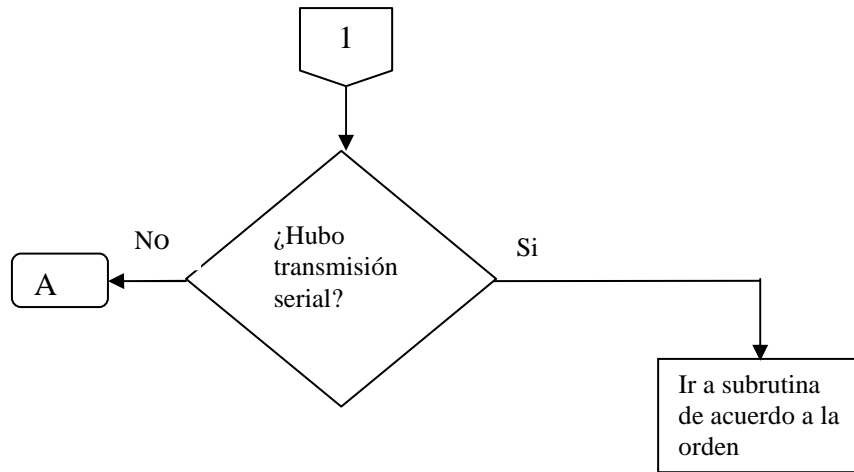


Fig. IV.57: Diagrama de flujo Lectura serial (Continuación)

4.4.2.1 Subrutina ade: (Adelante)

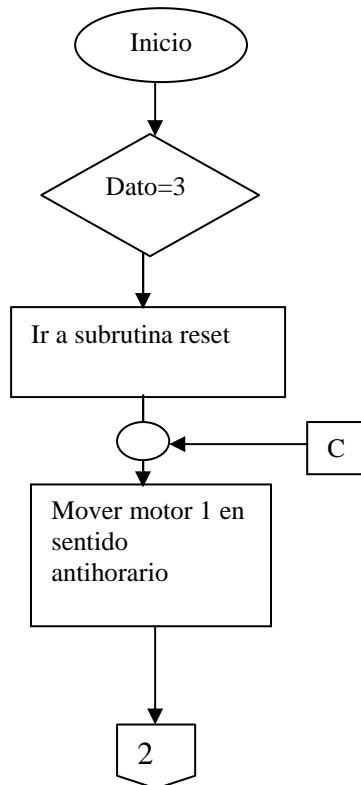


Fig. IV.58: Subrutina Adelante

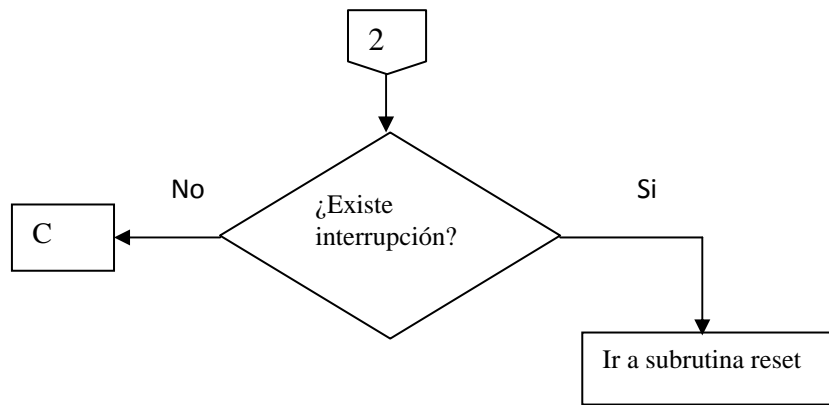


Fig.IV.58: Subrutina Adelante (Continuación)

