VNIVERSIDAD NACIONAL AVFNOMA DE MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

Diseño y construcción de un sintetizador de voz basado en multidominio

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA (COMPUTACIÓN)

P R E S E N T A:

Noé de Jesús Romero Serrano

DIRECTOR DE TESIS: Dr. José Abel Herrera Camacho.

México, D.F. **2009.**

Índice

Introducción

La siguiente tesis tiene por objeto desarrollar en hardware un sintetizador de voz de vocabulario ilimitado de palabras, para el idioma español hablado en México. Existen restricciones de normalización y una función multimodal, capaz de identificar ciertos tópicos del texto y dar una salida adecuada dependiendo del contexto. Una de las ventajas de este sistema es la modificación sencilla de la base de datos, pudiendo entonces sustituir fácilmente la voz sintetizada.

El sintetizador fue desarrollado en un sistema electrónico mínimo. Para este propósito, se diseño una tarjeta electrónica que es de fácil traslado y volumen pequeño, aunado a un costo relativamente bajo.

A continuación se explica el contenido de cada uno de los capítulos.

En el capítulo 1 se presentan los conceptos básicos de un sistema de síntesis de voz y se explican los módulos que lo componen, se realiza una revisión histórica de los sintetizadores de voz y, por último, se explican los tipos de síntesis de voz.

En el capítulo 2 se abordan los componentes físicos de un sintetizador de voz, las características de cada componente y la selección del dispositivo elegido para la construcción del sintetizador. En cada punto se presentan las especificaciones técnicas, su funcionamiento y los resultados esperados. Como punto final, se puede observar su diagrama lógico.

El capítulo 3 se dedica al enfoque multidominio de la síntesis de voz, iniciando con este propio concepto. Se abordan los trabajos relacionados con este enfoque y como se compara con los ya existentes. Se explica como este concepto ha sido implementado en el sintetizador aquí diseñado.

En el capítulo 4 se fundamentan las unidades básicas usadas en un sintetizador con la lógica de concatenación. Se presenta el procesamiento de señales requerido para la síntesis.

En el capítulo 5 se revisan algunos de los algoritmos de síntesis de voz más usados, haciendo una revisión especial en SOLA (*Synchronous Overlap Add*) y sus variantes más representativas.

En el capitulo 6 se explica como el sintetizador diseñado cumple con las expectativas, además de examinar como realiza cada una de las operaciones de síntesis.

No se presentan diseños nuevos para la etapa de simulación. Sin embargo, el diseño de la tarjeta electrónica como un sistema mínimo, constituye una aportación original, que es comparable a diseños extranjeros.

1. Aspectos generales de la síntesis de voz

1.1. Elementos de un sintetizador de voz

Un sistema de síntesis de voz es aquel que genera voz por medios electrónicos o software intentado simular la voz humana. Estos sistemas son también conocidos como sistemas de texto a voz (**TTS**, siglas de las palabras en inglés **T**ext–**T**o-**S**peech). La idea general en este tipo de dispositivos es tener una entrada de texto libre, la cual será procesada para la obtención de fonemas que es la expresión básica del lenguaje, y con esto obtener una respuesta que en este caso sería la voz sintetizada. Se pueden identificar claramente dos aspectos fundamentales en la síntesis de voz: la entrada del texto y la relación entre el texto procesado y la voz.

En primera instancia, el análisis del texto de entrada presenta dos grandes retos:

- La normalización del texto que consiste en tomar todos aquellos símbolos (números, abreviaciones, etcétera), que no indican explícitamente la forma en que deben ser pronunciados y que deben ser expandidos a palabras. Es decir, si encontramos una expresión como la siguiente: “hoy 2/01/2009 el Ing. x” en un correcto análisis del texto debería leerse así “ hoy dos de enero de dos mil nueve el ingeniero equis ”. La normalización no es un aspecto trivial dado que en muchos idiomas varios símbolos o abreviaturas contienen un significado especial. Un ejemplo es: “El Ing. x mide una tabla de 2 x 2 m”, un sintetizador podría leerlo de la siguiente forma “el ingeniero equis mide una tabla de dos equis dos eme” con lo cual se pierde por completo la idea que se quería transmitir.

- El análisis lingüístico obtiene la entonación y la variación de la frecuencia fundamental en el tiempo. Esto da como resultado la duración, entonación, ritmo y amplitud de las palabras a ser reproducidas.

La estructura prosódica depende normalmente del contexto y la intención, pero debido a la dificultad o imposibilidad de obtenerla a partir de un texto, se utiliza una entonación neutral que se obtiene de análisis estadísticos. En años recientes este rubro ha tomado una fuerte importancia.

La pronunciación correcta se puede obtener por dos métodos básicos: [2]

- Reglas de pronunciación, que almacenan la relación entre símbolos y sonidos. Se debe hacer notar, sin embargo, que la relación exacta es aún un problema abierto y diferentes sistemas ocupan tablas diferentes. Este método será el utilizado en el desarrollo del sistema y su implementación.

- Tablas de pronunciación, que constan de una lista de palabras y su pronunciación. Éstas se utilizan en español normalmente sólo para excepciones, ya que en la mayoría de los casos la pronunciación está basada en las reglas. También son comunes en otros idiomas (como el inglés) donde la relación entre grafías y fonemas no es tan directa.

La señal puede ser generada por varios métodos. El método más sencillo y el más utilizado actualmente es el concatenativo, el cual consiste en unir segmentos de sonido previamente grabados que son unidos para generar la salida.

1.2. Historia de la síntesis de voz

Mucho antes del desarrollo del procesado de señal moderno, los investigadores de la voz intentaron crear máquinas que produjesen habla humana. El Papa Silvestre II (1003), Alberto Magno (1198-1280) y Roger Bacon (1214-1294) crearon ejemplos tempranos de "cabezas parlantes".

En 1779, el científico danés Christian Gottlieb Kratzenstein construyó modelos de tracto vocal que podían producir las cinco vocales largas (a, e, i, o, u) [16, 17].

En la segunda mitad del siglo XVIII Wolfgang von Kempelen de Viena, en una de sus obras, describe cómo construir una máquina, capaz de emitir sonidos similares a los del habla humana. En el siglo XIX se extendió el uso de máquinas similares a la de Kempelen, que no aportaron nada nuevo, hasta que llegó el británico Joseph Faber y se decidió a recrear fielmente el sonido del habla [18]. *Euphonia* se trataba de un aparato de considerable tamaño, manejado a través de teclados y pedales. Durante muchos años se intentó simular las vías respiratorias humanas para mejorar este tipo de aparatos, como en el caso del artefacto perfeccionado por el estadounidense R.R. Riesz en 1937 [48].

El vocoder es un analizador y sintetizador de voz, que fue desarrollado en la década de 1930 como un codificador de voz para telecomunicaciones [24]. Su primer uso fue la seguridad en radiocomunicaciones, donde la voz tiene que ser digitalizada, cifrada y transmitida por un canal de ancho de banda estrecho. El vocoder es útil para la encriptación de voz que garantiza que sólo otra persona con el decodificador pueda escuchar lo que se ha dicho. Por lo tanto, un vocoder es básicamente un dispositivo de encriptación.

El vocoder examina el habla encontrando su onda básica, que es la frecuencia fundamental, y midiendo cómo cambian las características espectrales con el tiempo grabando el habla. Esto da como resultado una serie de números representando esas frecuencias modificadas en un tiempo particular a medida que el usuario habla. Al hacer esto, el vocoder reduce en gran medida la cantidad de información necesaria para almacenar el habla. Para recrear el habla, el vocoder simplemente revierte el proceso, creando la frecuencia fundamental en un oscilador electrónico y pasando su resultado por una serie de filtros basado en la secuencia original de símbolos.

Muchos vocoders usan un gran número de canales, cada uno en una frecuencia. Los diversos valores de esos filtros no son almacenados como números, que están basados en la frecuencia original, sino por una serie de modificaciones que el fundamental necesita para ser modificado en la señal vista en el filtro. Durante la reproducción esos números son enviados de vuelta a los filtros y entonces se modifican con el conocimiento de que el habla varía típicamente entre esas frecuencias. El resultado es habla inteligible, aunque algo mecánica. Los vocoders a menudo incluyen también un sistema para generar sonidos sordos, usando un segundo sistema consistente en un generador de ruido en lugar de la frecuencia fundamental.

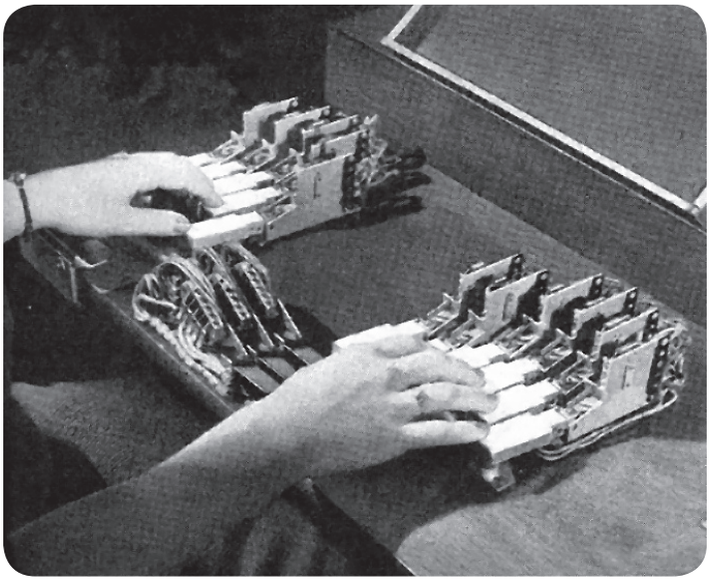

Fig. 1.1. VODER [24]

A partir de este sistema se construyó una segunda versión que fue mostrada en la Feria Mundial de Nueva York (1939). Este sistema, llamado VODER (mostrado en la figura 1.1.) y desarrollado por Homer Dudley, generaba una salida de ruido o una salida de audio senoidal de acuerdo con un selector. La frecuencia de esta salida podía ser controlada por un pedal. Esta salida era después filtrada por 10 filtros paso-banda cuya amplitud se modificaba por medio de los dedos [2,16 y 18].

Lamentablemente, como suele suceder en estos casos, lo que es simple en la teoría en la práctica resulta extremadamente complicado. Para conseguir que la máquina hablara se necesitaba un operador que manipulara el juego de llaves y el pedal para poder convertir el

ruido y los tonos, en vocales, consonantes, pausas e inflexiones. Aunado a ello, el operador necesitaba un año de práctica para poder dominar las teclas.

En 1951 se presentó el reproductor de patrones desarrollado en los laboratorios Haskins de la Universidad de Yale. Éste utilizaba espectrogramas, los cuales se iluminaban y eran enviados a un conjunto de celdas fotovoltaicas, cada una de las cuales controlaba la intensidad de una onda fundamental de diferentes frecuencias en saltos de 120 Hz, que podían reconstruir aproximadamente la señal del espectrograma. Franklin Cooper, Alvin Liberman, Pierre Delattre y otros asistentes, experimentaron con espectrogramas reales y con adaptaciones dibujadas a mano para evaluar la importancia de diferentes factores. Este sistema generó una inteligibilidad mucho más alta (más de 90% con espectrogramas reales y más de 80% con los espectrogramas dibujados a mano) [15].

Estos dos sistemas funcionaban copiando los patrones espectrales de la voz. Poco después del desarrollo de estos sistemas, se dio un nuevo enfoque a la teoría de síntesis de voz. Este nuevo enfoque fue la generación de una teoría acústica de la forma en que se produce la voz y no sólo en los resultados del proceso. Esta teoría es la base de la síntesis por formantes. Según esta teoría, la voz se puede considerar como la salida de un filtro lineal excitado por una o más fuentes, principalmente las cuerdas vocales, y ruido turbulento debido a diferencia de presiones a través de un estrangulamiento [15]. El filtro en este caso, es una simulación de los efectos del tracto vocal.

En 1953 se crearon los primeros sintetizadores de formantes como el Paramteric Artificial Talker (PAT) construido por Walter Lawrence y el OVE I construido por Gunnar Fant.

El PAT tenía tres resonadores en paralelo. Se tenía una señal de entrada de ruido o periódica y de ahí, a través de patrones dibujados sobre un vidrio que se deslizaba, se controlaban 3 frecuencias del formante, así como las amplitudes del ruido, del fraseo y de la fundamental.

El OVE utilizaba filtros en cascada en lugar de en paralelo. Los dos más bajos eran controlados por movimientos en 2 dimensiones de un brazo mecánico, mientras que la amplitud y la fundamental eran controladas por potenciómetros. Sin embargo, este sistema sólo podía generar vocales.

Es interesante notar que, aunque ambos sistemas parten de la misma teoría y usaban los mismos principios, utilizaron diferentes métodos para llevarla a la práctica, y hasta la fecha aún existe controversia sobre cuál de los dos métodos es mejor, o si la mejor opción es utilizar una mezcla de ambos; teoría propuesta en 1972 por Klatt [15].

Una de las modificaciones más grandes a esta clase de sistemas fue la introducción de sistemas híbridos (cascada y en paralelo). En la propuesta de Klatt [15] cada sistema se utilizaba para modelar diferentes tipos de sonidos (en paralelo para sonidos sonoros y en cascada para sonidos sordos). Este sistema propuesto por Klatt fue además presentado como un listado en Fortran en 1980, lo que permitió su uso más extendido.

Otro punto importante en la historia de los sintetizadores por formantes ocurrió en una conferencia en Boston, en 1972, cuando John Holmes presentó una salida de voz prácticamente indistinguible de una voz natural [15]. Desafortunadamente, esta señal fue generada de forma manual y basada en un proceso de prueba y error de varios meses de duración. Aunque este experimento demostró varios factores importantes en la generación de oraciones, su método no ha podido ser automatizado.

El siguiente avance importante en este tipo de sintetizadores consistió en cambiar la señal de entrada, de una señal monótona (triangular, tren de pulsos) en una señal que se asemejara más a la señal que entra al tracto vocal.

El primer avance de este tipo se dio en 1975 (Rothenberg), cuando se utilizó un sistema de tres parámetros de acuerdo con la apertura de la glotis, la amplitud de la respiración y la frecuencia fundamental. Se han creado métodos que simulan más parámetros, pero hasta la fecha el resultado aún no es completamente natural, debido posiblemente a la falta de conocimiento del modelo real [15].

Aparte de este tipo de sintetizadores, otra línea paralela de investigación es la generación de líneas de transmisión que simulen un tubo similar al tracto vocal. Sin embargo, debido a restricciones en el conocimiento del tracto vocal y en la cantidad de cálculos, se ha avanzado poco en esta área, aunque existen algunos modelos de sintetizadores de este tipo.

Una vez que se obtuvieron sistemas que pudieran simular la voz humana, una aplicación muy importante, que sólo se hizo posible con el advenimiento de computadoras y circuitos integrados, es la generación de una señal de voz a partir de una entrada fonémica o de texto.

El primer programa de este tipo se desarrolló en 1961 (Kelly y Gerstman) con un sintetizador en cascada de tres formantes, cuyos parámetros posteriormente se modificaban a mano [15].

En 1964 apareció otro sistema (Holmes) que funcionaba a partir de síntesis por formantes en paralelo y un conjunto de tablas que permitía generar resultados más complejos como coarticulación y el uso de alófonos. En 1966 (Mattingly) [25] modificó el programa para generar transiciones más realistas, pero obtuvo poca mejoría. El primer uso práctico que se le intentó dar a este sistema, fue una adaptación como parte de una máquina de lectura para ciegos, pero nunca se concretó por falta de recursos.

A finales de los 60's y principio de los 70's se continuó la investigación de los sistemas de síntesis por regla, ajustando diferentes parámetros para hacerlos más similares a la voz natural, principalmente por Klatt, [15] dando como resultado el sistema de síntesis del M.I.T. Este sistema conocido como MITalk (1976) fue vendido y cambió de manos varias veces durante los siguientes años. Después de este sistema se desarrolló el Klattalk que continuó siendo mejorado hasta finales de los 80's.

Otro importante desarrollo fue la introducción de sistemas dentro de un circuito integrado. Este es el caso del Votrax SC-01 (1976) para síntesis por formantes y el TMS-5520 de Texas para síntesis por concatenación.

Además, en ese año se introdujo el LPC (Linear Prediction Code) (Mackhoul, e independientemente Markel y Grey) que es la base de la mayoría de los sistemas actuales de síntesis, en las cuales sólo se varía el tipo de entrada y el tipo de filtro donde se alimentan los parámetros .

Otra línea de investigación consiste en tomar segmentos de voz pregrabados como bloques para construir una frase cualquiera. Debido a las características de la voz, no se pueden

usar palabras o sílabas (debido a su gran cantidad) ni fonemas (debido a que no toman en consideración los efectos de coarticulación ni de transición entre fonemas). Debido a estos problemas, en 1958, Peterson [15] propuso una unidad denominada difonema, que corresponde al segmento entre el centro de un fonema y hasta el centro del siguiente, lo que permite tomar en cuenta los efectos de transición y coarticulación. En teoría, se requiere el cuadrado del número de fonemas de una lengua, pero hay combinaciones que no pueden existir, con lo que se reduce el número. Se pueden agregar algunos di-fonemas para grabar diferencias entre sílabas acentuadas y no, alófonos, etcétera. Peterson estimó que se requieren unos 8 000 difonemas en inglés, aunque normalmente se utilizan aproximadamente 1 000.