4.4.2.2 Subrutina atr: (Atrás)

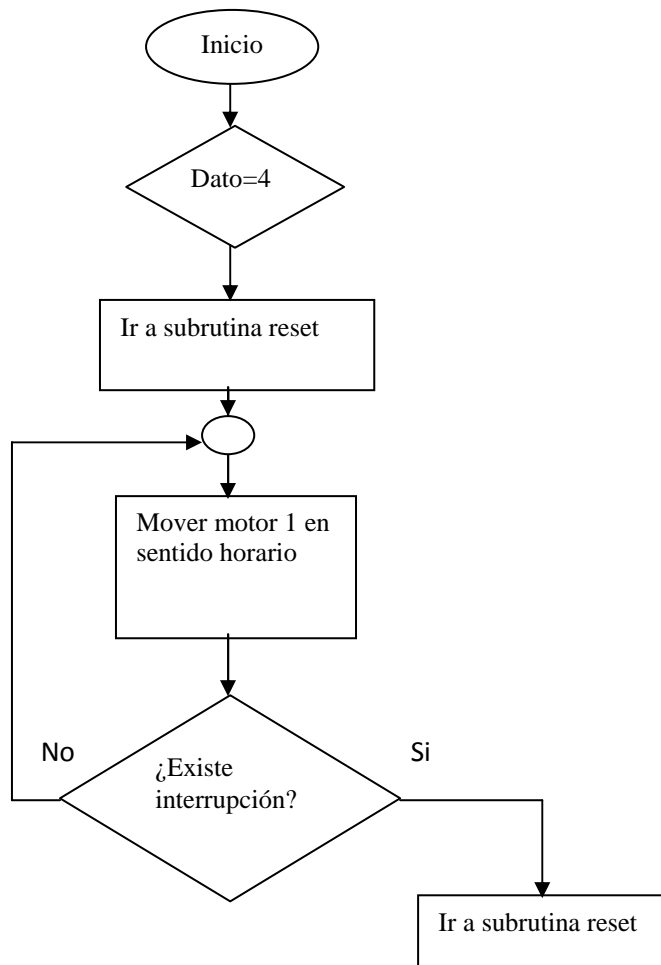


Fig. IV.59: Subrutina atrás

4.4.2.3 Subrutina izq: (Izquierda)

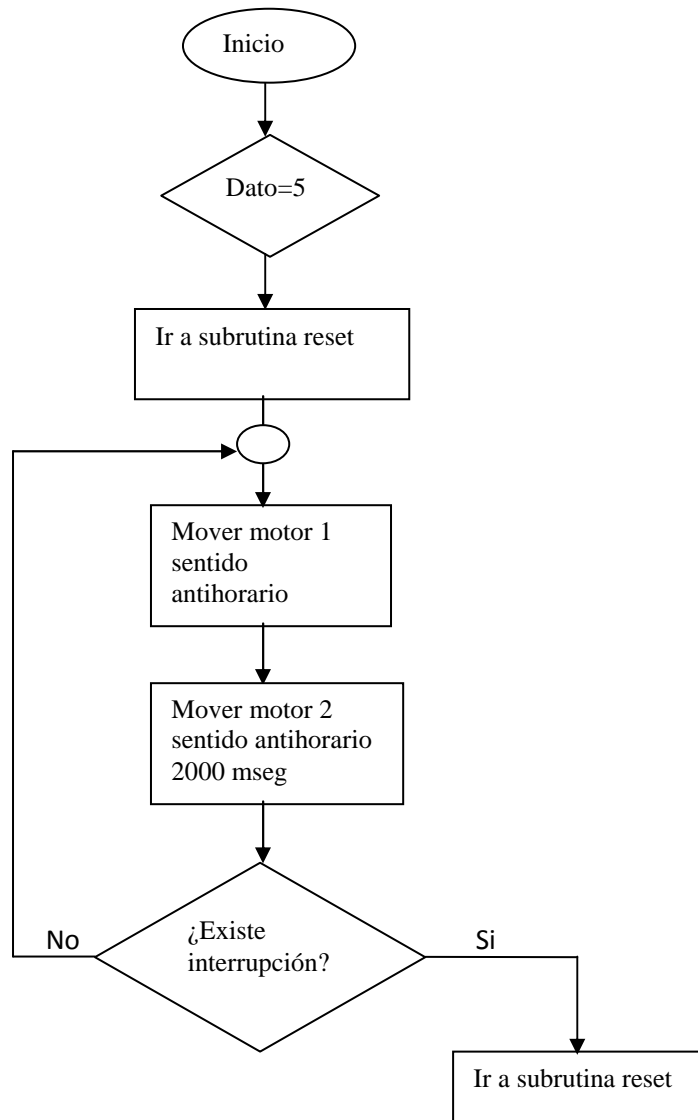


Fig. IV.60: Subrutina Izquierda

4.4.2.3 Subrutina der: (derecha)

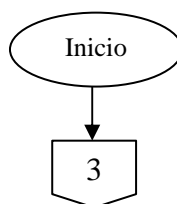


Fig. 61: Subrutina derecha

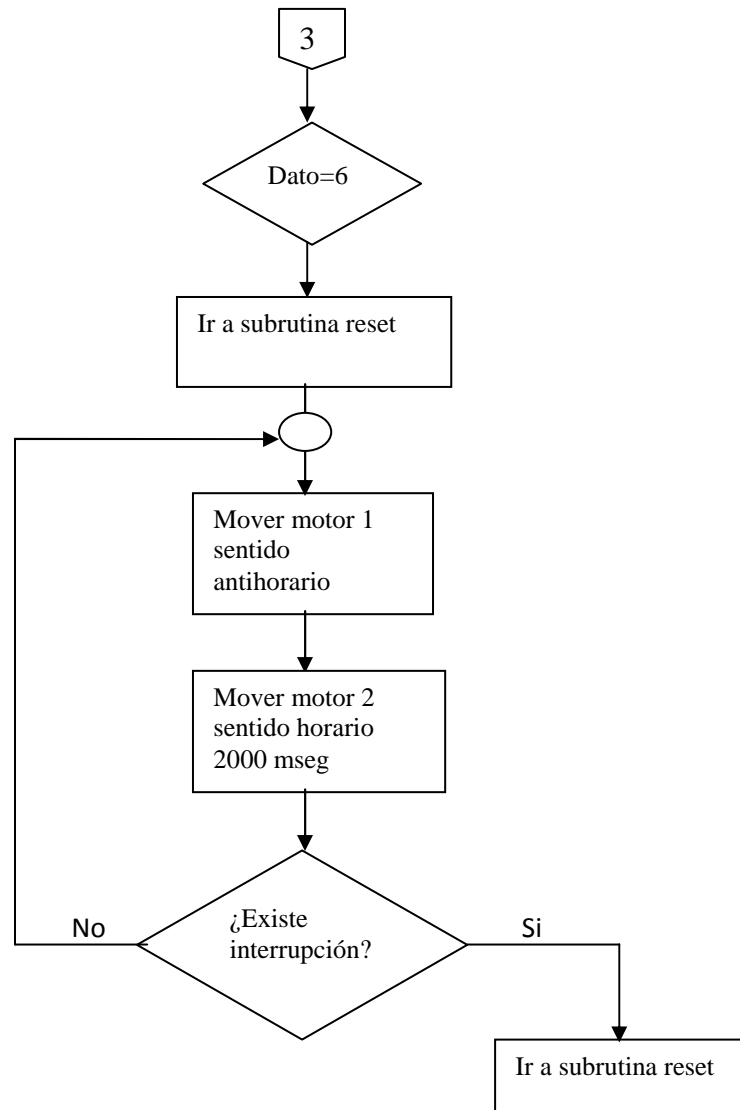


Fig.IV.61: Subrutina derecha (Continuación)

4.4.2.4 Subrutina reset

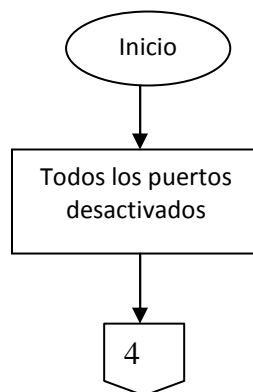


Fig. IV.62: Subrutina reset

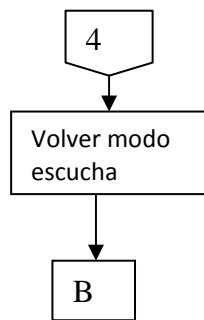


Fig.IV.62: Subrutina reset(Continuación)

4.2 Evaluación Del Módulo Reconocimiento De Voz

Para la evaluación del módulo de reconocimiento de voz implementado en el robot, se realizan pruebas con ambientes ideales, con ruido moderado y niveles altos de ruido ambiental.

4.2.1 Evaluación en ambientes Ideales

En ambientes ideales y luego de probar primero con los comandos de dialogo se establece que el nivel de confiabilidad alcanza entre 90% y 100%, sin pronunciar palabras con fonética parecida. Los comandos de control doméstico tienen el mismo nivel de aceptación que los comandos de dialogo, mientras que los comandos de movilidad cambian la confiabilidad especialmente entre el comando 'atrás' y 'bailar' que tienen niveles acústicos parecidos la confiabilidad para este grupo de comandos esta entre un 75% a 85%, se debe principalmente a que; al moverse el robot por ejemplo 'adelante' el usuario se aleja del micrófono del módulo de reconocimiento de voz haciendo que ya no presten las condiciones ideales para su control. Los dos grupos de comandos se realizaron a una distancia entre 60 cm y 1 m, del micrófono con respecto al usuario, mientras que los de movilidad estuvieron entre los 2m de distancia.

4.2.2 Evaluación en ambientes con ruido moderado

Con ruido moderado, la palabra de inicio Robotrax, evita que se ejecuten algún comando sin

haber pronunciado, exitosamente ha rechazado palabras que no concuerdan con la palabra de inicio, sin embargo en los comandos anidados algunos de estos comandos tienen aceptación al 100% y no necesariamente pertenecen a un mismo grupo, por ejemplo 'radio' es una palabra que presenta este nivel de confiabilidad, pero a nivel general los comandos de grupo de dialogo se encuentran entre 80 a 90% de aceptación, de igual manera con los comandos de control domótico, mientras que los comandos de movilidad presentan mayor dificultad en la ejecución tienden a reconocer otro comando, por lo tanto la aceptación está entre 60 al 70% de confiabilidad.

4.2.3 Evaluación en ambientes con ruido Alto

Con niveles de ruido alto como por ejemplo encendida la televisión con un volumen aceptable de igual manera un radio, el nivel de confiabilidad cambia drásticamente, haciendo que se disparen los comandos sin haberlos pronunciado, el comando que más tiende a ejecutarse y reconocer erróneamente es 'bailar', tiene esta vulnerabilidad porque su fonética no es la apropiada, en este tipo de ambientes no es apto para ser usado por su alto porcentaje de error.

Sin embargo en niveles moderados o de bajo ruido su funcionamiento es aceptable, pese a esto se debe tomar en cuenta que se puede configurar para que el reconocimiento sea más estricto y evite ejecutar comandos no pronunciados.

El transmisor y receptor de radio frecuencia de 47 MHz, utilizado para el control de electrodomésticos alcanzó una eficiencia del 100% a una distancia de 2 m, se llega a transmitir hasta 3 m, sin embargo varía por factores externos como línea visual, orientación de la antena, lluvia, granizo, paredes, entre otros.

CONCLUSIONES

1. El módulo de reconocimiento de voz VR Stamp, es apropiado para el control por comandos de voz, posee audio preamplificada lo que permite interactividad con el robot a través de mensajes de voz emitidos por este, el usuario logra escuchar si la orden que se dio es la correcta o no.
2. El nivel de confianza para el reconocimiento de voz es del 90%, sin embargo varía por factores externos como por ejemplo el ruido del ambiente, la pronunciación debe ser clara y las palabras a ser reconocidas no deben tener similitud de fonemas.
3. Se logra disminuir la influencia del ruido ambiental configurando algunos parámetros a través de software, del mismo módulo (vr stamp) pues posee internamente filtros para disminuir este efecto.
4. El modulo de reconocimiento de voz (vr stamp), utiliza conceptos híbridos: Modelos Ocultos de Markov (HMM) para reconocer texto independiente (T2SI) y Pattern-based (Método basado en patrones), estos son utilizados para el reconocimiento dependiente del usuario, y verificación de reconocimiento (SV).
5. La palabra de inicio para el reconocimiento de voz debe ser un modelo acústico que tenga mayor información fonética, en este proyecto se utilizó la palabra "Robotrax", esta combina varias silabas especialmente 'trax', que hace que el reconocimiento sea mas estricto, permitiendo que en ruidos ambientales altos no se disparen comandos que no fueron pronunciados.

6. Se utilizó el pic16f877A, como protección para el módulo de reconocimiento de voz, pues este no está disponible en nuestro país (su importación tuvo algunos contratiempos), el VR Stamp envía una señal serial asíncrona hacia el pic16f877a para el control de los motores, y no directamente a través de sus puertos disminuyendo su manipulación para que no sufra daños.

7. El uso de un sistema de engranajes adecuado permitió ganar en torque y reducir la velocidad en los motores de C.C, consiguiendo la movilidad adecuada del robot.

RECOMENDACIONES

1. Se recomienda el uso del módulo de reconocimiento de voz VR Stamp, para aplicaciones combinadas entre reconocimiento independiente, dependiente y verificación del locutor de esta manera se explota su capacidad máxima de las utilidades con que cuenta este módulo, es pequeño en tamaño pero con grandes contenidos tecnológicos.
2. Se recomienda que al ser reconocida la palabra de inicio se emita un mensaje de voz, para que el usuario sepa cuando ejecutar una orden y encender un diodo led durante el tiempo de espera del comando, de esta forma el usuario sabrá si está en modo escucha, ejecución de la orden o tiene que volver a pronunciar la palabra de inicio.
3. Para obtener mejores resultados se recomienda acoplar adecuadamente el micrófono con el módulo, y hablar desde una distancia de 60 cm a 1m, fuerte y claro y en lo posible en ambiente sin ruido.
4. El uso del driver L293D prestó las facilidades para el control de los motores de C.C, la desventaja es su consumo de corriente alto, como reemplazo se puede utilizar relés a 3V, que se encuentra fácilmente en el mercado.
5. Para abaratar costos, se sugiere material reciclado, como por ejemplo para la transmisión inalámbrica se utilizo transmisor receptor de pequeños juguetes sin uso.
6. Para imprimir las pistas de los circuitos se podría utilizar hojas de la revista Dinners reciclada (no importa que se imprima sobre otros gráficos).