En 1961, Sivertsen [15] propuso mezclar difonemas y unidades más largas llamadas diádas, que contienen la mitad final de un fonema, un fonema completo y la mitad inicial del siguiente para conservar algunos fenómenos que pueden no estar previstos en los difonemas.

Aunque tienen problemas como discontinuidades en algunas uniones, estos sistemas son muy utilizados debido a su relativa sencillez y alta inteligibilidad. El primer sistema de este tipo fue mostrado en 1967, pero el desarrollo de este sistema se canceló por falta de recursos [15].

En 1976, Olive y Spickenagle intentaron extraer las características de los fonemas para crear un sistema que generara un catálogo de difonemas de forma automatizada [15].

En 1980, Texas Instruments introdujo un chip (TMS-5100) con predicción lineal usado para síntesis. En 1982, Street Electronics realizó un sintetizador por difonemas de bajo costo basado en una versión más reciente del chip de Texas Instruments (TMS-5220) [15].

1.3. Tipos de síntesis de voz

Actualmente la mayoría de los sistemas de síntesis de voz están basados principalmente en 3 tipos básicos de síntesis: [18]

- Articulatorios: tratan de modelar directamente el sistema generador de voz.
- Por formantes: modelan la función de transferencia o de polos de frecuencias del tracto vocal.
- Por concatenación: utilizan segmentos pregrabados que son unidos (concatenados).

Los dos tipos más usados son por formantes y por concatenación. Aunque el primero fue más usado inicialmente debido, a limitaciones en la capacidad de almacenamiento, actualmente el segundo es más usado debido a que son posibles mayores capacidades de almacenamiento.

1.3.1. Síntesis articulatoria

Este tipo de síntesis trata de modelar los órganos vocales lo más perfectamente posible, por lo que teóricamente es el sistema que podría generar síntesis de más alta calidad, pero a la vez es el sistema más complicado y de más alta carga computacional. Este tipo de síntesis involucra normalmente modelos de las cuerdas vocales humanas, de la lengua (posición, altura, entre otros), apertura del velo, presión de los pulmones, apertura glotal.

El modelo de articulación se genera normalmente a partir de radiografías, pero éstas no proporcionan información suficiente para conocer todos los parámetros necesarios. La ventaja de éste es que puede utilizar efectos que difícilmente se podrían incluir en otros sistemas.

Debido a las complejidades de análisis y la carga computacional requerida, este tipo de síntesis ha recibido poca atención, por lo que ha tenido muy poco desarrollo.

1.3.2. Síntesis por formantes

Este es uno de los métodos de síntesis más usados. Se basa en un modelo de entrada-filtro-salida del cual existen básicamente 2 tipos (filtros en cascada y en paralelo), y combinaciones de ambos, lo cual produce un mejor resultado. Además, este sistema proporciona mayor flexibilidad que la síntesis por concatenación y una menor dificultad que la síntesis articulatoria.

Este sistema fue muy utilizado en un principio, debido a que tiene pocos requerimientos de memoria y almacenaje. Sin embargo, este método ha sido usado en mucha menor cantidad últimamente y reemplazado en gran medida por los sistemas concatenados, ya que estos tienen una mejor salida de audio. Además, esta se genera de forma más sencilla y la capacidad de almacenamiento que requieren ya no es una limitante.

Normalmente se usan al menos tres formantes para producir la señal de voz, aunque a veces se emplean hasta cinco para mejorar la calidad. Cada formante se modela por medio de un resonador basado en un filtro centrado en la frecuencia del formante, y modelado con un par de polos, con lo que se puede indicar el ancho de banda del filtro.

La síntesis por formantes utiliza cierto conjunto de reglas que determinan los parámetros necesarios para cada sonido. Algunos de estos parámetros pueden ser: frecuencia fundamental (F0), grado de excitación (VO), frecuencias y amplitudes de los formantes (F1, F2, F3, A1, A2, A3).

Los resonadores en cascada funcionan mejor con los sonidos no nasales, pero tienen problemas con las fricativas y oclusivas. Los resonadores en paralelo tienen problemas con las vocales pero funcionan bien para nasales, fricativas y oclusivas. Debido a esto Klatt [15] (1980) ideó un modelo con una mezcla de ambos filtros y 6 formantes, la adición de un ruido de alta frecuencia y un sistema de excitación compleja. Este modelo es el más utilizado en los sistemas comerciales actuales como el MITalk, DECtalk y Prose-2000.

Otro sistema mixto es el PARCAS [18], introducido por Laine en 1982, que es modelado por pares de ecuaciones de transferencia parciales y un conjunto de constantes para mantener

las amplitudes balanceadas en diferentes salidas. Se modela por medio de filtros, tanto en cascada como en paralelo, y se usan entradas de pulsos y de ruido, según el fonema a sintetizar.

Una vez que se tiene el conjunto de resonadores a utilizar, éstos se alimentan con un tren de pulsos con una frecuencia igual a la de la fundamental (Fo) para sonidos con fundamental (por ejemplo vocales) o con una fuente de ruido para sonidos sin fundamental (por ejemplo /s/).

En algunos sistemas modernos se introducen señales diferentes a un tren de pulsos para mejorar la salida. Una versión alternativa de este sistema de síntesis es LPC (Predicción lineal). En éste, los coeficientes de los filtros son estimados previamente de segmentos de voz pregrabados (por lo que también tienen relación con la síntesis concatenativa).

La idea básica de LPC es que una muestra de sonido y(k) puede ser aproximada por medio de una combinación lineal de cierto número de muestras pasadas y coeficientes a(k) precalculados. Esta predicción causa un pequeño error e(n) conocido como señal residual [3].

$$y\,(n) = e\,(n) + \sum_{k=1}^{p} a\,(k)\; y\;(n - k)$$

$$e\;(n) = y\,(n) - \sum_{k=1}^{p} a\,(k)\; y\;(n - k) = y\;(n) - \tilde{y}\;(n)$$

Para calcular los coeficientes, se minimiza el error medio cuadrático.

Una vez calculados estos coeficientes se utiliza el sistema básico de un filtro con entrada de pulsos o ruido, según el tipo de fonema (sonoro o sordo). Esta señal es filtrada por un filtro al que se le alimentan los coeficientes a(k). El número de k de coeficientes (orden del filtro) es normalmente 10. Estos coeficientes deben ser modificados para cada ventana (aproximadamente entre cada 5 y 10 ms.).

La calidad de este sistema es baja debido a que las oclusivas tienen cambios muy súbitos, difícilmente modelables con un filtro. Para aumentar la calidad de los sistemas LPC se utilizan métodos alternativos, por ejemplo, filtros que aumentan la calidad de ciertas frecuencias a costa de otras, de una forma similar a la curva de respuesta del oído. Además se puede modificar la entrada, para ser alimentado por trenes de pulso de diferentes frecuencias simultáneamente, agregando la señal residual para eliminar el error o por medio de un "libro de códigos", donde se almacenan diferentes tipos de excitaciones según el fonema a generar [18].

Un método alternativo, conocido como método senoidal, consiste en obtener las características espectrales de la señal y su variación en el tiempo por medio de transformadas de Fourier. A partir de esta transformada se obtienen los picos de magnitud frecuencial (w_i) y la fase (ϕ_i) de la señal en estas frecuencias. A partir de estos parámetros se puede reconstruir la señal por medio de una sumatoria de cosenos [18].

$$s(n) = \sum_{i=i}^{L} A_i \cos(w_i n + \phi_i)$$

Debido a sus características este sistema genera problemas en fonemas que no tienen formantes al no existir picos de frecuencias. Este sistema se usa principalmente en sistemas que tratan de sintetizar canto, donde se requiere un control más estricto de las frecuencias y no importa tanto su inteligibilidad.

1.3.3. Síntesis por concatenación

Este tipo de sistema se basa en conectar segmentos de voz previamente grabados y almacenados. Es el sistema que presenta menor complejidad, aunque tiene la limitación de que no puede simular voces diferentes a las de la persona que originalmente grabó los segmentos.

Uno de los aspectos más importantes de este tipo de síntesis es el de seleccionar un tipo y tamaño de segmento de acuerdo con el tipo de sistema que se quiere desarrollar. En unidades grandes, se requieren menores puntos de concatenación, por lo que se obtiene un mejor

control de coarticulación, pero se requiere un número muy grande de unidades. En unidades pequeñas, se tienen más puntos de concatenación, pero se reduce el número de unidades requeridas.

Las unidades más grandes son las frases y las palabras. Estas funcionan bien para sistemas que tienen un vocabulario limitado, por ejemplo un sistema que dé la hora del día. Para sistemas de entrada libre hay demasiadas unidades, además de que al ser libre la entrada, puede contener palabras inexistentes o frases mal construidas.

Otra unidad que se podría considerar es la de las sílabas. Su número es considerablemente menor al de palabras o frases, pero aún es demasiado grande (10 000+). Además, no se pueden almacenar los efectos de coarticulación entre sílabas ni la prosodia. El uso de sílabas como unidad es factible sólo en lenguajes silábicos, como por ejemplo el japonés, donde existen menos de 100 sílabas. Debido a estos problemas, no existe al momento ningún sistema de conversión texto-habla que utilice estas unidades.

La siguiente unidad que se puede considerar es la de los fonemas. Su número es bastante reducido (normalmente entre 25 y 50, según el idioma). Los fonemas presentan el problema de la falta de información de coarticulación, por lo que son poco usados; aunque en muchos sistemas se utilizan los fonemas como unidades lógicas que son transformadas a la unidad correspondiente después de su análisis en el proceso de concatenación.

Otra unidad es la demisílaba, que representa la parte inicial y final de las sílabas. Su número es grande pero aceptable (aproximadamente 1 000). Estas cubren un buen número de problemas como la coarticulación, algunos alófonos y requieren menos puntos de concatenación que los fonemas. Su número es grande, pero aún aceptable, desafortunadamente, su número no puede ser determinado fácilmente y hay algunas combinaciones que no pueden ser generadas con demisílabas, por lo que normalmente se usan sólo en sistemas mixtos.

Un tipo de interpolación es el PSOLA [6] (Suma con Traslape Sincronizado al Tono). En éste, se toman los diferentes segmentos y se suman con un cierto traslape, ajustando por medio de ventanas centradas a una distancia igual a la frecuencia fundamental estimada previamente, con lo que, además de mejorar los puntos de concatenación, se puede modificar el tono al modificar la frecuencia e interpolar puntos intermedios o la duración al repetir ventanas. El problema básico de este método ocurre en aquellos fonemas que no tienen frecuencia fundamental, en los cuales se genera un pequeño ruido tonal, debido a la repetición de ventanas iguales. Abordaremos estos problemas posteriormente.

2. Componentes del sintetizador de voz

Como se trató en el Capitulo 1, el esquema básico de un sintetizador de voz consta de la captura y procesamiento del texto, para su conversión en fonemas y su posterior reproducción en voz. A lo largo del tiempo, los dispositivos que intentan hacer síntesis de voz han disminuido en complejidad, volumen, operatividad, etcétera. Los dispositivos dedicados a este tipo de aplicaciones son limitados, dado que los requerimientos computacionales son altos. La mayoría se desarrollan para computadoras personales.

Existen en el mercado intentos de dotar un sintetizador autónomo entre los que destaca el SOP3, un módulo sintetizador de voz que se puede encontrar hoy en el mercado de robótica, se basa en el chip sintetizador WTS701EM/T de Winbond. Su calidad de sintetización es buena, aunque está implementado con fonemas en versión inglesa. El módulo sintetizador de voz SP03 incorpora un amplificador de audio, un regulador de tensión de 3 V con conversión a 5 V, un microcontrolador encargado de facilitar la comunicación con el procesador central, un altavoz de 35 mm, y el chip sintetizador WTS701. La conexión con el exterior se realiza por medio de un bus I^2C. El encapsulado de 56 patas, en formato para montaje superficial, con un espaciado entre patas de sólo 0,5 mm y difícil manejo [23].

A la fecha, todas las implementaciones de este tipo están hechas para el idioma inglés. Sin embargo, existe la necesidad de llevarlos a cabo en otros idiomas, en particular en el español.

2.1. Los microcontroladores

El microcontrolador es un circuito integrado que contiene todos los componentes de una computadora. Se emplea para controlar el funcionamiento de una tarea determinada y, debido a su reducido tamaño, suele ir incorporado en el propio dispositivo al que gobierna. Esta última característica es la que le confiere la denominación de «controlador incrustado» (*embedded controller*). Se dice que es "la solución en un chip" porque su reducido tamaño minimiza el número de componentes y el costo [28, 29].

El microcontrolador es una computadora dedicada. En su memoria sólo reside un programa destinado a gobernar una aplicación determinada; sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar. Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea asignada [28, 29].

Un microcontrolador es una computadora completa, aunque de limitadas prestaciones, que está contenida en el chip de un circuito integrado y se designa a gobernar una sola tarea. El número de productos que funcionan con base en uno o varios microcontroladores aumenta de forma exponencial. Casi todos los periféricos de la computadora (ratón, teclado, impresora) son regulados por el programa de un microcontrolador. Los electrodomésticos de línea blanca (lavadoras, hornos, licuadoras) y de línea marrón (televisores, videos, aparatos de música) incorporan numerosos microcontroladores. Igualmente, los sistemas de supervisión, vigilancia y alarma en los edificios utilizan estos chips para optimizar el rendimiento de ascensores, calefacción, alarmas de incendio, robo [29].

Cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos, de forma que es posible seleccionar la capacidad de la memoria, el número de líneas de E/S, la cantidad y potencia de elementos auxiliares, la velocidad de funcionamiento [30].

Se considera a Intel como el padre de los microcontroladores y al 8048 como el primer microcontrolador de 8 bits (fabricado por Intel en la década de los 70). Otra de las principales empresas del mundo de dispositivos programables es Motorola, que dispone del potente microcontrolador 68HC11. Los microcontroladores PIC de la empresa americana Microchip

se emplean en la actualidad cada vez más debido a su reducido consumo, bajo coste, pequeño tamaño, facilidad de uso y la abundancia de información y herramientas de apoyo [30].

2.1.1. PIC18F452

Los 'PIC' son una familia de microcontroladores tipo *Reduced Instruction Set* (Conjunto de Instrucciones Reducidas, RISC por sus siglas en inglés) fabricados por Microchip Technology Inc. y derivados del PIC1650, originalmente desarrollado por la división de microelectrónica de General Instruments.

El nombre actual no es un acrónimo. En realidad, el nombre completo es PICmicro, aunque generalmente se utiliza como *Peripheral Interface Controller* (Controlador de Interfaz Periférico) [31].

El PIC original se diseñó para ser usado con la nueva *Central Processing Unit* (Unidad Central de Procesamiento, CPU por sus siglas en inglés) de 16 bits CP16000. Siendo en general una buena CPU, ésta tenía malas prestaciones de entrada/salida (E/S), y el PIC de 8 bits se desarrolló en 1975 para mejorar el rendimiento del sistema quitando peso de E/S a la CPU. El PIC utilizaba microcódigo simple almacenado en Read Only Memory (memoria de solo lectura, ROM por sus siglas en inglés) para realizar estas tareas; y aunque el término no se usaba por aquel entonces, se trata de un diseño RISC que ejecuta una instrucción cada 4 ciclos del oscilador [31].

En 1985, dicha división de microelectrónica de General Instruments se convirtió en una filial y el nuevo propietario canceló casi todos los desarrollos, que para esas fechas la mayoría estaban obsoletos. El PIC, sin embargo, se mejoró con EPROM para conseguir un controlador de canal programable. Hoy en día multitud de PICs vienen con varios periféricos incluidos (módulos de comunicación serie, UARTs, núcleos de control de motores) y con memoria de programa desde 512 a 32 000 palabras (una *palabra* corresponde a una instrucción en ensamblador y puede ser 12, 14 o 16 bits, dependiendo de la familia específica de PICmicro) [31].

Microchip proporciona un entorno de desarrollo freeware llamado MPLAB que incluye un simulador software y un ensamblador. Otras empresas desarrollan compiladores C y BASIC. Microchip también vende compiladores para los PICs de gama alta (C18 para la serie F18 y C30 para los dsPICs) y se puede descargar una edición para estudiantes del C18 que inhabilita algunas opciones después de un tiempo de evaluación [31].

Para Pascal existe un compilador de código abierto, JAL, lo mismo que PicForth para el lenguaje Forth. GPUTILS es una colección de herramientas distribuidas bajo licencia GNU que incluye ensamblador y enlazador, y funciona en Linux, MacOS y Microsoft Windows. GPSIM es otra herramienta libre que permite simular diversos dispositivos hardware conectados al PIC [31].

La familia de microcontroladores con memoria Flash de Microchip, identificados por PIC18FXX2, combinan un extenso conjunto de periféricos con un potente núcleo. Los PICs proporcionan un rendimiento de 10 MIPS a 10 MHz y un voltaje de operación en el rango de 2.0V a 5.5V. Los PIC18F242, PIC18F252, PIC18F442 y PIC18F452 ofrecen una solución extremadamente flexible para aplicaciones de control integrado. Los microcontroladores son compatibles con la familia PIC18FXX2 OTP ya existente y proporcionan una ruta de migración integrada a plataformas flexibles basadas en Flash y autoprogramables [30].

Estos dispositivos ofrecen hasta 32K bytes de memoria Flash autoprogramable, 1.5K bytes de ***Static Random Acces Random*** (Memoria Estatica de Acceso Aleatorio, SRAM por sus siglas en inglés) de usuario y 256 bytes de EEPROM de almacenamiento de información. Sus características incluyen un convertidor analógico-digital de 10 bits con hasta 8 canales de entrada, capacidad de programación a bajo voltaje, un rico grupo de periféricos con módulos tanto analógicos como digitales y oscilador con opciones seleccionables [30].

Otras características que incluye son un periférico de comunicación serie sincrona, configurable tanto en modo SPI como en I^2C de 2 cables. Además, cuenta con circuitos programables de caída de alimentación y detector de bajo voltaje, dos PWMs de 10 bits, USART de 9 bits con modo direccionable, tres temporizadores de 16 bits, uno de 8 bits, circuito de “watchdog” y multiplicador de 8 x 8 bits por hardware en un ciclo de instrucción. En la figura 2.1 se observa la descripción de cada uno de los puertos y demás periféricos que lo componen [30].

El ‘Stack’ de los PIC18FXX2 posee 31 palabras de 21 bits, direccionadas por un apuntador de Stack de 5 bits. Cada vez que se produce una interrupción o una llamada a subrutina, el apuntador del Pila se incrementa en 1 y el valor del contador de programa es almacenado en una palabra de 12 bits.

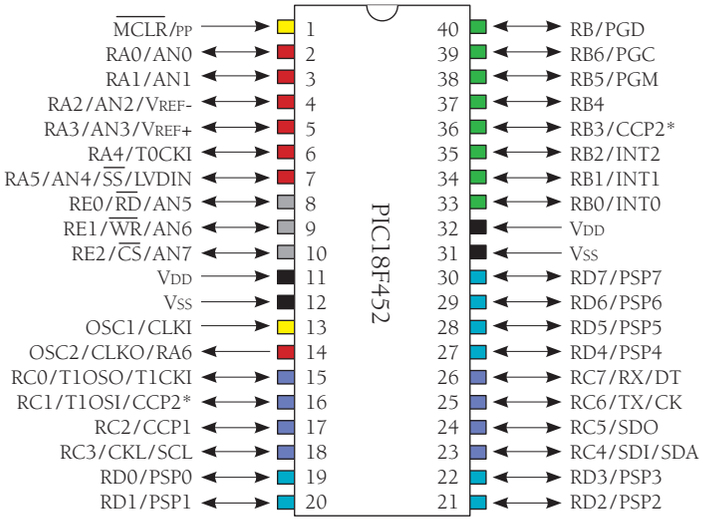

Figura 2.1. Esquema PIC18F452 descripción de cada puerto.

Los puertos de los PIC de la familia 18 en general constan de 3 registros para su operación. El registro TRIS, controla el funcionamiento del puerto. El registro PORT lee los niveles del puerto para entrada [30]. El registro LAT sirve como seleccionador véase la figura 2.2 como el cada uno de los registros se encuentran presentes del puerto B del 1 al 7.

Las referencias son consultas tanto en escritos como citas de Internet, para tener especificaciones más técnicas se deberá consultar el manual del PIC18f452. Todos los datos expuestos en este apartado fueron cotejados con dicho manual.

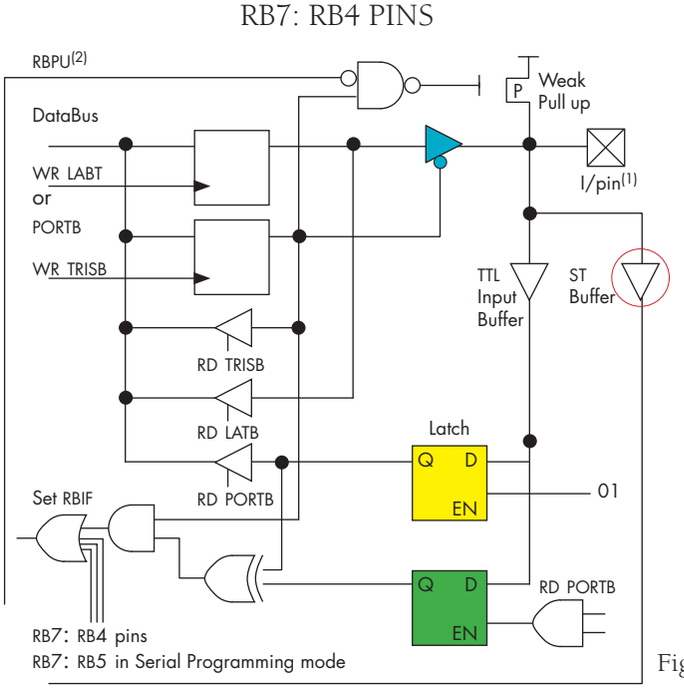

Figura 2.2. Esquema del puerto B del PIC18F452

2.2. Dispositivos de almacenamiento

En una primera clasificación, se puede distinguir entre memorias de almacenamiento masivo, caracterizadas por ser memorias baratas y lentas, y semiconductoras o memorias de estado sólido, más caras y rápidas. En las primeras, la prioridad es disponer de una gran capacidad de almacenamiento, como ocurre en los discos duros, en tanto que en las segundas, la prioridad es disponer de velocidades de acceso rápidas compatibles con la mayor capacidad de almacenamiento posible. Estas últimas son las habitualmente utilizadas como memorias de almacenamiento de programa y de datos en la mayoría de las aplicaciones. Cada tipo de memoria, así como las tecnologías de fabricación, han permitido un espectacular avance en velocidades y escalas de integración [32, 33, 34].

2.2.1. MultiMediaCard/Secure Digital (MMC/SD)

Las memorias denoninadas *MultiMediaCard* (MMC, por sus siglas en inglés), es un estándar de tarjeta de memoria flash. Prácticamente igual a la Secure Digital (SD por sus siglas en inglés), carece de la pestaña de seguridad que evita sobrescribir la información grabada en ella. Su forma está inspirada en el aspecto de los antiguos disquetes de 3'5 pulgadas. Actualmente ofrece una capacidad máxima de 4GB [35].

La memoria que presentó Siemens AG y SanDisk en 1997, [21] se basa en la memoria flash de Toshiba base NAND, y por ello es más pequeña que sistemas anteriores basados en memorias flash de Intel base NOR, tal como la CompactFlash. MMC tiene el tamaño de un sello de correos: 24 mm x 32 mm x 1.4 mm. Originalmente usaba un interfaz serie de 1-bit, pero versiones recientes de la especificación permite transferencias de 4 o a veces incluso 8 bits de una vez. Han sido más o menos suplantadas por las Secure Digital (SD), pero siguen teniendo un uso importante porque las MMCs pueden usarse en la mayoría de aparatos que soportan tarjetas SD (son prácticamente iguales), pudiendo retirarse fácilmente para leerse en un PC [24].

Las MMCs están actualmente disponibles en tamaños de hasta 4GB con modelos de 8GB. Se usan en casi cualquier contexto donde se usen tarjetas de memoria, como teléfonos móviles, reproductores de audio digital, cámaras digitales y PDAs. Desde la introducción de

la tarjeta Secure Digital y la ranura SDIO (Secure Digital Input/Output), pocas compañías fabrican ranuras MMC en sus dispositivos, pero las MMCs, ligeramente más delgadas y de pines compatibles, pueden usarse en casi cualquier dispositivo que soporte tarjetas SD si lo hace su software/firmware [19 y 21].

En el diseño del sintetizador se ocupo una memoria SanDisk de 1 Gigabyte. La posibilidad de subir la capacidad de almacenamiento hasta 4 Gigabytes es viable, debido al protocolo de comunicación empleado. Sin embargo, al incrementar la cantidad de memoria, el microcontrolador bajará su rendimiento. El valor máximo de memoria al que puede ser expandido es de 2 Gigabytes, si es que se desea mayor capacidad de almacenamiento [21].

Las citas son fuentes de Internet, las especificaciones técnicas coinciden con lo expuesto por el manual de SanDisk para memorias MMC.

2.3. Convertidor digital-analógico

La conversión digital-analógica (D/A) es el proceso de tomar un valor representado en código digital y convertirlo en un voltaje o corriente que sea proporcional al valor digital.

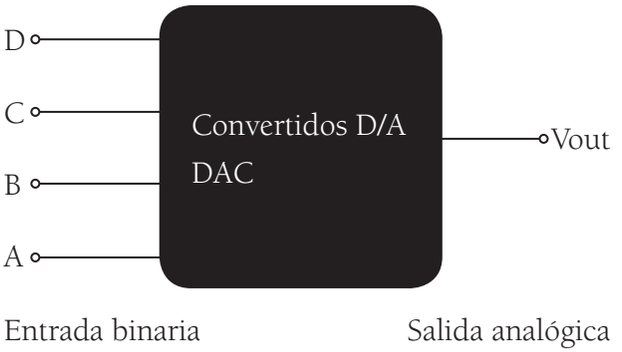

Figura 2.3. Convertidor analogico-digital (DAC, por sus siglas en inglés) de 4 bits.

En la figura 2.3. las entradas digitales D, C, B y A se derivan generalmente del registro de salida de un sistema digital. Los $2^4 = 16$ diferentes números binarios representados por estos 4 bits se enlistan en la tabla siguiente. Por cada número de entrada, el voltaje de salida del convertidor D/A es un valor distinto. De hecho, el voltaje de salida analógico Vout es igual en voltios al número binario (no es así en todos los casos). También podría tener dos veces el número binario o algún otro factor de proporcionalidad. La misma idea sería aplicable si la salida del D/A fuese la corriente [36, 37].

Entrada digital				Salida analógica
D	C	B	A	Vout en voltios
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Tabla 2.1. Entrada binaria a un DAC y su salida de conversión en volts.

El DAC descrito en la tabla tiene una escala de 15 - 0 = 15V, el tamaño de la etapa es de 1V (la etapa es el cambio de la señal de salida ante un cambio de la señal de entrada de un valor a otro consecutivo) [36, 37].

La expresión que define a la resolución de un DAC es la siguiente:

$$res\ (\%) = \frac{tamaño\ de\ etapa}{escala\ total}\ X\ 100$$

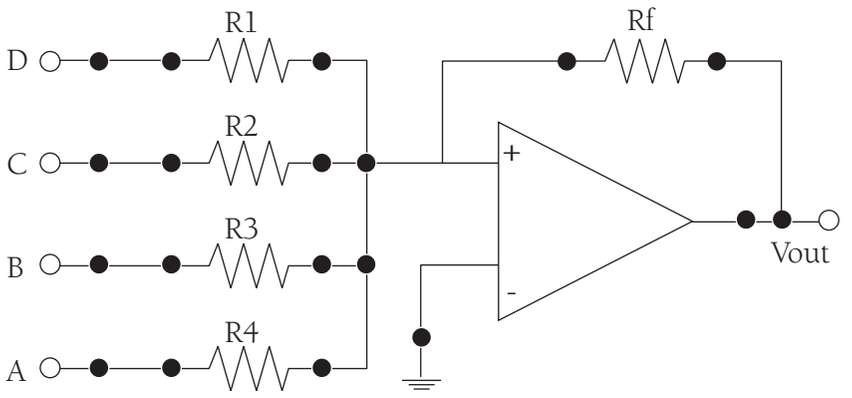

Figura 2.4. DAC construido con un amplificador operacional

Existen varios métodos y circuitos para producir la operación D/A. Uno de ellos es el que se muestra en la figura 2.4. Las entradas A, B, C y D son entradas binarias que se suponen tienen valores 0V o 5V. El amplificador operacional sirve como amplificador sumador, el cual produce la suma con valor asignado de estos voltajes de entrada [36, 37].

Se dispone de una amplia variedad de DAC como circuitos integrados o bien como paquetes encapsulados autocontenidos. Se debe estar familiarizado con las especificaciones más importantes de los fabricantes a fin de evaluar un DAC en una determinada aplicación [36, 37].

- Resolución: La resolución porcentual de un DAC depende únicamente del número de bits. Por esta razón, los fabricantes por lo general especifican una resolución de DAC como el número de bits. Un DAC de 10 bits tiene una resolución más sensible (mayor exactitud) que uno de 8 bits.

- Precisión: Los fabricantes de DAC tienen varias maneras de especificar la precisión o exactitud. Las dos más comunes se las llama Error de Escala Completa y Error de Linealidad, que normalmente se expresan como un porcentaje de la salida de escala completa del convertidor.

El error de escala completa es la máxima desviación de la salida del DAC de su valor estimado (teórico).

E1 error de linealidad es la desviación máxima en el tamaño de etapa del teórico. Algunos de los DAC más costosos tienen errores de escala completa y de linealidad en el intervalo 0.01% - 0.1%.

- Tiempo de respuesta: La velocidad de operación de un DAC se especifica como el tiempo de respuesta, que es el tiempo que se requiere para que la salida pase de cero a escala completa cuando la entrada binaria cambia de todos los ceros a todos los unos. Los valores comunes del tiempo de respuesta variarán de 50ns a 10μs. En general, los DAC con salida de corriente tendrán tiempos de respuesta más breves que aquellos con una salida de voltaje. Por ejemplo, el DAC 1280 puede operar como salida de corriente o bien de voltaje. Su tiempo de respuesta a su salida es 300ns cuando se utiliza salida de corriente 2.5μs cuando se emplea salida de voltaje. El DAC 1280 es un convertidor D/A construido con un amplificador sumador.

Voltaje de balance**:** En teoría, la salida de un DAC será cero voltios cuando la entrada binaria es cero. En la práctica, habrá un voltaje de salida pequeño producido por el error de desbalance del amplificador del DAC. Este desplazamiento es comúnmente de 0.05% FS. Casi todos los DAC con voltaje tendrán una capacidad de ajuste de balance externo que permite eliminar el error de desbalance [36, 37].

2.3.1. Potenciómetro digital MCP41010

El integrado dispone de un potenciómetro con un valor nominal de 10 kΩ, el cual va variando dependiendo de los valores ingresados a él. Con está variación logramos un cambio de tensión a la salida, la cual nos proporciona una salida un tanto escalonada que será filtrada. El potenciómetro es programable mediante el protocolo de comunicación SPI, lo cual facilita mucho su configuración porque el microcontrolador ya dispone del hardware necesario para ello.

Entre las características más sobresalientes de este dispositivo destacan: [48]

- El ajuste del potenciómetro se realiza mediante 8 bits, lo que implica que tiene 256 valores posibles.

- Interfaz de programación SPI.
- Precisión de ± 1 LSB (Last Significant Bit, Bit Menos Significativo).
- Tecnología CMOS de bajo consumo.
- Consumo de corriente de 1 µA en estado de reposo.
- Consumo de corriente máxima de 500 µA en estado dinámico.
- Alimentación simple de 2.7 a 5 V.
- La máxima frecuencia del reloj es de 10 MHz.

En la figura 2.5., podemos observar la distribución de los pines de los pines de conexión.

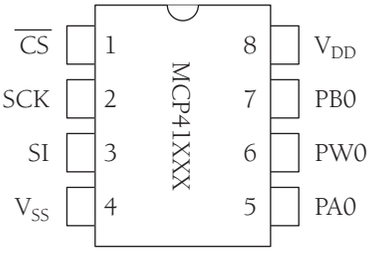

Figura 2.5. Conexión del encapsulado.

2.4. Filtros de salida

Los filtros son redes que permiten el paso o detienen el paso de un determinado grupo de frecuencias (banda de frecuencias). En estos filtros, una de sus principales características es su frecuencia de corte, que delimita el grupo de las frecuencias que pasan o no pasan por el filtro. En el filtro paso bajo pasarán las frecuencias por debajo de la frecuencia de corte y en el filtro paso alto pasarán las frecuencias por encima de la frecuencia de corte.

La curva A (en negro) figura 2.6.:

Muestra una frecuencia central fo, el ancho de banda esta en el rango de f1 a f2. Mientras la curva B figura 2.6., tiene una frecuencia central fo ancho de banda comprende de f3 a f4.

Las dos curvas son de dos filtros con la misma frecuencia central.

Las frecuencias utilizadas para determinar el ancho de banda (f1, f2, f3, f4) se llaman frecuencias de corte y se obtienen cuando la amplitud de la onda (figura 2.6.) cae en 3 decibeles de su máxima amplitud.

La curva B muestra un filtro de mayor selectividad, pues las frecuencias de corte están más cerca de la frecuencia central fo. En este caso el ancho de banda del filtro es menor.

La curva A muestra un filtro de menor selectividad, pues sus frecuencias están más alejadas de la frecuencia central, pero su ancho de banda es mayor.

Para encontrar el factor de calidad de un filtro se utiliza la fórmula: Q = fo / AB, donde:

fo = frecuencia de resonancia

AB = ancho de banda (f2 - f1) o (f4 - f3).

En este caso el factor de calidad del filtro B es mayor.

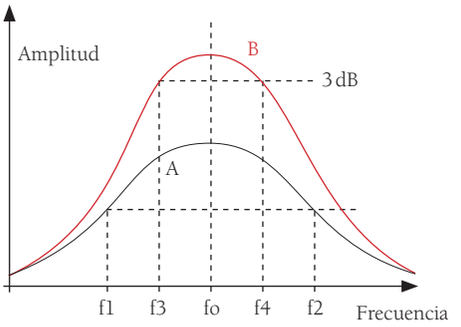

Figura 2.6. Representación de un filtro

El desfase lineal para todas las frecuencias de la banda pasante significa que la frecuencia fundamental y los armónicos de una señal no sinusoidal en la entrada del filtro se desfasarán linealmente a la salida del mismo. Por esto, la forma de la señal de salida es la misma que la de la señal de entrada, si se aplica una tensión en la entrada del filtro y se observa su salida en un osciloscopio, se comprueba que tiene la mejor respuesta al escalón de todos los filtros [4].