RESUMEN

El diseño del control electrónico en esta tesis, tiene como finalidad implementar nuevas tecnologías en el reconocimiento de voz en un prototipo de robot, a la vez que sirva para el control automático de electrodomésticos.

Se utilizó microcontroladores, transmisor y receptor de radio frecuencia a 47 MHz; el módulo de reconocimiento de voz VR Stamp de la empresa Sensory; se aplicó método científico analítico para implementar tecnologías híbridas como los Modelos Ocultos de Markov (HMM) y la técnica basado en patrones.

La nueva técnica en el robot tuvo un 90% de efectividad en ambientes ideales, a una distancia del micrófono y usuario de 60 cm a 1m. Con ruidos moderados se obtuvo una efectividad del 90% con los mismos requerimientos de distancia. Para ruidos extremos, los comandos fueron erróneamente detectados bajando su efectividad al 60%. El transmisor y receptor de radio frecuencia utilizado para el control de electrodomésticos alcanzó una eficiencia del 100% a una distancia de 2 m, se llega a transmitir hasta 3 m, sin embargo varía por factores externos como línea visual, orientación de la antena, lluvia, granizo, paredes, entre otros. El robot se mueve en todas direcciones, evitando obstáculos a través de sus sensores infrarrojos hasta una distancia de 3 cm, con los mecanismos de tracción alcanza una efectividad del 95 % en movilidad.

La tecnología implantada alcanzó gran desempeño en ambientes de bajo ruido.

Para lograr mayor eficiencia en ambientes de alto ruido, se debe investigar otras tecnologías de reconocimiento de voz dependiente del locutor.

SUMMARY

The electronic control of this thesis deals with implementing new technologies in recognizing the voice in a robot prototype; at the same time it will serve for the automatic control of appliances. Microcontrollers transmitter and receiver with a radio frequency at 47 MHz and the voice recognition module VR stamp of the sensory enterprise were used. The analytical scientific method was applied for implementing hybrid technologies such as the Hidden Models of Markov (HMM) and the pattern – based technique.

The new technique in the robot had 90% effectiveness in ideal environments at a distance of the microphone and user of 60 cm to 1 m. With moderate noises an effectiveness of 90% was obtained with the same distance requirements. For external noises the commands were erroneously detected lowering their effectiveness by 60%.

The radio frequency transmitter and receiver used for the control of appliances reached 100% effectiveness at a 2m distance. It is possible to transmit up to 3 m; however transmission varies because of external factors such as visual line antenna orientation rain, hail and walls among others. The robot moves in all directions, avoiding obstacles, through its infrared sensors up to distance 3 cm. with the traction devices it reaches 95% effectiveness in mobility. The implanted technology reached a great performance in low-noise environments. To attain a major effectiveness in high-noise environments, other technologies of voice recognition depending on the speaker must be investigated.

GLOSARIO

Acústica: referente a la producción o propagación del sonido.

Amperio: Unidad de medición de la corriente eléctrica (A).

Androide: Autómata de figura de hombre.

Aplicación: Programa que realiza una serie de funciones y con el cual trabajamos en el ordenador.

Arquitectura: Término que se refiere al tipo de estructura hardware de la máquina y que también se aplica a la clasificación de los microprocesadores o el tipo de ranuras de expansión.

Asíncrono: Término utilizado para especificar la ejecución de distintos procesos de forma independiente unos de los otros respecto al tiempo.

Baudio: Unidad de medida. Número de cambios de estado de una señal por segundo.

Binario: Código básico de la informática que reduce todo tipo de información a cadenas de ceros y unos, que rigen las instrucciones y respuestas del microprocesador.

Bit (binary digit): Unidad básica de información representada por ceros y unos que se van sucediendo para conformar los distintos significados.

Cargador: Programa encargado de cargar las distintas líneas de código de un programa ejecutable en la memoria principal del ordenador para su posterior ejecución.

Chip: Circuito integrado y encapsulado.

Corriente alterna: (CA) Corriente eléctrica que cambia su amplitud en forma periódica con el tiempo.

Corriente continua: Modo de suministro de energía eléctrica donde la polaridad de la tensión se mantiene constante. (Caso contrario a la corriente alterna).

Corriente: Cantidad de carga que circula por un conductor por unidad de tiempo.

Frecuencia: Número de veces que se repite un proceso periódico por unidad de tiempo.

Hardware: Componentes materiales propios del computador, sean mecánicos eléctricos o electrónicos.

Módulo: Son procedimientos que nos permiten la realización de una tarea deseada.

Ohm: Unidad de medición de la resistencia eléctrica, representada por la letra griega Ω .

Path: Camino, es la dirección de un archivo o directorio.

PC: Personal Computer (Computador Personal).

Programa: Conjunto de órdenes, palabras y símbolos producto de un análisis efectuado, creado expresamente para ordenar de modo racional los pasos que una computadora ha de dar para realizar un trabajo.

PWM: siglas en inglés para modulación por ancho de pulso.

Resistencia: Es la medida de cuanto se opone un circuito al paso de la corriente eléctrica a través de él.