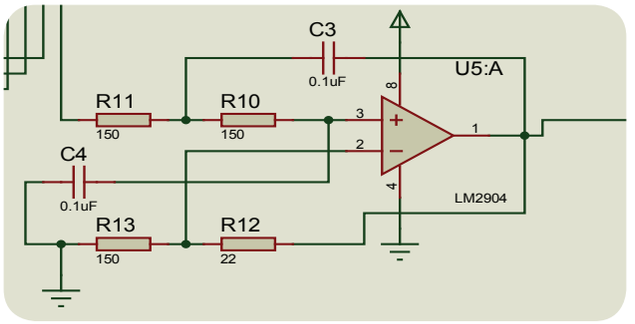

Figura 2.7. Implementación del filtro en el sintetizador

2.5. Amplificador

Se llama a los amplificadores de acuerdo con el rango de frecuencias a las cuales presenta una mayor ganancia, misma que depende de las características inductivas, capacitivas y resistivas del circuito. Es importante notar que la reactancia es distinta para cada frecuencia. Un amplificador de audio es el que está diseñado para amplificar de forma relativamente plana o sea lineal, todas las señales de audio frecuencia (AF). Estas señales están comprendidas entre 10 y 20 000 ciclos por segundo. Cuando se utiliza un amplificador para la voz y no para música, no se requiere un ancho de banda amplio [4, 38, 39].

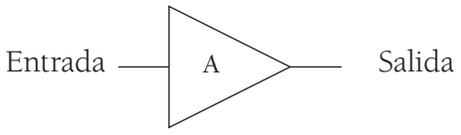

Figura 2.8. Símbolo de un amplificador

En ocasiones, la amplificación puede causar que la señal a la salida del amplificador salga distorsionada causada por una amplificación muy grande. Hay que considerar que un amplificador no puede tener en su salida niveles de voltaje mayores a los de la fuente de alimentación.

Cuando de diseñar un amplificador se trata, su clasificación se determina por las frecuencias con las que trabajará. Cuando los amplificadores están comprendidos dentro de la banda audible se les denomina amplificadores de audiofrecuencia (AF) o amplificadores de baja frecuencia (BF). Otros amplificadores son:

1. Amplificadores de voltaje: Son aquellos que están diseñados para entregar una tensión mayor en su salida respecto a la entrada.
2. Amplificadores de fuerza o de potencia: Son los que pueden entregar mayor corriente como mayor voltaje.

Tenemos 3 clases de amplificadores para las señales de audiofrecuencia [38, 39].

- Amplificadores clase A: Cuando el voltaje de polarización y la máxima amplitud de la señal entrante poseen valores que hacen que la corriente de salida circule durante todo el ciclo de la señal de entrada, se les denomina. Los amplificadores clase A se caracterizan por la baja deformación de la señal, rendimiento y eficiencia relativamente bajos y alta amplificación. Con respecto a la deformación de la señal podría estar en un 5% máximo, imperceptible al oído humano. Estos amplificadores se recomiendan en casos en los que el rendimiento deseado sea moderado y con buena fidelidad del sonido.

- Amplificadores clase B: La característica principal de este tipo de amplificadores es el alto factor de amplificación en corriente.

- Amplificadores clase AB: Estos básicamente son la mezcla de los anteriores. Cuando el voltaje de polarización y la máxima amplitud de la señal entrante poseen valores que hacen que la corriente de salida circule durante menos del ciclo completo y más de la mitad del ciclo de la señal de entrada. Dado que ocupa un lugar intermedio entre las de clase A y B, cuando el voltaje de la señal es moderado funciona como uno de clase A, cuando la señal es fuerte se desempeña como uno de clase B, con una eficiencia y deformación moderadas.

- Amplificadores clase C: Cuando el voltaje de polarización y la máxima amplitud de la señal entrante poseen valores que hacen que la corriente de salida circule durante menos de la mitad del ciclo de la señal de entrada [5, 38, 39].

2.5.1. LM386N

Un amplificador versátil, pequeño, de potencia aceptable para aplicaciones móviles, con pocos componentes externos, económico y con muchas ventajas más, es una excelente opción para su uso en dispositivos pequeños. El LM386 fue diseñado para aplicaciones de bajo voltaje, sin embargo si son agregadas resistencias y capacitancias externos podemos incrementar la ganancia a 46 dB [40].

Entre sus características destacan un bajo consumo de corriente 4 mA, tierra de referencia a la entrada, una distorsión que no sobrepasa el 2% que para nuestros fines resulta muy conveniente, el rango de voltaje de operación va desde los 4 hasta los 12 volts [40].

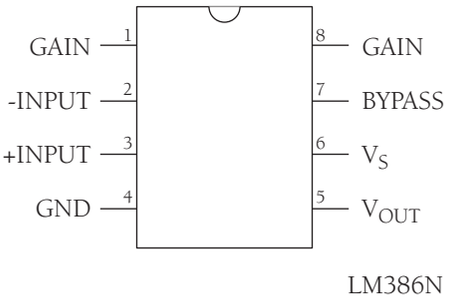

Figura 2.9. Entradas y salidas del LM386N

Para hacer más versátil al amplificador existe la posibilidad de colocar entre los pines 1 y 8 una resistencia de 1.35 kΩ y la ganancia será de 26dB, pero si la resistencia es puesta en serie con un capacitor la ganancia podrá ser puesta en un valor de 46 dB. Con estas características el LM386 fue escogido para la implementación en hardware del sintetizador mostrándose a continuación la conexión con sus componentes externos [40].

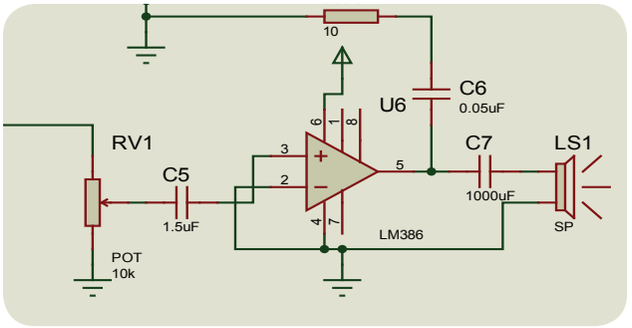

Figura 2.10. Implementación del amplificador en el sintetizador de voz

2.6. Periféricos

Se denominan periféricos a aquellas unidades que sin pertenecer al núcleo, permiten hacer operaciones de entrada y salida de datos, siendo fundamental el que permite la salida de audio en un sintetizador de voz, pero existen dos dispositivos más a los cuales se les debe prestar atención dado que uno funge como entrada de texto y el segundo permite supervisar la correcta entrada de datos.

En un principio la entrada de datos se realiza por medio de un teclado, el cual utiliza comunicación serial, lo cual permite sustituir la entrada por cualquier otro dispositivo capaz de enviar información por medio de ese protocolo. La oportunidad de no depender exclusivamente de un teclado abre la posibilidad de ser una aplicación con múltiples usos.

2.6.1. Comunicación serie asíncrona

La información en una cadena serial de bits esta contenida en su forma de onda dependiente del tiempo: los bits se representan por códigos que se transmiten por un periodo de tiempo fijo. El periodo de tiempo usado para transmitir cada código se conoce como *baud*. Las cadenas seriales de bits generadas por los puertos serie de la PC usan una forma muy simple de codificación. Un bit se transmite durante cada periodo baud, con un bit "1" representado por un voltaje alto TTL (Transistor-Transistor Logic, TTL por sus siglas en inglés) y un "0" por un voltaje bajo TTL. Así, la velocidad en baudios (baud rate, 1/[periodo baud]) de un puerto serie de la PC es igual al número de bits por segundo que se transmiten o reciben [41, 42].

Para enviar información codificada de esta manera, el transmisor y receptor registran el tiempo, el cual define el periodo baud, deben estar a la misma frecuencia y estar sincronizados. Los bits se transmiten como grupos separados, con una longitud típica de 7 u 8 bits, llamados *caracteres*. El nombre carácter se usa porque cada grupo de bits representan una letra del alfabeto cuando el texto esta codificado en ASCII (acrónimo inglés de American Standard Code for Information Interchange). Cada carácter se envía en una armazón (frame) consistiendo de un bit "0", llamado un *bit de inicio*, seguido por el caracter mismo, seguido (opcionalmente) por un bit de paridad y después un bit "1" llamado *bit de paro*. La lógica del bit bajo de inicio

le dice al receptor que está empezando una armazón, y la lógica del bit alto de paro denota el final de la armazón. Un ejemplo de la forma de onda del voltaje usada para transmitir un solo carácter se muestra en la figura 2.11. [41].

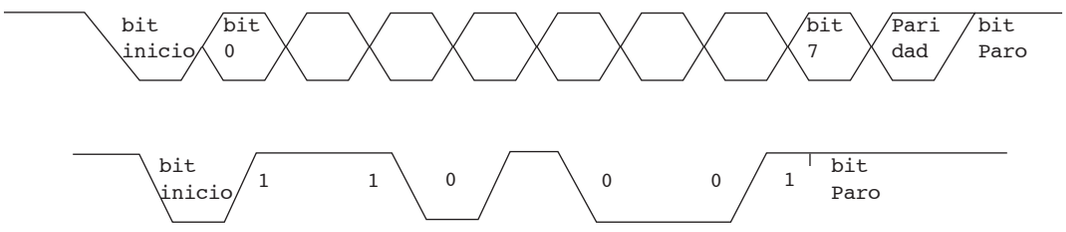

Figura 2.11 Arriba formato de cadena serial de bits asíncrona.
Abajo cadena serial del carácter ASCII "K" (4Bh) usando 7 bits, sin paridad [42].

Esta aproximación de transmitir datos seriales se llama comunicación serial asíncrona porque el receptor se resincroniza el mismo con el transmisor usando el bit de inicio de cada armazón. Los caracteres se pueden transmitir en cualquier tiempo, con un retraso de tiempo arbitrario entre caracteres. Existen también protocolos de comunicación serial síncrona donde los caracteres se envían en bloques sin una armazón de bits circundante. En esta aproximación, el transmisor continuamente transmite señales, con un carácter de sincronización especial que se transmite si no hay datos reales disponibles para transmitir [41].

Los bits dentro de cada carácter transmitido se envían con el bit menos significativo primero, cada bit durando un periodo baud. Los transmisores y receptores seriales se pueden instruir para enviar o recibir de 5 a 8 bits por carácter (ambos deben de estar de acuerdo en cuántos).

Después de que los bits de cada carácter se envían, puede seguir un bit de paridad opcional. El bit de paridad es útil si la línea de datos esta muy ruidosa como para proporcionar una transmisión fiel. El bit de paridad, P, se puede elegir para dar ya sea paridad par o impar. Para paridad par se tiene que, P = 1 si el número de 1's en el carácter es impar y P = 0 si el número es par. Es decir, en la paridad par, P se elige tal que el número de 1's incluyendo P es par. Para paridad impar, P se elige de tal forma que el número de 1's incluyendo P es impar. El receptor local checa para asegurar que la paridad es aun la misma a pesar de que el cable

haya recogido ruido. Si la paridad ha cambiado, entonces algún bit se ha perdido y el receptor pone una bandera de error de paridad en el registro de estado [41].

Después de los bits del carácter y paridad, el transmisor inserta uno o más bits de paro en la cadena de datos. Básicamente la línea debe venir en alto lo suficiente para permitir al receptor estar listo para el siguiente bit de inicio. Típicamente un bit de paro es suficiente, aunque los transmisores pueden ser instruidos para insertar 1, 1.5 o 2 bits de paro. Cuando no se están transmitiendo caracteres, la línea permanece en la lógica de nivel alto del bit de paro.

No es obvio cómo el transmisor y receptor se sincronizan, ya que tienen relojes independientes que sólo nominalmente son iguales. También los cambios de nivel lógico en el inicio de cada periodo baud pueden ser cambiados en tiempo, debido al ancho de banda limitado del medio transportador. La solución estándar a este problema es que el receptor y transmisor usen relojes internos cuyas frecuencias sean 16 veces la velocidad en baudios. Así, cuando el flanco delantero del bit de inicio se detecta, la forma de onda serial entrante se muestrea cada 16 periodos de reloj, empezando con el octavo periodo de reloj después del flanco delantero del bit de inicio. Esto asegura que la forma de onda siempre se muestrea cerca de la mitad de cada periodo baud, haciéndolo tolerante a pequeños corrimientos del flanco y diferencias de frecuencia de reloj del transmisor/receptor [41, 42].

Las velocidades en baudios son: 50, 110, 134.5, 150, 300, 1 200, 2 400, 4 800, 9 600, 14 400, 19 200, 38 400, 56 000, y el no estándar 115 200 (no disponible en la PC original). Las PC's, impresoras y otros dispositivos con frecuencia no pueden funcionar en la velocidad más alta [42].

2.6.2. Liquid Crystal Display (LCD)

Cada píxel de un LCD típicamente consiste de una capa de moléculas alineadas entre dos electrodos transparentes, y dos filtros de polarización, en los ejes de transmisión de cada uno están (en la mayoría de los casos) perpendiculares entre sí. Sin cristal líquido entre el filtro polarizante, la luz que pasa por el primer filtro sería bloqueada por el segundo polarizador [43].

La superficie de los electrodos que están en contacto con los materiales de cristal líquido es tratada a fin de ajustar las moléculas de cristal líquido en una dirección en particular. Este tratamiento normalmente consiste en una fina capa de polímero que es unidireccionalmente frotada utilizando, por ejemplo, un paño. La dirección de la alineación de cristal líquido se define por la dirección de frotación [43].

Antes de la aplicación de un campo eléctrico, la orientación de las moléculas de cristal líquido está determinada por la adaptación a las superficies. En un dispositivo twisted nematic, TN (uno de los dispositivos más comunes entre los de cristal líquido), las direcciones de alineación de la superficie de los dos electrodos son perpendiculares entre sí, y así se organizan las moléculas en una estructura helicoidal, o retorcida. Debido a que el material es de cristal líquido birefringente, la luz que pasa a través de un filtro polarizante se gira por la hélice de cristal líquido que pasa a través de la capa de cristal líquido, lo que le permite pasar por el segundo filtro polarizado. La mitad de la luz incidente es absorbida por el primer filtro polarizante, pero por lo demás todo el montaje es transparente.

Cuando se aplica un voltaje a través de los electrodos, una fuerza de giro orienta las moléculas de cristal líquido paralelas al campo eléctrico, que distorsiona la estructura helicoidal. Esto reduce la rotación de la polarización de la luz incidente, y el dispositivo aparece gris. Si la tensión aplicada es lo suficientemente grande, las moléculas de cristal líquido en el centro de la capa son casi completamente desenrolladas y la polarización de la luz incidente no es rotada ya que pasa a través de la capa de cristal líquido. Esta luz será principalmente polarizada perpendicular al segundo filtro, y por eso será bloqueada y el píxel aparecerá negro. Por el control de la tensión aplicada a través de la capa de cristal líquido en cada píxel, la luz se puede permitir pasar a través de distintas cantidades, constituyéndose los diferentes tonos de gris.

El efecto óptico de un dispositivo twisted nematic (TN) en el estado del voltaje es mucho menos dependiente de las variaciones de espesor del dispositivo que en el estado del voltaje de compensación. Debido a esto, estos dispositivos suelen usarse entre polarizadores cruzados de tal manera que parecen brillantes sin tensión (el ojo es mucho más sensible a las variaciones en el estado oscuro que en el brillante). Estos dispositivos también pueden funcionar en paralelo entre polarizadores, en cuyo caso la luz y la oscuridad son estados invertidos. La tensión de compensación en el estado oscuro de esta configuración aparece enrojecida debido a las pequeñas variaciones de espesor en todo el dispositivo. Tanto el material del cristal líquido como el de la capa de alineación contienen compuestos iónicos. Si un campo eléctrico de una determinada polaridad se aplica durante un período prolongado, este material iónico es

atraído hacia la superficie y se degrada el rendimiento del dispositivo. Esto se intenta evitar, ya sea mediante la aplicación de una corriente alterna o por inversión de la polaridad del campo eléctrico que está dirigida al dispositivo (la respuesta de la capa de cristal líquido es idéntica, independientemente de la polaridad de los campos aplicados) [43].

Cuando un dispositivo requiere un gran número de píxeles, no es viable conducir cada dispositivo directamente, así cada píxel requiere un número de electrodos independiente. En cambio, la pantalla es multiplexada. En una pantalla multiplexada, los electrodos de la parte lateral de la pantalla se agrupan junto con los cables (normalmente en columnas), y cada grupo tiene su propia fuente de voltaje. Por otro lado, los electrodos también se agrupan (normalmente en filas), en donde cada grupo obtiene una tensión de sumidero. Los grupos se han diseñado de manera que cada píxel tiene una combinación única y dedicada de fuentes y sumideros. Los circuitos electrónicos o el software que los controla, activa los sumideros en secuencia y controla las fuentes de los píxeles de cada sumidero [43].

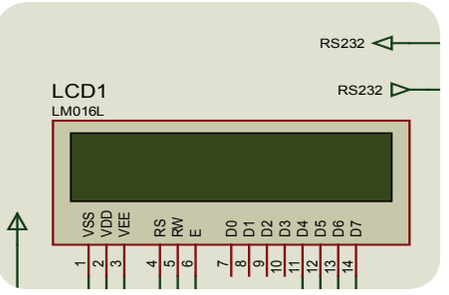

Figura 2.12. LCD utilizada en el sintetizador.

2.6.3. Reguladores de voltaje

Los reguladores de tres terminales son dispositivos que se caracterizan por ser sistemas de regulación completos, que ofrecen utilidades extras, además de la fundamental que es la de estabilizar tensiones continuas, todo esto en un empaque fácil de manejar y conectar.