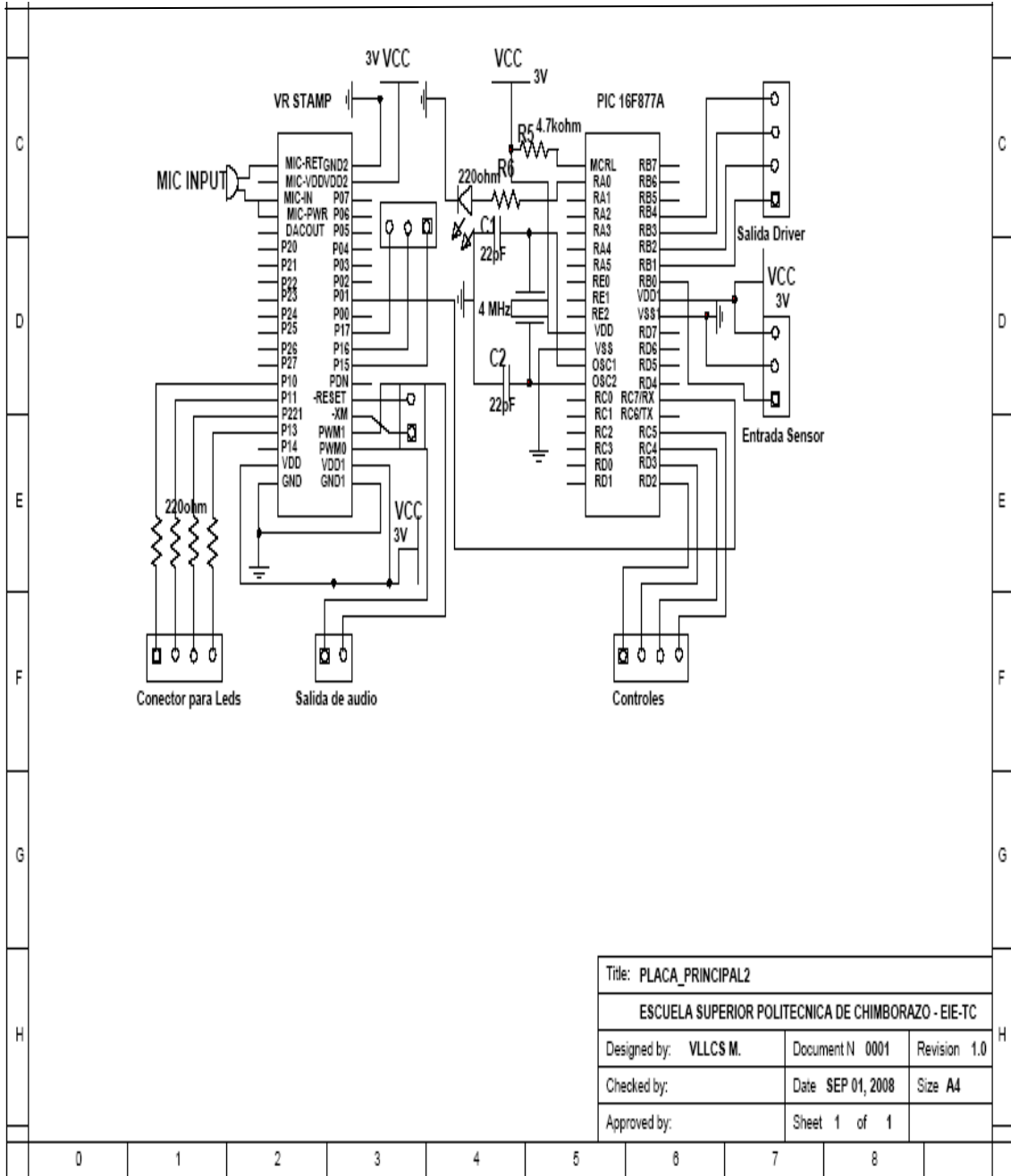
Robot: Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas.

Serie: Método de procesamiento o transmisión de datos basado en etapas sucesivas, no simultáneas.

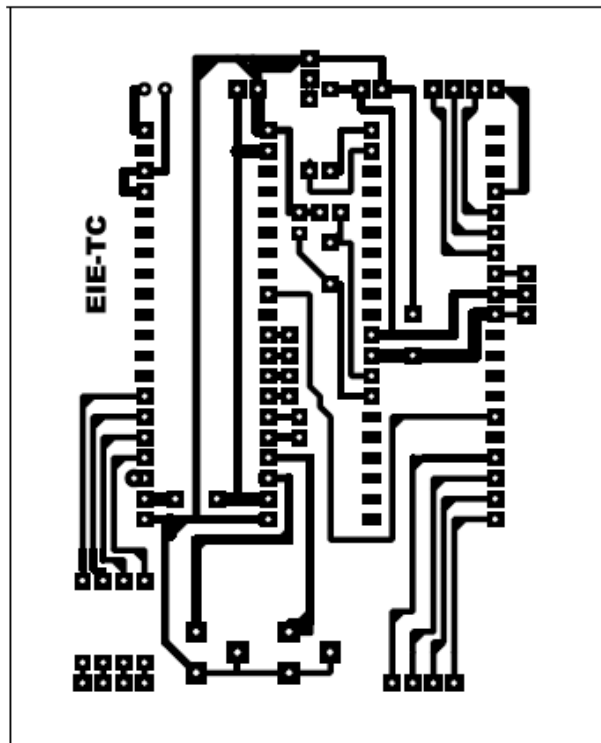
ANEXOS

ANEXO I

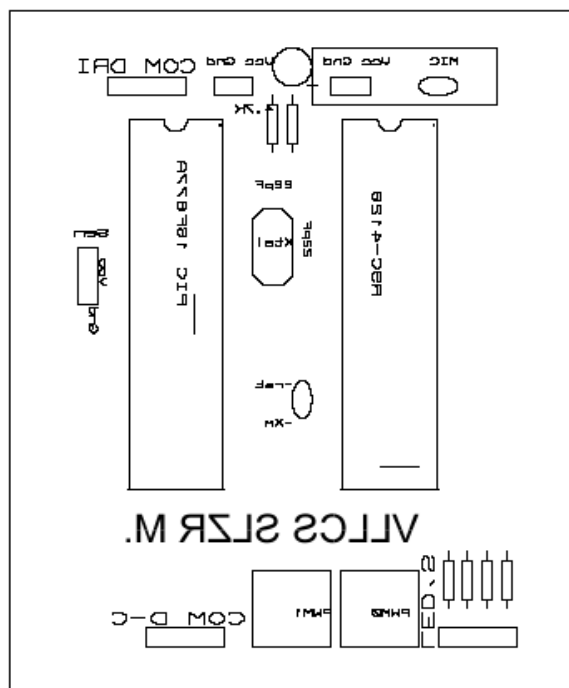
DIAGRAMA PLAQUETA PRINCIPAL



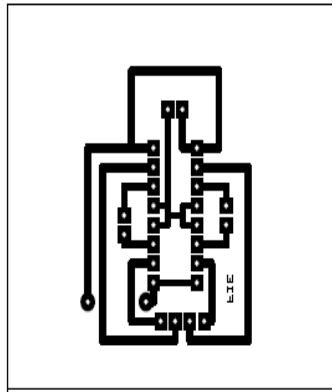
Title: PLACA_PRINCIPAL2		
ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO - EIE-TC		
Designed by: VLLCS M.	Document N 0001	Revision 1.0
Checked by:	Date SEP 01, 2008	Size A4
Approved by:	Sheet 1 of 1	