Como se puede observar, solo consta de tres terminales denominadas: entrada, salida y tierra. La entrada recibe la tensión a regular, la salida proporciona la tensión estabilizada y finalmente la tierra provee un punto común entre las tensiones de entrada y salida, si en este pin se coloca una tensión diferente de cero se pueden obtener otras tensiones diferentes para las cuales fue diseñado el integrado [44].

Básicamente existes 4 tipos de reguladores comerciales disponible: reguladores positivos fijos, reguladores negativos fijos, reguladores positivos variables y reguladores negativos variables. Existen varias empresas dedicadas a la construcción de este tipo de integrados (nacional Semiconductors, Texas Instruments, Fairchild, etc), y cada uno tiene una forma específica de etiquetarlos, por lo que es difícil hacer una generalización [44, 46].

Sin embargo, una de las más usadas es denominar a la familia de los reguladores positivos con los número 78XX en donde XX representa la tensión que tendrá a la salida. Por su parte los reguladores negativos fijos comenzaran con 79XX [44, 46].

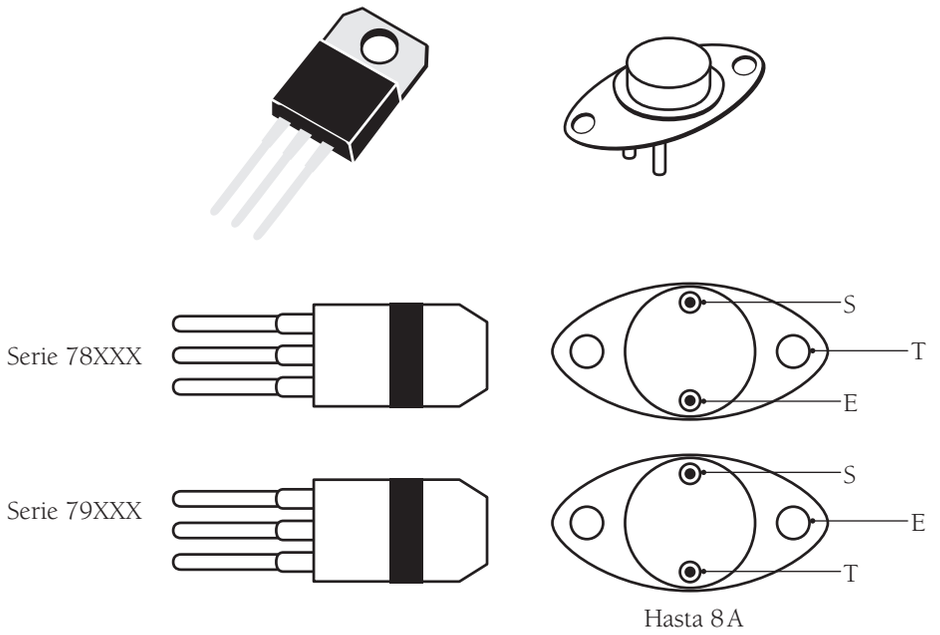

Figura 2.13. Tipos de encapsulado para reguladores [45].

Un factor importante que a veces se pasa por alto, es la distribución de pines (E,S,T), entre familias de reguladores no coinciden como podría esperarse. Es importante ver que tipo de empaque es el más conveniente. En la figura 2.14 se observa los diferentes tipos de encapsulados.

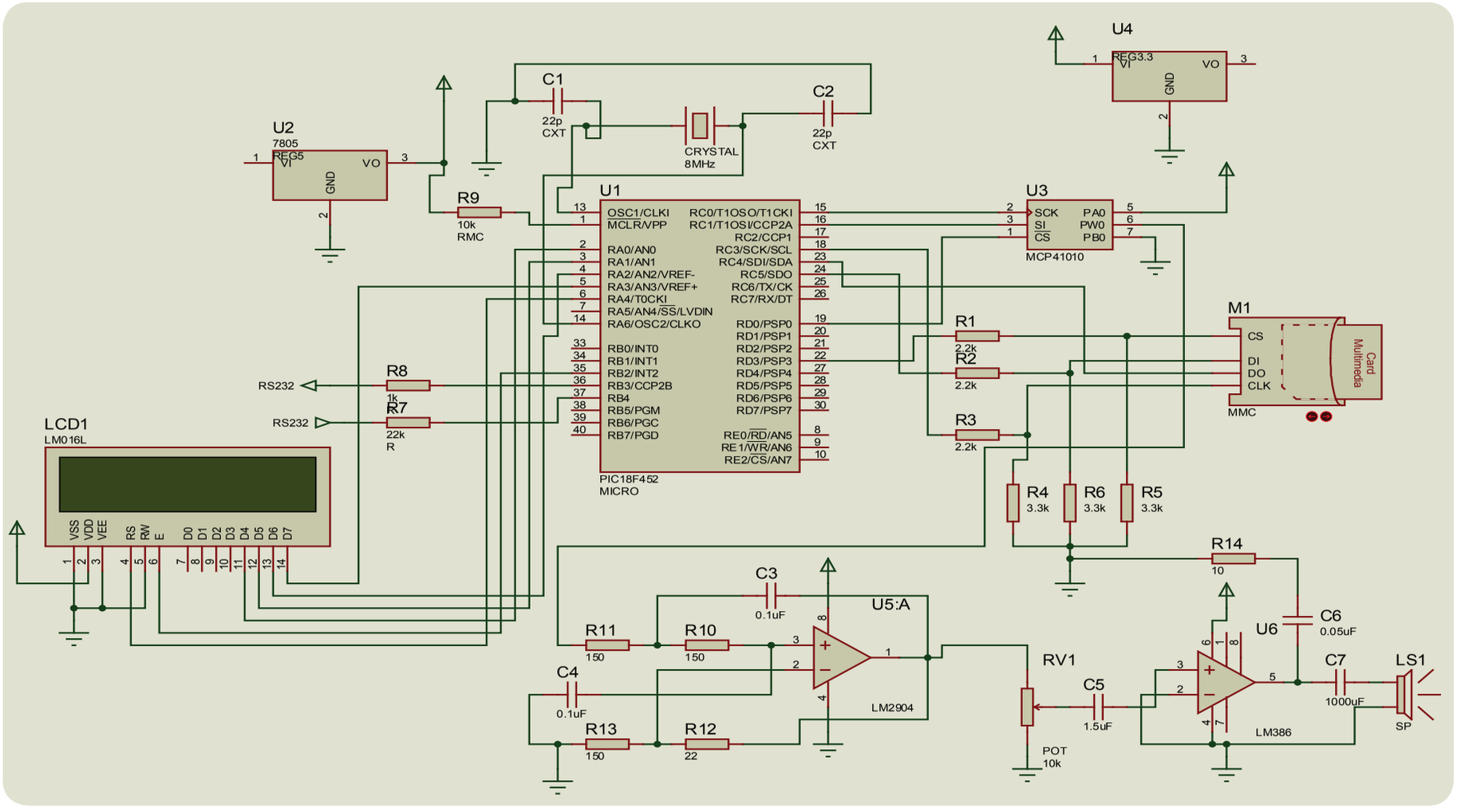

Figura 2.14. Diagrama completo del sintetizador de voz

3. Síntesis de voz con enfoque Multidominio

De acuerdo a, [26] la siguiente generación de sistemas de síntesis de voz son puestos en entre dicho para hacer frente a: expresividad (emociones, estilo al hablar), flexibilidad (multi lenguajes, transformación de la voz), espontaneidad (susurros, risas, pausas, entre otros), [8] y de ser posible cantar, entre otras. Así, los sistemas de síntesis de voz deberán ser capaces de producir el mensaje con la más apropiada prosodia, forma de hablar. Un tema que puede ser afrontado por la extracción de más información del texto de entrada. Por esta razón, una nueva línea de investigación ha emergido en el campo de la síntesis de voz, para extender el análisis del texto más allá de las típicas capacidades de los sistemas tradicionales. Muchos artículos recientes pueden ser encontrados en la literatura enfocados a este tema, por ejemplo, extrayendo la actitud del usuario del texto. O adivinando el subyacente mensaje emocional [26].

En este contexto, un enfoque para incrementar la flexibilidad de los sistemas de síntesis de voz mientras se intenta mantener la calidad de voz equivalente a los de dominio limitado DL TTS, es el desarrollo de sistemas multidominio con un alto grado de naturalidad de la voz. Este enfoque puede ser un paso más en la convergencia de diferentes módulos de sistemas hablados. Usando información respecto de la conversación para mejorar la calidad de la salida sintética.

Por ejemplo, dependiendo de una implementación particular del TTS, conociendo el dominio del texto de entrada nos permite:

1. Ayudar en el proceso de normalización, si la entrada de texto pertenece al dominio matemático, el texto 1 / 2 debería traducirse “mitad” en vez de “enero dos”.
2. Conocer el mejor modelo para la prosodia o considerar diferentes parámetros prosódicos durante la búsqueda de unidades.
3. Considerar un subconjunto del corpus, que enfoque o guíe la unidad de selección ponderando las unidades del dominio de acuerdo a la mezcla de métodos [26].
4. Control del módulo de procesamiento de señal, dependiendo de las características de la voz (calidad de la interpolación de la voz)
5. Activar el módulo de transformación de la voz, para armar el objetivo del dominio si es necesario.

Sin embargo, los sistemas multidominio MD TTS necesitan conocer en tiempo real, cuál es el mejor dominio para el análisis del texto de entrada así como obtener la más alta calidad de voz posible. Así, si la detección del dominio es de forma automática la entrada de texto crudo es necesario redefinir la arquitectura clásica de los TTS, para incluir un módulo de clasificación de dominio.

3.1. Manejo del contexto en Text-To-Speech (TTS)

Existen diferentes formas de clasificar un texto, [9] tradicionalmente la clasificación de texto (TC), esta principalmente enfocada a la temática y caracterización de documentos. En este contexto, los documentos son delimitados por la ocurrencia de los términos clave que constituyen el texto. Así, ignorando la relación y la estructura del texto [26]. La información del tema es usada para organizar el cuerpo de la voz, depender solamente del contenido del texto es insuficiente para dar la inherente naturalidad al hablar, así una adecuada TC para TTS debería considerar ambos aspectos temática y estilo como en otras aplicaciones relacionadas TC detección de genero, entre otras.

La importancia de las marcas de puntuación en los sistemas TC de los TTS, es fundamental, no solo porque permiten una separación del texto, sino que funge como pausas naturales al momento de hacer una reproducción del texto. Existen textos que, su extensión es corta. En esos casos se puede hacer caso omiso a dichas marcas.

El propósito final de cualquier sistema de síntesis de voz es la generación de una voz sintética perfecta y natural, no importando del tipo de entrada de texto. En la búsqueda de esa reproducción encontramos dos estrategias complementarias. Una se preocupa solo por la transmisión correcta del mensaje, mientras la otra propone tener una mayor flexibilidad en el mensaje pera darle más énfasis a la forma en como se dará dicha reproducción de voz [26].

1. Los TTS de propósito general PG TTS, son aquellos que dan prioridad a la flexibilidad de la aplicación que lograr voz sintética de calidad.
2. Dominio limitado DL TTS, se restringen el alcance de la entrada de texto (como lo hecho por los primeros sistemas de síntesis), y dan una calidad alta de voz sintética. El éxito de los sistemas de voz de dominio limitado tiene mayor impacto en aplicaciones multimedia [9], por su efectividad ha ido creciendo el interés en el desarrollo de DL TTS para fines comerciales.

3.2. Trabajo relacionado

Otros campos de investigación relacionados a la voz han adoptado ideologías comparables al multidominio, gracias a su adopción han mejorado la naturalidad y manejo de los sistemas. La investigación en el campo del TTS ha sido largamente una excepción. Esta situación es motivada por dos cuestiones, el hecho de que tempranamente los sistemas TTS fueron bastante capaces de enfrentar el propósito general, en contraste los sistemas de reconocimiento de voz (automatic speech recognition ASR) fueron restringidos y dedicados a tareas específicas. Para lograr razonables desempeños [26]. Segundo, que los sistemas TTS en ocasiones tienen un rol secundario en los sistemas de lenguaje hablado.

El desarrollo de aplicaciones multidominio es una de las recientes direcciones de investigación en los sistemas de lenguaje hablado [26]. La mayoría de los sistemas que adoptan el multidominio, como los de reconocimiento de voz, excluyendo a los de propósito general, son usados para un limitado número de dominios de interacción, ejemplo: cuando en una comunicación abordamos un tema especifico que dominamos el dialogo se vuelve sencillo. Conociendo el dominio de la comunicación (en este contexto, un dominio generalmente corresponde a un tema) permite mejorar el rendimiento.

3.3. Aplicaciones en el sintetizador de voz

La mayoría de los sistemas incluyen el propósito general, operan sobre un conjunto cerrado de dominios de interacción. El manejo de muchos temas en los sistemas de traducción o cualquier otro dispositivo donde se use el TTS, provoca que existan subdominios en los cuales se vuelva complejo el dar una salida adecuada al texto. El mejorar el contexto en el que se habla da como resultado un mejor rendimiento. El manejo de varios contextos, puede provocar que la resolución no sea la adecuada, provocando que se caiga en el extremo opuesto a los de propósito general. La calidad de la voz generada por un sistema así, dejara aún lado su naturalidad por completo.

La implementación de la clasificación por dominio, puede ser definida como una tarea externa al sintetizador o puede ser incluida dentro de la arquitectura del TTS como un modulo automático, la selección del dominio se puede realizar de forma manual permitiendo al usuario hacer el cambio del contexto en el que se quiere reproducir la voz [26].

La implementación de un sintetizador con la base multidominio abre la posibilidad de dejar a un lado la reproducción por concatenación en ciertas frases o expresiones. Una correcta implementación es dar a cada palabra reproducida la entonación correcta dependiendo del contexto. Las limitantes de memoria y velocidad del microcontrolador hacen difícil que sea implementado de esa forma.

La solución implementada en el sintetizador fue dotar de una lista de frases con entonación. La responsabilidad de dar una salida de voz natural o no, recae en el usuario. El microcontrolador al no contar con una memoria de programa grande, no le es posible guardar un texto y procesarlo.

La palabra “lárgate”, en la mayoría de las ocasiones no es dicha de una forma calmada. Existe la posibilidad que alguien no desea usarla de una forma agresiva. Sin embargo, el sintetizador da la posibilidad de hacer una reproducción con una voz natural o con tono de enojo. La lista de palabras o frases guardadas en memoria es limitada, ya que no se realizó ningún análisis estadístico para saber cuales eran las más usadas en ciertos tópicos. Las guardadas en memoria son solo de carácter demostrativo para ejemplificar la posibilidad de escoger un dominio con el sintetizador.

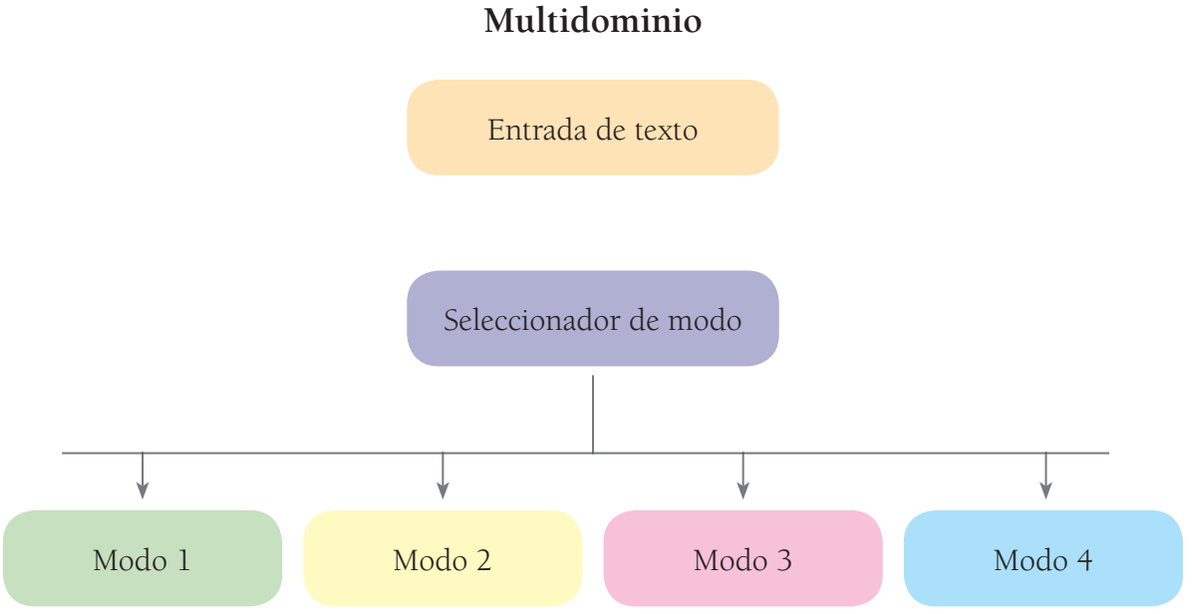

Figura 3.1. Filosofía multidominio utilizada en el sintetizador

En la figura 3.1. se observa como la entrada del texto llega directamente al seleccionador, sin ningún tipo de pre procesamiento. En el microcontrolador se ejecuta el algoritmo que permite escoger la mejor salida. En caso de no contar una salida apropiada a un contexto la reproducción siempre la hace con voz natural. Cada uno de los modos hace referencia a estados de entonación ira, alegría, tristeza

El guardado de cada frase en la MMC ocupa aproximadamente 50 Kbytes. Una cantidad mínima comparado con la capacidad de almacenamiento disponible. La lista de palabras o frases a reproducir puede crecer. El algoritmo de selección contempla todas esas posibilidades.