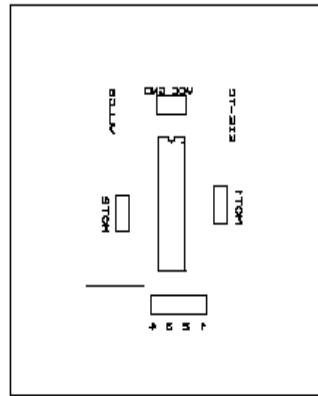
Circuito Impresso



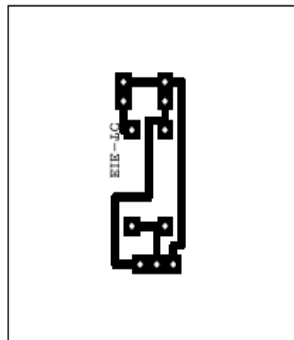
Componentes



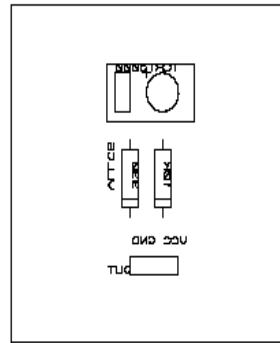
Circuito Impreso Driver



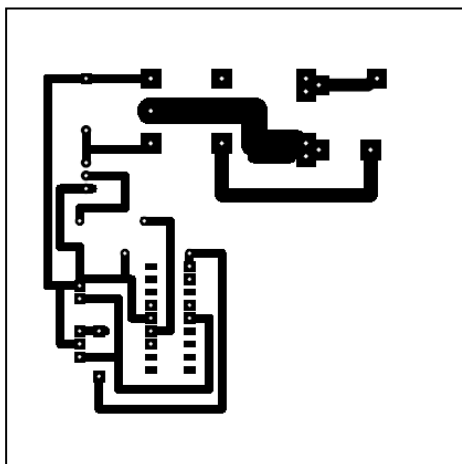
Componentes Driver



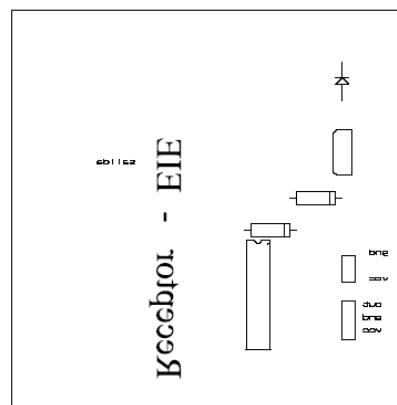
Circuito impreso Sensor



Componentes Sensor



Circuito Impreso Receptor



Componentes receptor

ANEXO II

CODIGO FUENTE PIC 16F877A Y PIC 16F628A

```
'* Name   : CEREBRO.BAS           PIC 16F877A           *
'* Author : VLLCS SLZR           *
'* Notice : Copyleth para quien le interese aprender se autoriza la copia*
'* Date   : 30/07/2008           *
'* Version : 1.0                 *
'* Notes  : Programa que maneja genera subrutinas para el control de motores,
'* ingreso de señal de los sensors.           *
'*       :                       *
```

```
include "modedefs.bas"
```

```
ADCON1=7 ;Digitalizo Puertos
```

```
DATOS VAR BYTE; declaro variable datos para recibir información VR Stamp.
```

```
x var byte
```

```
@ device XT_osc
```

```
@ device BOD_OFF
```

```
inicio:
```

```
    high porta.0
```

```
    serIN portC.7,T9600,DATOS;recibo por el puerto C.7 informacion a 9600 baudios ;desde el
```

```
;módulo de reconocimiento de voz
```

```
    IF DATOS =5 THEN  izq ;izquierda
```

```
    IF DATOS =6 THEN  der ;derecha
```

```
    IF DATOS =3 THEN  ade ;ADELANTE
```



```

IF DATOS =4 THEN atr ;ATRAS

IF DATOS =11 THEN bal ;BAILAR

IF DATOS =20 THEN reset ;alto como robotrax

IF DATOS =15 THEN reset ;alto

IF DATOS =13 THEN rad ;radio

IF DATOS =16 THEN rad ;apagar funciona

goto inicio

izq: ;sub rutina izquierda

    gosub reset

    high portb.1 ; combinacion 1 motor 1

    low portb.2 ; combinacion 0 motor 1

    high portb.3 ; combinacion 1 motor 2 dirección izquierda

    low portb.4 combinacion 0 motor 2 dirección izquierda

    pause 2000 ;espero 2 segundos y desactivo

    low portb.3

    low portb.4

    for x =1 to 40 ; leyendo interrupción.

        if portb.0=1 then atr ;envoi hacia atras

        pause 100 ;pausa con 100 milisegundos

    next

    gosub reset

goto inicio

der: ;sub rutina derecha ; descripción análoga subrutina izquierda

    gosub reset

    high portb.1

```

```
low portb.2
high portb.4
low portb.3
pause 2000
low portb.3
low portb.4
for x =1 to 40
  if portb.0=1 then atr
    pause 100
  next
  gosub reset
goto inicio
```

```
ade: ;sub rutina adelante
gosub reset ;reseteo puertos
high portb.1;polaridad antihorario adelante
low portb.2
for x =1 to 50 ,espero por señal de sensor
  pause 100
  if portb.0=1 then atr
next
gosub reset
goto inicio
```

atr: ;sub rutina atrás

 gosub reset

 high portb.2 ; polaridad horario motor1

 low portb.1

 ; pause 4000

 for x =1 to 50

 pause 100

 if portd.3=1 then ADE

 next

 gosub reset

goto inicio

bal: ;

 gosub reset ;genero secuencia de movimiento horario por un tiempo de 3,5 ;segundos

 for x =1 to 5

 high portc.5 ; mueve las manos del robot

 high portb.1 ;motor 1 antihorario

 low portb.2

 high portb.4

 low portb.3

 pause 350

 high portb.3 ;motor 2 izquierda derecha

 low portb.4

```
high portb.2

low portb.1

pause 350

;pause 100

if portb.0=1 then reset

next

low portc.5

gosub reset

goto inicio

;*****

rad: ; Envía un pulso para activar módulo inalámbrico

low portd.2

pause 1000

high portd.2

goto inicio

reset: ;resetea los puertos

low portb.1

low portb.2

low portb.4

low portb.3

low portc.5

return

end
```