4. Creación de difonemas y llenado en memoria

4.1. Unidades de reproducción

Las unidades más usadas hasta el momento han sido fonemas, sílabas, semisílabas y difonemas. El hecho de escoger un tipo de unidades u otro viene dado por un compromiso. En general una lengua románica suele tener unos 40 alófono, así que si utilizamos los alófonos como unidades básicas de síntesis, obtendremos una base de datos de unidades de pequeñas dimensiones y fácilmente manejable. Sin embargo, la calidad de voz obtenida en este caso no sería la deseada.

Las unidades utilizadas para la conversión de texto en habla dependen, en gran medida, de la estrategia elegida para la síntesis. En síntesis, existen dos criterios básicos para decidir qué tipo de unidad se elige: el tamaño más adecuado para el almacenamiento en la base de datos y el ruido que se genera al concatenar dichas unidades.

Las unidades que participan en la concatenación pueden ser de diferente tamaño en un mismo sistema, haciendo todas las consideraciones pertinentes para ello. Las unidades que más frecuentemente se han utilizado para la síntesis son cinco.

- Fonema: Es la unidad mínima capaz de diferenciar significados en palabras. Los fonemas agregan mucha flexibilidad a los sintetizadores que los utilizan y desde el punto de vista del espacio, sus requerimientos de almacenamiento no son exigentes. Una limitante en el uso de fonemas para la sintetización, es su propiedad abstracta que engloba variaciones fonéticas contextuales, lo que desemboca en una mala calidad de voz sintética.

- Difonema: Los efectos coarticulatorios tienden a minimizarse en el centro acústico de un fonema, el trozo de voz desde la mitad de un fonema a la mitad del siguiente fonema, como la unidad más satisfactoria para la concatenación.

- Trifonema: Es aquella unidad constituida por un fonema más la mitad del segmento precedente y la mitad del segmento siguiente con objeto de evitar la concatenación que ocupa el centro del trifonema. Su uso dota de más naturalidad aun sintetizador

de voz, pero no todas las frases y palabras pueden formarse utilizándolas con lo que siempre suelen concatenarse con fonemas y difonemas.

- Sílabas y semisilabas: las semisilabas es aquella unidad formada por la mitad de una sílaba estableciendo el centro del núcleo silábico, el número de unidades depende del numero de fonemas, del número de estructuras silábicas y de las combinaciones posibles.

- Palabra: La palabra es la unidad que proporciona más calidad en la síntesis, pero sería necesario contar con una base de datos con todas las palabras de la lengua donde además cada palabra se haya grabado en diferente contextos para solucionar el problema de la coarticulación. Por este motivo, esta unidad se utiliza en dominios restringidos, como en aplicaciones de diálogo.

Podemos ver intuitivamente que cuanto mayor sean las unidades, mayor será la calidad de la voz generada. Pero si utilizamos palabras o frases pregrabadas para sintetizar cualquier tipo de mensaje, las dimensiones de la base de datos serían demasiado grandes y poco manejables. De este modo se llega a un compromiso de calidad memoria que nos lleva a manejar los difonema. Los difonemas son unidades de voz que confinen dos alófonos (combinaciones de vocal + vocal, vocal + consonante).

Como se ha observado, para los sistemas de síntesis de voz basados en concatenación se requieren de segmentos pequeños pregrabados. Los segmentos más comunes son los difonemas, los cuales son usados casi en la totalidad de los sistemas actuales de síntesis de voz. Otro conjunto de unidades usado con cierta frecuencia es el de las palabras completas, y justamente de esas palabras haremos uso para poder dar entonación a la base en la perspectiva multimodo, dado que hay frases o palabras que claramente en el español denotan ira o felicidad.

4.2. Grabación de unidades

El número teórico de unidades es el cuadrado del número total de fonemas. Sin embargo, en la práctica, este número es menor debido a que existen combinaciones inexistentes. Ejemplo de esto serían las combinaciones /r//R/ y /R//r//, que no pueden ser representadas de forma escrita. Una vez que se ha obtenido toda la lista de combinaciones, éstas requieren ser grabadas.

Una vez que se tiene la grabación lista, se deben catalogar los segmentos. Por ejemplo, en el caso de los difonemas se deben dar el inicio y el final de cada segmento que correspondan con el punto medio de los fonemas. La división debe ser siempre en el mismo punto del ciclo de los fonemas para minimizar las discontinuidades. Normalmente se utiliza como inicio de ciclo el punto de máxima amplitud. Además, en el caso de los difonemas también se indica el punto de división entre fonemas.

Esta grabación se llevó a cabo en una PC de escritorio, con una tarjeta de sonido integrada y muestreada a 11 Khz y 16 bits por muestra. Cada segmento se encuentra almacenado dentro de un archivo independiente, cuyo nombre corresponde al difonema almacenado. Posteriormente a cada archivo se le hizo un preprocesamiento antes de ingresarlo a la tarjeta de almacenamiento.

Cada uno de los archivos grabados fue normalizado a un valor entre 0 y 2. Se consideraron 29 fonemas para generar la base (existen 23 fonemas en la lengua española, pero se consideraron 2 juegos de vocales, acentuadas y no acentuadas).

La grabación de cada uno de los difonemas se hizo por medio de la Toolsuite de audio de Adobe, teniendo la base completa en formato crudo con terminación (.pcm). Una vez obtenidos los archivos en formato crudo se convirtieron dichos archivos a formatos de texto teniendo en su contenido a enteros positivos de 8 bits. Ello se logra con Matlab que proporciona la facilidad de abrir archivos de audio con extensión “pcm”. En el siguiente código observamos como se hace la apertura y conversión de cada difonema creado.

```
function cambio (archivo)                    % Nombre de función
cadena = strcat(archivo,'.pcm');             % Apertura de archivo
dd = fopen(cadena);
y = fread(dd,inf,'int16');
x = max(y);
z = y/32768;                                 % Llevar a la unidad
sound (z,11000,16);                          % Reproducción de sonido
v=z+1;                                       % Eliminar negativos
max (v);
k = v*127;                                   % Trasformar a 8 bits
k1=fix(k);                                   % Solo enteros

plot(y); title('señal inicial'); xlabel('tiempo'); ylabel('amplitud');
```

Listado 4.1. Conversión de .pcm a .txt de 8 bits.

En el listado 4.1. se observa el código con el cual fue posible pasar cada uno de los difonemas y frases que iban a ser guardadas en memoria (MMC), a continuación se observa en las gráficas como la señal de entrada es modificada de sus valores iniciales para dejarla solo en enteros positivos. También es visible que la señal no es deformada y se cuenta con la mayor parte de la información, asegurando que al momento de la reproducción no diferirá mucho de la voz grabada originalmente.

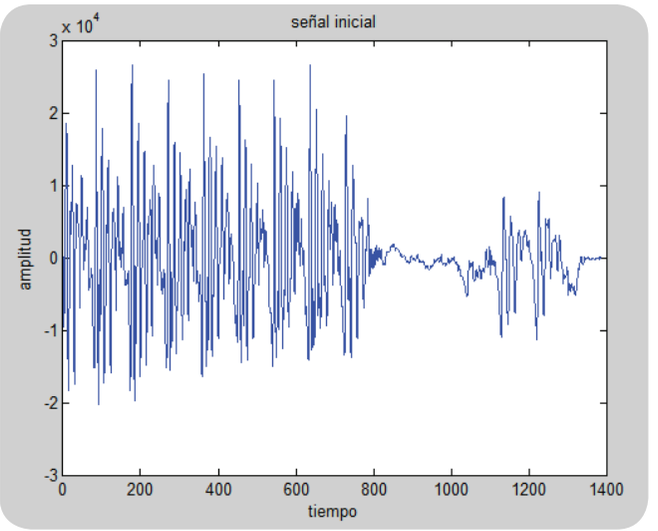

Figura 4.1. Señal original abierta de archivo pcm.

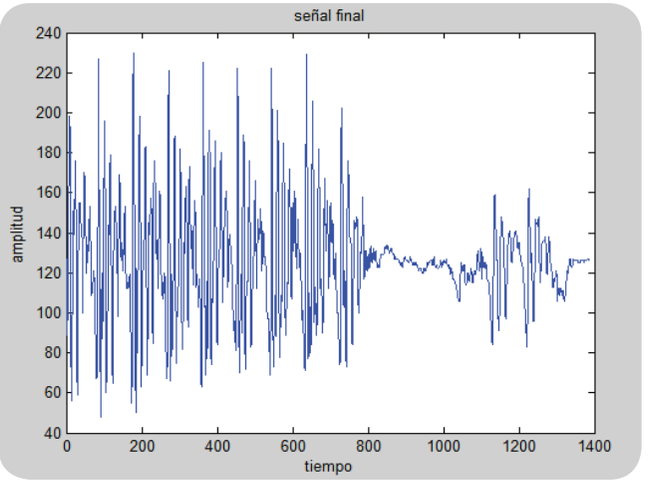

Figura 4.2. Señal de salida después del tratamiento en Matlab.

Se observa como en la señal original esta en el rango de -2 hasta 2, además de no ser enteros, los datos son altamente complicados para ser usados en el microcontrolador. En la figura 4.2 todos los valores son enteros que van desde 0 hasta 240, con ello aseguramos la no saturación y homogenizar a todas las señales en esos umbrales.

Una vez que se tiene cada uno de los archivos en formato *.txt estos pueden ser introducidos en la memoria, se deben cumplir ciertos requerimientos para la lectura, concatenación y posterior reproducción.

A los archivos que fueron introducidos en la memoria se les colocó una cabecera. La primera parte hace referencia a la dirección en la cual será guardado. En la siguiente parte de la cabecera se da el número de bloques. Si un archivo es de 2 Kbytes el número de bloques será igual a 4, es importante hacer notar que las lecturas y escrituras a memoria solo permiten bloques de 512. Las Marcas de periodo se refieren a la cantidad de ceros contenidas en el archivo que deberán ser eliminados a la hora de reproducción, para reducir el ruido en cada traslape. Por último se dan los datos en enteros de 8 bits

Los fonemas guardados en la unidad de almacenamiento presentan la siguiente estructura:

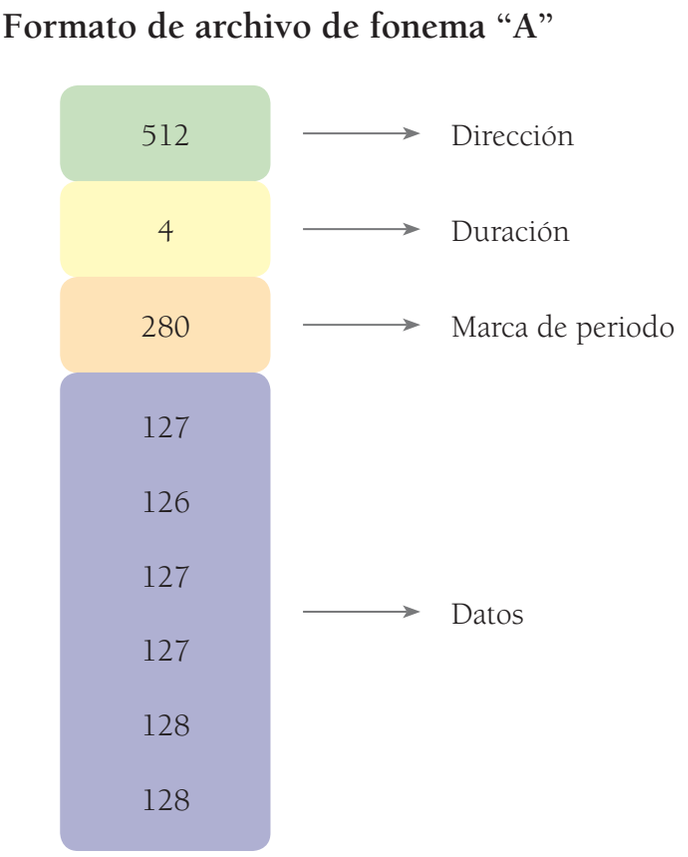

Diagrama 4.1. Estructura de archivo guardado en MMC.

5. Algoritmos de síntesis de voz

En 1979 se introdujo el algoritmo de escalamiento armónico en el dominio del tiempo. En este algoritmo se extraen segmentos en ventanas a intervalos iguales al tono de la señal; estos segmentos se pueden duplicar u omitir para ampliar o disminuir la duración del fonema, así como modificar el tono de la señal. En el caso de que no sea una señal sonora, se utilizan ventanas equiespaciadas a un intervalo predefinido.

Este algoritmo está basado en un experimento de 1959 (Miller, Licklider). En este experimento se demostraba que la señal de voz era redundante dentro de ciertos intervalos de tiempo, por lo que algunos segmentos pueden ser omitidos o repetidos sin alterar la comprensión.

Otro algoritmo basado en este experimento es SOLA [2 y 6] (Synchronized Overlap Add Method–Método de suma de traslape sincronizado), introducido en 1985 (Roucos, Wilgus). Este algoritmo es no recursivo y toma en cuenta las características de la transformada de Fourier de la señal de voz. En este algoritmo, se juntan los dos segmentos a concatenar y se traslapan hasta obtener el punto donde su autocorrelación es máxima. Una vez obtenido este punto, se promedian ambas señales en una ventana de traslape para obtener un punto de unión más suave, manteniendo las características de magnitud, fase y tono de la señal.

Otra ventaja de este algoritmo es que requiere poca carga computacional, por lo que puede ser usado en aplicaciones de tiempo real. Debido a esto, versiones modificadas de este sistema son muy utilizadas en sistemas de síntesis modernos.

Algunos de los algoritmos basados en SOLA son el PSOLA (Pitch Synchronous Overlap Add) y el TD-PSOLA [2] (Time Domain Pitch Synchronous Overlap Add), introducidos en 1987 (Lukaszewicz y Karjalainen), en los cuales no se usa ningún modelo paramétrico, sino se trabaja directamente con la señal original de voz, lo que simplifica aún más los cálculos y se evitan distorsiones ocasionadas por el uso de parámetros en lugar de la señal original. El PSOLA y su versión modificada TD-PSOLA, son variaciones de SOLA, que usan escalamiento armónico, sobreponiendo las señales, promediándolas y después modificando sus propiedades por medio de ventanas centradas en los puntos de amplitud máxima de cada periodo.

Un problema que se presenta con este algoritmo es en las señales que no tienen tono, donde al repetir ventanas para modificar la longitud del fonema, se crea ruido tonal. Una manera de reducir esto es invertir las ventanas que se repiten (Charpentier y Moulines, 1989) [6].

Otro problema que se presenta con este algoritmo, es que se requiere un intenso proceso previo en la base de datos para obtener segmentos que cumplan con las características necesarias.

En el sistema básico se requiere generar gran cantidad de segmentos para descartar aquellos que no cumplan con las características; pero se han generado métodos alternativos de resíntesis de segmentos que permiten modificar los segmentos para que puedan ser utilizados.

Ejemplos de estas modificaciones son el LP-PSOLA (acrónimo inglés de Linear Prediction Pitch Synchronous Overlap) (Charpenier y Moulines, 1990) [6], donde se utiliza un predictor lineal y a la salida de éste se aplica el algoritmo PSOLA, y el MBR-PSOLA (acrónimo inglés de Multi-Band Re-Synthesis Pitch-Synchronous Overlap-Add) [2] (Dutoit y Leich en 1993), donde se modifican los parámetros de acuerdo con el tono de cada segmento y finalmente se ajustan para que coincida con el del siguiente segmento. Este proceso sólo se lleva a cabo en segmentos que son sonoros, para evitar los efectos de ruido armónico en segmentos sordos.

5.1 Time domain pitch synchrounous overlap-add (TD PSOLA)

El nombre TD-PSOLA se deriva de Time Domain Pitch Synchronous Overlap Add (Suma con traslape sincronizada al periodo en el dominio del tiempo) y es uno de los métodos más recientes de síntesis, al haber sido introducido al inicio de los 90's [6].

Una característica muy importante de este método es que funciona directamente sobre la señal de audio original, en lugar de en una señal parametrizada, como en la mayoría de los demás métodos, con lo que se evita introducir distorsiones debidas al cálculo de los parámetros.

Para poder utilizar este método se requiere tener dentro de la base de datos para cada segmento una lista donde se incluye el inicio de cada ciclo dentro de los segmentos (marcas de periodo). El inicio de cada ciclo se ubica en un máximo local dentro de la señal. Para aquellos segmentos no sonoros se agregan marcas a una distancia predeterminada. También se debe agregar a cada marca una indicación de si corresponde a un segmento sonoro o no.

El método consiste en tomar ventaneos de la señal original, centrados en cada una las marcas de periodo y con longitud de dos periodos. Después se calcula una nueva posición para las marcas de acuerdo con la nueva frecuencia que se desea y se reacomodan los segmentos, previamente multiplicados por una ventana de Hanning, sumando los traslapes. Además para modificar la duración se pueden repetir o eliminar ciertos segmentos de la señal.

Este proceso se puede observar en la figura 5.1., donde se muestra cómo de la señal original (parte central) se reacomodan los centros de las ventanas para aumentar (parte superior) o disminuir (parte inferior) la frecuencia. Para mantener el tiempo se eliminan o repiten ventanas de la señal.

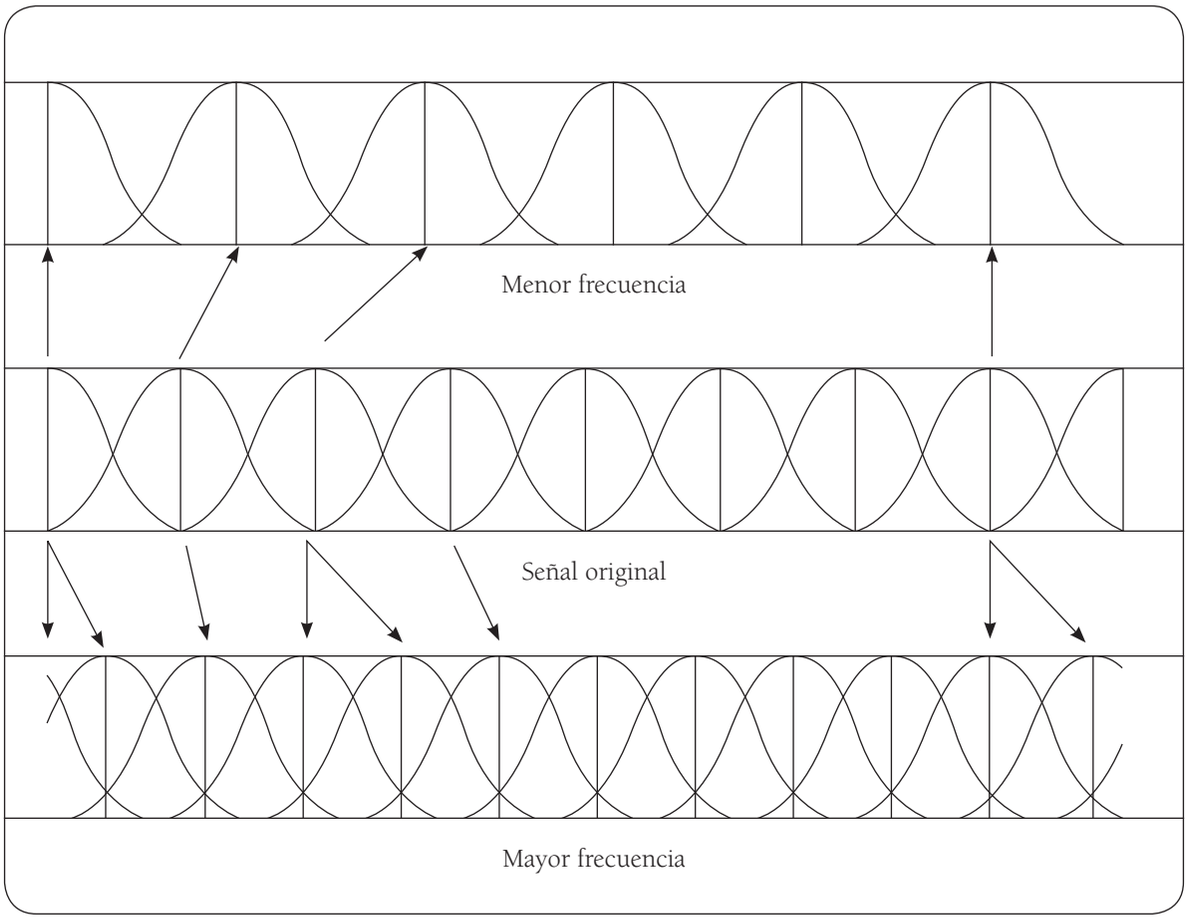

Figura 5.1. Re-síntesis utilizando el algoritmo PSOLA.

5.2 Método de mezcla de tramas

Una de las variables que afecta a la calidad de voz sintética es la modificación del parámetro prosódico de la duración de los alófonos, los difonemas almacenados en la base de datos cuentan con duraciones diferentes, así si deseamos una menor o mayor duración tendremos que procesar la señal de voz para conseguirlo. Esto se puede lograr con PSOLA como ya se vio, en principio parece ser una buena solución, sin embargo aparecen efectos indeseados que merman la calidad del habla generada. Al repetir la misma trama idéntica se forma una señal perfectamente periódica que produce una sensación extremadamente sintética. Además el hecho de tener una evolución espectral produce un efecto de discontinuidad en la unión de los difonemas que en un caso de duración no modificada.

Por otra parte, cuando intentamos disminuir en exceso la duración de la señal eliminando las últimas tramas del primer alófono y las primeras del segundo, podemos llegar al caso de quitar la parte estable del alófono, con lo cual obtendríamos una señal compuesta solo de transiciones [47].

Al mantener el punto de unión en el centro del alófono y realizar una combinación lineal con las tramas de la primera con la segunda, se suaviza la transición entre los difonemas y los efectos de discontinuidad disminuyen. La combinación se realiza con factores de ponderación en forma de rampas complementarias de forma que las dos forman la unidad [47].

Este proceso temporal repercute en el dominio de las frecuencias de manera que los formantes del primer alófono varían progresivamente hasta el segundo.

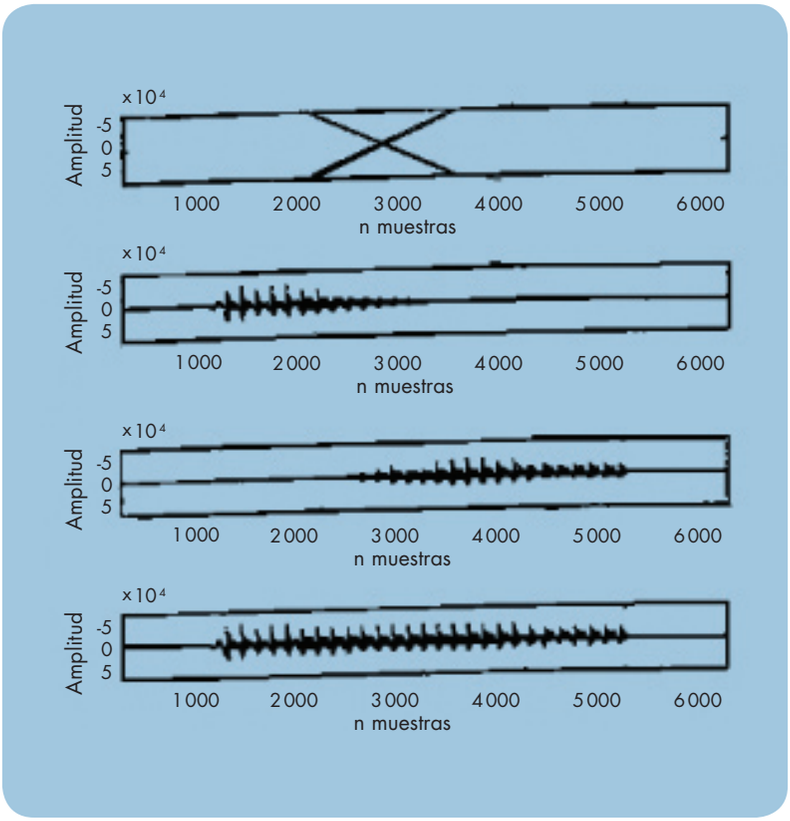

Figura 5.2. Formantes por mezcla de tramas [47].

Lo anterior expuesto se ve reflejado en el siguiente listado 5.1. donde se implementa dicha ideología, el código se encuentra en Matlab, posteriormente fue pasado a PicBasic, lenguaje utilizado para programar el microcontrolador. Observamos que fue necesario hacer un recorte de la señal, esa disminución de datos se hizo de forma empírica a prueba y error para cada uno de los difonemas guardados.

Cada una de las líneas cuenta con la explicación de cómo va el orden y ejecución, es importante ver que en su implementación en el microcontrolador esta operación se hace tantos difonemas vayan a ser mezclados.

```
 function traslape (arc1,arc2)
cad1 = strcat(arc1,'.pcm');
cad2 = strcat(arc2,'.pcm'); % Recibe archivo
a1=fopen(cad1);
a2=fopen(cad2); % Abre archivo
sal1 = fread(a1,inf,'int16');
sal2 = fread(a2,inf,'int16'); % convierte archivo
sali1 = sal1/32768;
sali2 = sal2/32768;
t1 = size(sali1)
t2 = size(sali2)
final = t1(1) + t2(1) %+ t3(1) % longitud completa
menos=50; % estimación de primer difonema
menos1=50; % estimación del segundo difonema
fin1 =t1(1)- menos;
for i=1 :fin1 %comienzo de leido de señal
 salida(i) = sali1(i); % guardado
 end

 for j=1 : menos % comienzo de traslape eliminado 50 datos
 % y multiplicado constante lineal para su
 % suma de traslape
 i=i+1;
 salida(i) = (sali1(i)* .25) + (sali2(j)*.25);
 end

 fin2 = t2(1) - menos;
 for j = menos :fin2 % termina el difonema 2
 i=i+1;
 salida(i) = sali2(j);
 k=j;
 end
```

Listado 5.1. Código Mezclado de tramas en Matlab

Se tomaron los difonemas “pa” y “lo” figuras 5.3 y 5.4, en la gráfica podemos observar que ambos cuentan con datos que no son propiamente de la señal que se quiere reproducir, al no contar con archivos constantes, para cada difonema fue necesario saber que proporción de datos debe ser retirado. No se eliminaron desde un principio dado que estos datos sirven como pausa natural, ya que al hablar existe una pequeña pausa que permite que la dicción sea correcta, pero si es muy grande se oye cortada la voz sintética, si no existiera se pierde la entonación siendo esta muy rápida y poco entendible

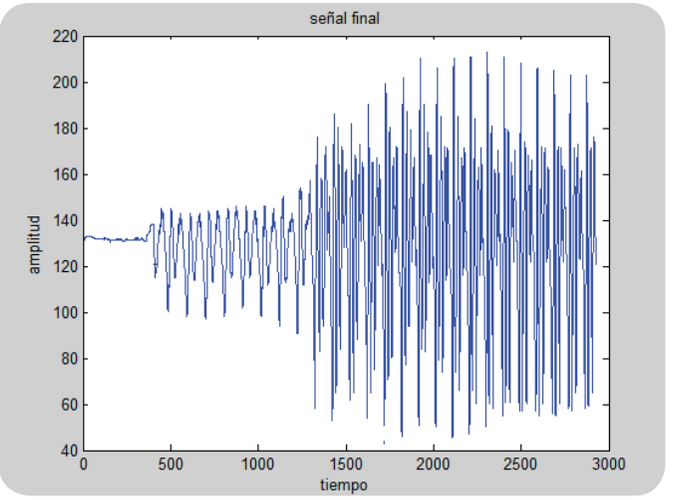

Figura 5.3. Difonema “LO” en su representación amplitud tiempo.

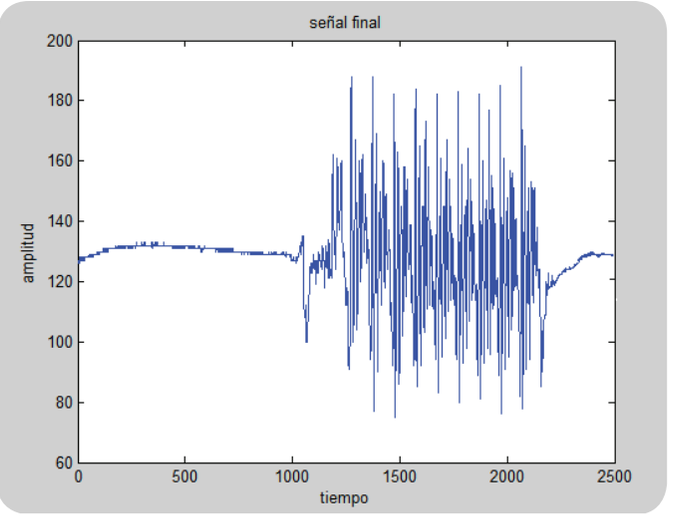

Figura 5.4. Difonema “PA” en su representación amplitud tiempo.

En la figura 5.5. se observa la unión de ambos difonemas, resulta obvio el cambio abrupto de una señal a otra haciendo que a su hora de reproducción exista una pausa prolongada, nada natural. Es por esta razón que se implementó el mezclado de tramas. Con las mismas señales en la figura 5.6. la transición entre un difonema y otro es casi imperceptible y en cuestiones de reproducción, la voz generada cuenta con mayor naturalidad. Uno de los inconvenientes de esta técnica es la forma empírica de obtención de los datos a ser recortados y el factor de disminución de la señal. Si se sumaran en forma directa tendríamos saturación en la señal, sobrepasando los parámetros que se tienen contemplados de 0 a 256.

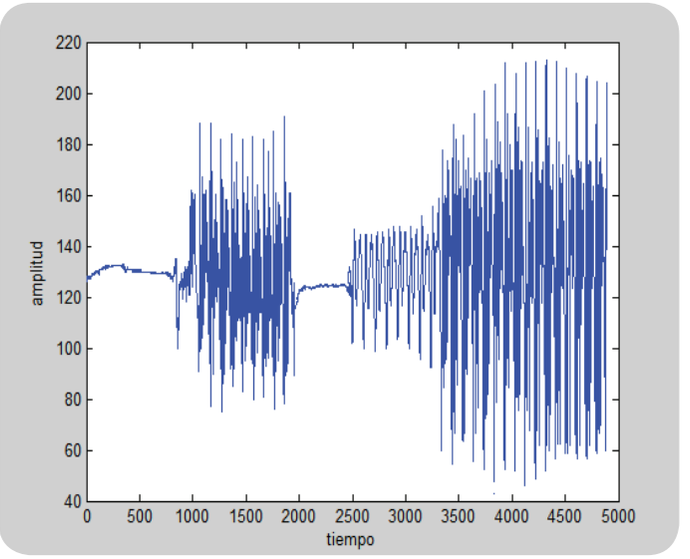

Figura 5.5. Concatenación de difonemas de “PA” y “LO”.

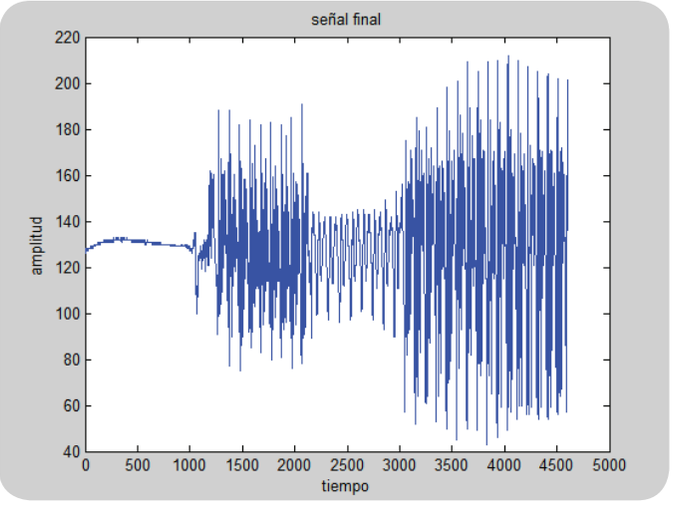

Figura 5.6. Traslape usando mezcla de tramas de difonemas de “PA” y “LO”.

6. Conversión de difonemas en voz

El proceso de conversión de texto a voz, requiere de varios pasos y consideraciones; desde el preprocesamiento dado al los bloques guardados en nuestra unidad de almacenamiento hasta la salida lista para su reproducción. Resuelto los problemas de hardware y aplicado el algoritmo para la concatenación seguiremos paso a paso, cada una de las etapas que realiza el sintetizador de voz, para lograr los resultados esperados.

6.1 Separación silábica en el español

Para separar las palabras en sílabas existen 10 reglas básicas [27]:

REGLA 1.- En las sílabas, siempre tiene que haber por lo menos una vocal. Sin vocal no hay sílaba.

REGLA 2.- Existen conjuntos de consonantes que deben ser mantenidos juntos y pertenecen siempre a la misma sílaba: br, bl, cr, cl, dr, fr, fl, gr, gl, kr, ll, pr, pl, tr, rr, ch, sh.

REGLA 3.- Cuando una consonante se encuentra entre dos vocales, se une a la segunda vocal.

REGLA 4.- Cuando hay dos consonantes entre dos vocales, cada vocal se une a una consonante excepto si son consonantes consideradas inseparables (ver regla 2)

Ejemplos: componer -> com-po-ner

Aprender -> a-pren-der

REGLA 5.- Si son tres las consonantes colocadas entre dos vocales, las dos primeras consonantes se asociarán con la primera vocal y la tercer consonante con la segunda vocal excepto si la segunda y tercera consonantes están dentro del grupo de inseparables. En las combinaciones de cuatro consonantes adyacentes la frontera silábica se situará entre la segunda y la tercera consonantes y ambos grupos deben pertenecer a la regla 2.

Ejemplos: transporte -> trans-por-te

Cumple -> cum-ple

Inscripción -> ins-crip-ción

REGLA 6.- Las palabras que contienen una h precedida o seguida de otra consonante se dividen separando ambas letras, excepto en aquellas combinaciones que pertenezcan a la regla 2.
Ejemplo. Anhelo -> an-he-lo

REGLA 7.- El diptongo es la unión inseparable de dos vocales. Se pueden presentar tres tipos de diptongos:
1) Una vocal abierta + una vocal cerrada
2) Una vocal cerrada + una vocal abierta
3) Una vocal cerrada + una vocal cerrada

Son diptongos sólo las siguientes parejas de vocales: ai, au, ei, eu, io, ou, ia, ua, ie, ue, oi, uo, ui, iu, ay, ey, oy.
Ejemplo: jaula -> jau-la

La unión de dos vocales abiertas o semiabiertas no forma diptongo, es decir, deben separarse en la segmentación silábica. Pueden quedar solas o unidas a una consonante.
Ejemplo: aéreo -> a-é-reo

REGLA 8.- La h entre dos vocales, no destruye un diptongo.
Ejemplo: ahuyentar -> ahu-yen-tar

REGLA 9.- La acentuación sobre la vocal cerrada de un diptongo provoca su destrucción.