Recepción módulo inalámbrico con el pic 16f628a

'* Name : Recepcion.BAS *

'* Author : VLLCS SLZR *

'* Date : 31/07/2008 *

'* Version : 1.0 *

'* Notes : *

'* : *

cmcon=7 ; digitalizo puerto

num var byte

; BAN var byte

num=0

; BAN=0

HIGH PORTB.1

inicio:

if porta.1=1 then ir; existe un pulso

goto inicio

ir:

IF porta.1=1 then ir ;espero que termine el pulso

if num=0 then prende ;son banderas para saber si esta prendido o apagado

if num=1 then apaga

goto inicio

prende: high portb.0

num=1

goto inicio

apaga:

low portb.0

num=0

goto inicio

end

ANEXO III

MANUAL DE USUARIO

1. Precauciones

La alimentación de la plaqueta principal debe ser de 3V

La alimentación del driver L293D debe ser de 5V

Evitar que el robot tenga caídas fuertes

No toque con las manos los componentes electrónicos del robot

2. Introducción

El robot llamado Robotrax fue diseñado para un prototipo de robot compañero, controlado por la voz que interactúa con el usuario a través de mensajes de voz emitidos por este cuando se pronuncia un comando y la orden es recibida.

Para la ejecución del robot el usuario debe conocer las órdenes:

Palabra de inicio: Robotrax

La palabra de inicio debe ser antepuesta antes de realizar una orden.

Comandos de Movilidad: Adelante, atrás, izquierda, derecha, bailar

Comandos de control domótico: Encender, apagar, radio, televisión.

Comandos de Dialogo: Saludo, bienvenida, cantar equivocaste, alto

Ejemplo de ejecución de comando:

Al escuchar 'Robotrax esta listo para empezar' se puede pronunciar la palabra 'Robotrax' fuerte y claro si es reconocido el comando se escuchara 'de una orden' y escogemos cualquier comando 'adelante' y se escuchara 'orden recibida adelante', termina la confirmación del comando y el robot empezara a moverse.

3. Características técnicas

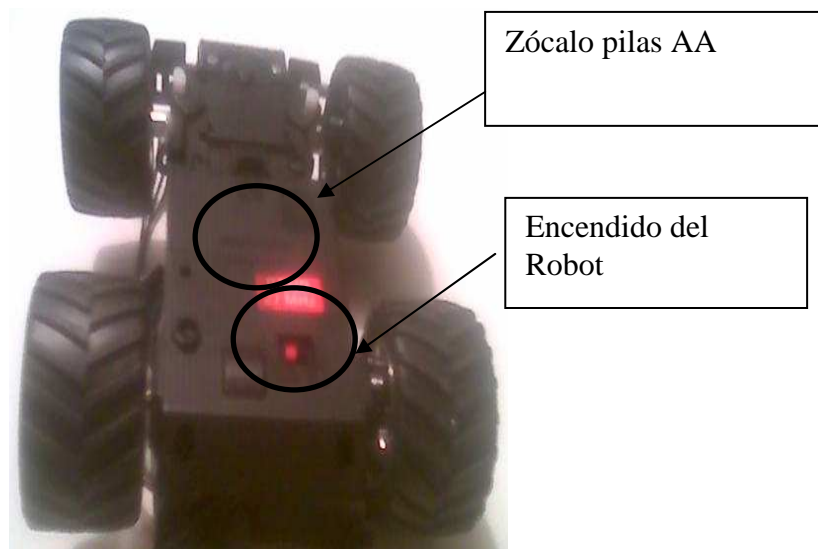
3.1 Alimentación eléctrica del Robot

3VDC, pila tipo C zócalo 1

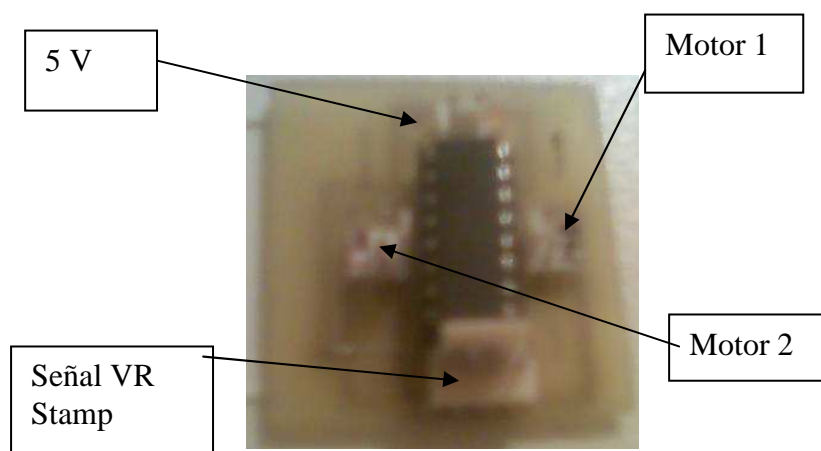
4 pilas AA para circuito de alimentación del Driver

3.2 Encendido y Driver del Robot

El robot cuenta con un contacto en la parte inferior del chasis el cual deja listo para funcionar el driver en la figura se muestra su ubicación.