Ejemplo: María -> Ma-rí-a

REGLA 10.- La unión de tres vocales puede formar un triptongo. La única disposición posible para la formación de triptongos es la que indica el esquema:
Vocal cerrada + vocal abierta o semiabierta + vocal cerrada

En caso de ser una combinación diferente a esta, el grupo debe ser separada en dos sílabas. Sólo las siguientes combinaciones de vocales forman un triptongo: iai, iei, uai, uei, uau, iau, uay, uey.

De acuerdo con estas reglas existen 4 tipos de sílabas:

1. V -> vocal (1 ó 2)
2. VC -> vocal (1 ó 2) + consonante (1 ó 2)
3. CV -> consonante (1 ó 2) + vocal (1,2 ó 3)
4. CVC -> consonante (1 ó 2) + vocal (1,2 ó 3) + consonante (1 ó 2)

Reduciendo el problema a la identificación de 4 tipos de sílabas la unidad de separación silábica da uso de la simplificación para implementar un algoritmo rápido de identificación, que le permita al microcontrolador no consumir demasiados recursos.

En el listado 6.1. vemos solamente la separación que se hace sobre las de tipo 1 enunciadas anteriormente, bajo esa ideología se programaron los demás tipos de sílabas. Es importante notar que una vez separada la sílaba se le agrega un separador el cual es fundamental para el sistema de control pues con ese identificador se sabe cuanta memoria ha de reservar. El "/" sirve como contador y proporciona la longitud del arreglo de direcciones a ser consultadas por el modulo de lectura.

```
FOR i=0 TO CONTA
m=m+1
l=l+1
L1 = DATOE[i]       'arreglo de entrada
 L2 = " "           'variables para recorrer el arreglo de entrada
 L3 = " "
 L4 = " "
  IF i < CONTA THEN
  J=i+1
  L2 = DATOE[J]
  ENDIF
       IF i < (CONTA-1) THEN
  J=i+2
  L3 = DATOE[J]
  ENDIF
       IF i < (CONTA-2) THEN
  J=i+3
  L4 = DATOE[J]
  ENDIF
       IF i >1 THEN
  J=i-1
  P1 = DATOE[J]
  ENDIF
a = DATOE[i]
GoSub VOCAL  'subrutina que verifica si la palabra es una vocal
IF X = 1 THEN
 a=L2
 GoSub VOCAL
 IF X=1 THEN
  SILABAS[Y]= DATOE[i]
     i=i+1
   Y=Y+1
  SILABAS[Y]= DATOE[i]
  Y=Y+1
  SILABAS[Y]= "/"  ' guarda en nuevo arreglo separando con / la sílaba
    'i=i+1
  Y=Y+1              ' caso de identificación V
NEXT CONTA
```

Listado 6.1. separación silábica en PicBasic

6.2. Tratamiento de difonemas

La separación en silabas del texto de entrada es un paso fundamental, de ello depende que el sintetizador de la salida sea pertinente. Un buen esquema de división silábica es la base para continuar el armado de un sintetizador. Ahora, sólo basta hacer el llamado a las unidades básicas de sonido, acomodarlas en el orden pertinente y con ello estaremos en capacidad de poder llevar a cabo su reproducción.

6.2.1. Llamado y concatenación de difonemas

El llamado del fonema resulta trivial pues basta identificar de cual se trata para apuntar a nuestro lector en la posición adecuada para el inicio de su lectura. Teniendo el número de bloques que va a reproducir y la ubicación de cada una de las partes, comienza la lectura tomando en cuenta qué parte hará el traslape de las señales para su operación final.

Un ejemplo de la palabra separada por sus sílabas es: ba/la/, en la ejecución de código seria analizada y puesta convertida a sus respectivas direcciones.

```
CASE “b”                ‘ Selección en AUX [0] de “b”
SELECT CASE AUX[1]      ‘ Selección en AUX [0] de  a, e ,i ,o, u,
CASE “a”
FINAL[CUENTA]= 139      ‘ Asigna dirección en Final[¿?] de “ba”
CUENTA=CUENTA+1         ‘ Incrementa contador
CASE “e”
FINAL[CUENTA]= 145
CUENTA=CUENTA+1
FINAL[CUENTA]=155
CUENTA=CUENTA+1
CASE “o”
FINAL[CUENTA]=161
CUENTA=CUENTA+1
CASE “u”
FINAL[CUENTA]=167
CUENTA=CUENTA+1
END SELECT
```

Listado 6.2. Direccionamiento de difonemas en PicBasic.

El difonema “ba” cuenta con la dirección 139, recordando que el número en realidad es 8B00 en hexadecimal y en decimal refiera a la posición 35584, el 00 siempre se mantiene fijo y es por ello que solo se asigna la parte alta de la dirección. Esta operación se realiza hasta 64 veces, que es el máximo número de variables que pueden ser manejadas en memoria por el PIC.

Al término de la asignación de cada una de las direcciones tendremos un arreglo con las referencias para ser solicitadas a nuestra memoria. La petición se realiza vía SPI, en el siguiente listado se observa cómo se hace la petición a la memoria.

```
LEE:
 SS=0

 PORTB.2=1
 Serout2 pinout,16468,[“DIRECCIONH= “,DEC DIRH,13]
 Serout2 pinout,16468,[“DIRECCIONL= “,DEC DIRL,13]

 DATOT=255
 GOSUB SPI
 SS=0

 DIRL=0
 GOSUB CMD17
 RES = 0
 GOSUB RESPUESTA
‘ Serout2 pinout,16468,[“LECTURA ANTES WHILE “,DEC CONFIRMA,13]
 ‘WRITE 5,RESPMMC
 RES = 254
 GOSUB RESPUESTA
‘ Serout2 pinout,16468,[“RESPUESTA AL 254= “,DEC CONFIRMA,13]
‘ WRITE 6,RESPMMC
 GOSUB SPI
 L=DATOR
‘ Serout2 pinout,16468,[“longitud= “,DEC L,13]
 FOR I = 0 TO 510
  GOSUB SPI
  PAUSEUS 50
 IF DATOR=0 THEN
  GOTO BRINCA0
 ENDIF
 BRINCA0:

 NEXT I
 TOGGLE PORTB.5
 GOSUB SPI
 GOSUB SPI
 SS = 1
```

Listado 6.3. Lectura de la MMC usando PicBasic y protocolo SPI.

Con cada uno de los difonemas para formar la palabra, se ejecutará el código tantas veces sea necesario. Al eliminar los ceros y mandar llamar a la concatenación, cada bloque que conforma la palabra a ser reproducida se guarda en una localidad especial de memoria para su posterior lectura.

6.2.2. Reproducción de voz

El último paso para realizar la síntesis es la salida de voz, con los difonemas guardados y procesados en su unión, para lo cual basta direccionar a la memoria en la posición 42C1D00 hexadecimal, que equivale aproximadamente a la 70 000 000 en decimal, la cual es una localidad no utilizada por los difonemas o frases con entonación. Al igual que en el proceso de concatenación el llamado a memoria se realiza ubicando la posición y el número de bloques de duración.

Cada dato obtenido de la memoria se pasa totalmente al potenciómetro digital por medio de una comunicación SPI, teniendo cuidado que dicha comunicación se realice a una velocidad de 11 KHz que es precisamente a la frecuencia que fueron grabados tanto los difonemas como las frases.

```
POTEN:

        CS=0

        DATOT=%11011101
        GOSUB SPI
        DATOT=I
        GOSUB SPI
        'Shiftout SI, SCK, MSBFIRST,[%11011101 ]
        'Shiftout SI, SCK, MSBFIRST,[DATOR]
        I=I+1
        CS=1
        GOTO POTEN
```

Listado 6.4. Envío de datos a potenciómetro con PicBasic.

En el listado 6.4. se observa como se inicializa el potenciómetro digital para la recepción de datos y enseguida se transmite de la memoria. Una vez en el potenciómetro, los datos son dirigidos al filtro, el cual elimina el escalonamiento de la señal y le da una transición más suave. La salida del filtro se pasa al amplificador el cual permite a las bocinas tener audio.

Conclusiones

El sintetizador desarrollado logra la mayoría de los objetivos planteados, entre los que destaca: fácil uso, bajo costo, diseño sencillo, buena calidad de voz sintética, fácil traslado y útil para diferentes aplicaciones. El diseño en hardware del sintetizador es único, sobresale el bajo consumo de energía, el uso de componentes comerciales, lo reducido del circuito y el manejo de un microcontrolador como el PIC18F54 que es usado al máximo de sus capacidades.

La adopción de algoritmos de fácil manejo tanto en la separación silábica como en la concatenación de fonemas, hacen del sintetizador una buena opción para aquellas aplicaciones donde el tamaño y el costo deban de ser reducidos. Dado que los algoritmos fueron implementados en un sistema mínimo en hardware.

Además de las ventajas ya mencionadas, la introducción del enfoque multidominio hace al sintetizador, uno de los primeros en adoptar este enfoque. Ahora no solo se trata de dar una salida de voz, sino que cuente con entonación, emita emociones y tenga sentido al contexto de la plática. La implementación del multidominio realizada en el sintetizador es básica, aunque no por ello deja de representar un avance, pues son pocos los sistemas que intentan adoptarla.

El usar una memoria extraíble y de bajo costo permite que la voz sintética sea fácilmente remplazada, ya sea por la de un hombre, una mujer, un niño, una niña. Con una capacidad de 1 Gigabyte de memoria expandible a 4 Gigabytes, ocupando de 7 a 10 Kilobytes una frase grabada, podemos ver que es posible tener una base amplia de palabras o frases en las cuales se desee tener una salida que no sea con voz natural. Para personas con problemas de lenguaje, el sintetizador representa una gran oportunidad para poder ser entendidos sin el uso del lenguaje de señas.

En México no existe un sintetizador con estas características. La mayoría de los sintetizadores son un software diseñado para diferentes plataformas, en las cuales la base de archivos de voz son grabados en español pero con acento de España, ese tipo de pronunciación resulta extraña en México.

El sintetizador responde a las necesidades y requerimientos para cualquier tipo de aplicación de síntesis El tamaño del sintetizador puede ser reducido aún más. El armado en montaje superficial reduce considerablemente el tamaño y el volumen. Sin embargo, el fabricar solo una unidad resulta muy costoso.

Se sientan bases sólidas en el desarrollo de este tipo de dispositivos. El manejo del multidomino, debe mejorarse ampliamente.

Bibliografía

1. Libros y material impreso

[1] del Río Ávila, Fernando, **Diseño de un Sintetizador de Voz por Difonemas** UNAM, Tesis de Licenciatura, 2002

[2] Dutoit, Thierry, **An introduction to text-to-speech synthesis.** Kluwer Academic,1997, Holanda

[3] Paralta Zarate, Felipe de Jesús, **Robot resolvedor de laberinto.** UNAM, Tesis de Licenciatura, 2006

[4] Malvino Albert, Paul, **Principios de Electrónica** Sexta edición, Mc. Graw Hill, 2000. España.

[5] L. Boylestad y Nashelsky, **Electronica: Teoria de circuitos** Sexta edición, Prentice Hall, 1997 México.

[6] Moulines, E. / Charpentier, F. **Pitch Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones.** *Speech Communication* 9, pags. 453-467, 1990

[7] M. Ostendorf and P. Taylor, **The impact of the speech recognition on speech synthesis** in Proc IEEE Workshop Speech Synthesis, Santa Monica, 2002, pp 99-106

[8] Sundaram, Shiva / Narayanan, Shrikanth, **An empirical text transformation method for spontaneous speech synthesizers**, In *EUROSPEECH-2003*, 1221-1224, 2003.

10 [9] Francesc, Alias **Towards High-Quality next-generation text-to-speech: A multimodal approach by automatic domain classification,** IEEE Transactions on audio, speech and language processing, vol. 16 No 7, pags 1340 – 1345. Septiembre 2008.

[11] Apuntes de la materia “Procesamiento Digital de Voz”
Herrera Camacho, Abel

[12] Manual de usuario del PIC16F87X
Microchip Technology inc. 2001

[13] Manual de usuario del TDA2003
SGS Thomson Microelectronics, 1998

[14] Manual de usuario del MMC
SanDisk Corporation. 2005

2. Referencias en Internet

[15] Klatt, Dennis H. Review of text-to-speech conversion for English
http://www.mindspring.com/~ssshp/ssshp_cd/dk_737a.htm

[16] Zur Geschichte der Sprachsynthese 1770-1970.
http://www.ling.su.se/staff/hartmut/kempln.htm

[17] G. Stork, David HAL's Legacy: 2001's Computer as Dream and Reality.
http://mitpress.mit.edu/e-books/Hal/

[18] Lemmetty, Sami Review of Speech Synthesis Technology.
http://www.acoustics.hut.fi/~slemmett/dippa/

[19] Catalog of JEDEC Engineering Standards and Publications online
http://www.jedec.org/Catalog/catalog.cfm Printed in the U.S.A.

[20] Tabla comparativa
http://es.wikipedia.org/wiki/Microcontrolador

[21] MultiMediaCard/RS-MultiMediaCard Product Manual
http://www.sandisk.com/Assets/File/OEM/Manuals/ProdManRS-MMCv1.3.pdf

[22] 10W CAR RADIO AUDIO AMPLIFIER
http://www.st.com/stonline/products/literature/ds/1449.pdf

[23] **Chip de síntesis para lengua inglesa, características y especificaciones.**
http://pdf1.alldatasheet.com/datasheet-pdf/view/91283/WINBOND/WTS701EM/T.html

[24] **Vocoder**
http://es.wikipedia.org/wiki/vocoder

[25] Mattingly, Ignatius **Speech Synthesis for Phonetic and Phonological Models**
http://www.mindspring.com/~ssshp/ssshp_cd/im_home.htm

[26] Xavier Sevillano, Joan Claudi Socoró, and Xavier Gonzalvo **Towards High-Quality Next-Generation Text-to-Speech Syntesis: A Multidomain Approach by Automatic Domain Classification** IEEE Transactions on audio, speech and language processing Vol. 16 NO 17 September 2008 pp 1340-1353.

[27] Ríos Mestre, Antonio **La Transcripción Fonética Automática Del Diccionario Electrónico De Formas Simples Flexivas Del Español: Estudio Fonológico En El Léxico.** Universitat Autònoma de Barcelona
http://elies.rediris.es/elies4/index.

[28] Hans Christian Guevara Parker, Alejandro Real Espinoza. **Sistema de Control y Monitoreo Integrado con WirelessApplication Protocol (WAP) Aplicación: Sistema de Seguridad**
http://www.upc.edu.pe/html/0/0/carreras/ing-electronica/proyectos/MonitoreoWAP.pdf

[29] **Microcontrolador: la solución está en un chip**
http://www.diazdesantos.es/wwwdat/pdf/SP0403414489.pdf

[30] Peralta Zarate Felipe, **Diseño de robot humanoide.** Capítulo 1. Microcontroladores aspectos generales RISC y SICS. Tesis UNAM, 2009.

[31] **El microcontrolador PIC**
http://es.wikipedia.org/wiki/Microcontrolador_PIC.

[32] **Memorias**
http://www.virtual.unal.edu.co/cursos/ingenieria/2000477/lecciones/100501.htm

[33] **Contadores y registros**
http://www.virtual.unal.edu.co/cursos/ingenieria/2000477/lecciones/100301.htm

[34] Alberto Dams.**Tarjetas inteligentes**
http://cactus.fi.uba.ar/crypto/tps/tarje.pdf

[35] **Multi Media Card**
http://es.wikipedia.org/wiki/Multi_Media_Card

[36] Jorge L. Morales Ortiz,Héctor M. Solís Villodas. **Digital to Analog Converter I (DAC)**
www.geocities.com/jorge_f379/**Digital**ToAnalogConverterIDAC.pdf

[37] **Conversión Digital-Analógica**
http://sistemas.itlp.edu.mx/tutoriales/sistdigitales/tem7_1_.htm

[38] **Amplificadores de Audio**
http://www.electronica2000.com/amplificadores/amplif.htm

[39] **Guía de Aplicación para Amplificadores**
http://www.crownaudio.com/pdf/135045.pdf

[40] **Datasheet LM368**
http://www.national.com/ds/LM/LM386.pdf

[41] **Comunicación serial**
http://www.rootshell.be/~wcruzy/cd/comunicacionserial.pdf

[42] **Puerto Serie.**
http://www.rootshell.be/~wcruzy/cd/puertoserie.pdf

[43] LCD
http://es.wikipedia.org/wiki/LCD

[44] José Trelles **Reguladores de voltaje integrados**
es.geocities.com/loslocosproyectos/pdfs/reg_volt_integr_7.pdf

[45] **Datasheet LM78XX**
http://www.alldatasheet.com/datasheet-pdf/pdf/222818/ESTEK/78XX.html

[46] **Datasheet 78XX**
http://es.wikipedia.org/wiki/78xx

[47] Roger Guaus i Térmens, Jaume Oliver **Sintesis de voz utilizando difonemas**
www.sepln.org/revistaSEPLN/revista/21/21-Pag69.pdf

[48] Datasheet MCP41010
http://ww1.microchip.com/downloads/en/devicedoc/11195c.pdf