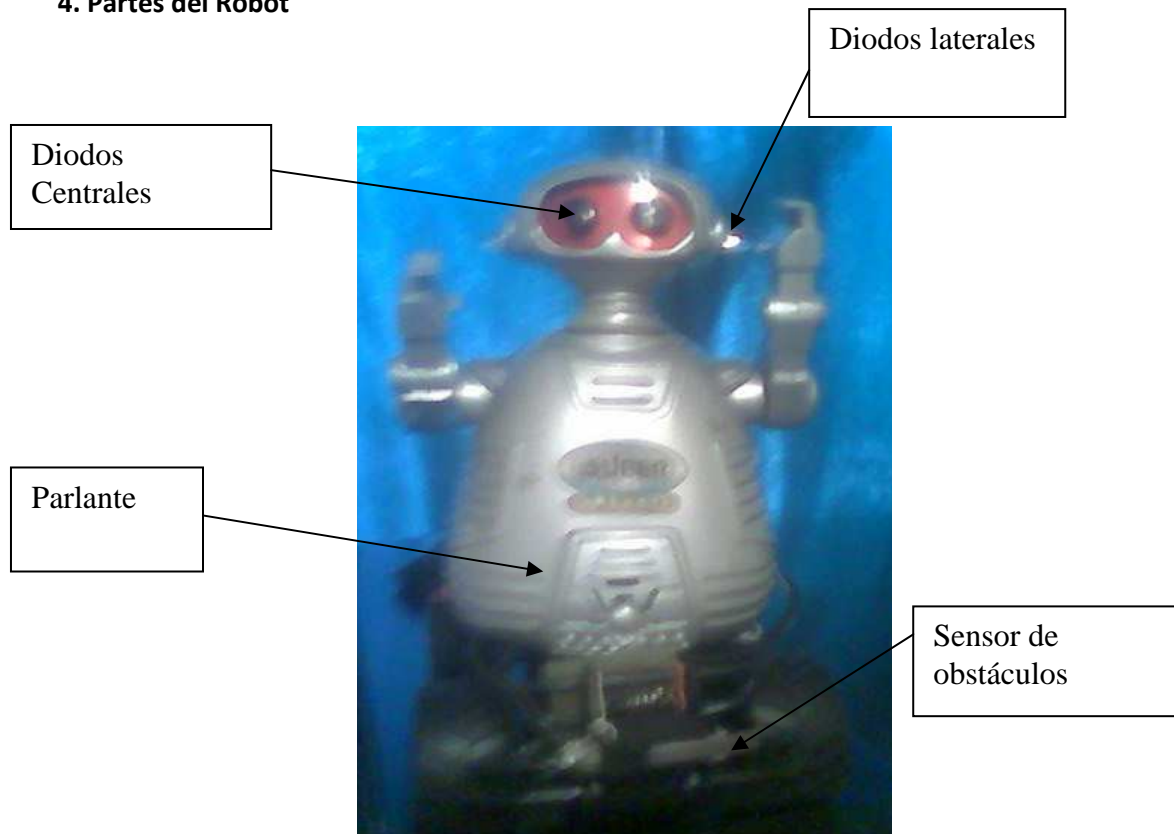
Chasis del robot



Driver del Robot

El driver del robot tiene cuatro zócalos, los cuales son detallados en la figura del driver del robot, cada uno de estos deben ser colocados de acuerdo a la etiqueta de cada una de las partes del robot.

4. Partes del Robot



Los diodos son indicadores al igual que los mensajes de voz emitidos por el robot, por ejemplo los diodos centrales encendidos significa que el robot está ejecutando mensajes de voz, los diodos laterales encendidos significa que está esperando la orden de inicio para escuchar comandos.

BIBLIOGRAFÍA

- 1.- AMESCUA, A.** Análisis y Diseño Estructurado y Orientado a Objetos de Sistemas Informáticos. México DF – México: McGraw-Hill, 2003. pp. 1-643
- 2.- ANGULO, J.** Microcontroladores PIC: Diseño práctico de aplicaciones. 3a ed. Madrid – España: McGraw-Hill, 2003. pp. 1-212
- 3.- ALVAREZ, M.** Diseño y Construcción de un Robot Hexápodo Inteligente Comandado por Voz. Tesis Ing. En Electrónica y Computación. Riobamba. Escuela Superior Politécnica de Chimborazo. Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica, 2006. pp. 62-80
- 4.- BOYLESTAD, L.** Electrónica Teoría de Circuitos. 6a. ed. México DF – México: Prentice Hall, 1997. pp. 560-577
- 5.- CORRALES, S.** Electrónica Práctica con Microcontroladores PIC. Quito- Ecuador. Imprenta Gráfica, 2006. 178 p.
- 6.- LAWRENCE, R. BIING-HWANG J.** Fundamentals of Speech Recognition. Washington - USA. Prentice Hall, 1993. pp. 20-63.
- 7.- REYES, C.** Aprenda a programar Microcontroladores. Quito – Ecuador: Gráficas Ayerve, 2004. pp. 1-201.

BIBLIOGRAFÍA DE INTERNET

8.- MANUAL DE PIC BASIC PRO

http://www.todopic.com.ar/pbp_sp.html

(2008-02-20)

9.- MECANISMOS

<http://es.wikipedia.org/wiki/Engranajes>

(2008-02-12)

<http://tigoe.net/pcomp/motors.shtml>

(2008-01-15)

<http://www.emc.uji.es/d/IngMecDoc/Mecanismos/index.html>

(2008-01-16)

<http://www.x-robotics.com/motorizacion.htm>

(2008-01-16)

10.- MICROCONTROLADORES

<http://www.covingtoninnovations.com/noppp/noppp-sp.html>

(2008-02-25)

<http://www.x-robotics.com/rutinas.htm>

(2008-02-25)

<http://www.jvmbots.com/viewtopic.php?t=19&start=45>

(2008-02-25)

11.- RECONOCIMIENTO DE VOZ

www.conscious-robots.com/raul/voz/rec_voz.htm

(2008-01-25)

http://www.conscious-robots.com/raul/voz/rec_voz.htm

(2007-12-29)

<http://www.eie.fceia.unr.edu.ar/~acustica/biblio/fonatori.pdf>

(2007-11-28)

<http://proton.ucting.udg.mx/dpto/tesis/quetzal/CAPITUL2.html>

(2007-10-18)

http://www.sensoryinc.com/products/RSC-4x_series.html

(2008-07-18)

http://www.sensoryinc.com/products/RSC-4x_toolkits.html

(2008-07-18)

http://www.sensoryinc.com/products/vr_stamp.html

(2008-07-18)

http://www.sensoryinc.com/products/vr_stamp_toolkits.html

(2008-07-18)

<http://www.sensoryinc.com/support/docs/80-0272-l.pdf>

(2008-07-18)

12.- SENSORES

<http://www.x-robotics.com/sensores.htm>

(2008-01-16)

<http://www.robotroom.com/Tach2.html>

(2008-01-17)