

TRABAJO ESPECIAL DE GRADO

DISEÑO E IMPLEMENTACION DE UNA UNIDAD TERMINAL REMOTA (RTU) QUE UTILICE LA RED DE TELEFONIA PÚBLICA PARA OBTENER ENERGIA Y PARA LA TRANSMISION DE LOS DATOS

Presentado ante la Ilustre
Universidad Central de Venezuela
por el Br. Alfonso A. Salinas Calderón
para optar al Título de
Ingeniero Electricista

Caracas, 2011

TRABAJO ESPECIAL DE GRADO

DISEÑO E IMPLEMENTACION DE UNA UNIDAD TERMINAL REMOTA (RTU) QUE UTILICE LA RED DE TELEFONIA PÚBLICA PARA OBTENER ENERGIA Y PARA LA TRANSMISION DE LOS DATOS

Tutor Académico: Ing. José Alonso

Presentado ante la Ilustre
Universidad Central de Venezuela
por el Br. Alfonzo A. Salinas Calderón
para optar al Título de
Ingeniero Electricista

Caracas, 2011

CONSTANCIA DE APROBACIÓN

Caracas, 02 de octubre de 2011

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller Alfonzo A. Salinas C., titulado:

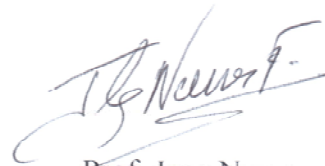
“DISEÑO E IMPLEMENTACION DE UNA UNIDAD TERMINAL REMOTA (RTU) QUE UTILICE LA RED DE TELEFONIA PÚBLICA PARA OBTENER ENERGIA Y PARA LA TRANSMISION DE LOS DATO”

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Electrónica, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.



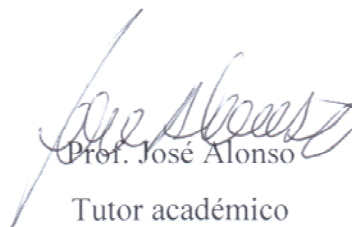
Prof. Luis Fernández

Jurado



Prof. Joao Nunes

Jurado



Prof. José Alonso
Tutor académico

DEDICATORIA

A mi esposa y mejor amiga, Yessibelt, por su apoyo a lo largo del desarrollo de este trabajo.

RECONOCIMIENTOS Y AGRADECIMIENTOS

A todas las personas que hicieron posible que yo pueda culminar esta carrera, en especial a todos los amigos que me apoyaron a lo largo de los años.

Alfonzo A., Salinas C.

**DISEÑO E IMPLEMENTACION DE UNA UNIDAD TERMINAL
REMOTA (RTU) QUE UTILICE LA RED DE TELEFONIA PÚBLICA
PARA OBTENER ENERGIA Y PARA LA TRANSMISION DE LOS
DATOS**

**Profesor Guía: Ing. José Alonso. Tesis. Caracas. U.C.V. Facultad de Ingeniería.
Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Opción: Electrónica.
Trabajo de Grado. 2011. 87h. + anexos**

Palabras claves: Red de Telefonía Fija; Modulación y Demodulación DTMF;
Concatenación de Voz; Alarma de voz.

Resumen. Dado que en la actualidad se requiere mantener un sistema de monitoreo de información en tiempo real sobre sistemas remotos y dada la inestabilidad del sistema eléctrico nacional, surge la necesidad de implementar dispositivos capaces de obtener la información aportada por diversos tipos de sensores y de procesar y transmitir esta información independientemente del estado de la red de energía pública. En vista de que la red de telefonía fija proporciona una alimentación constante y su dependencia del servicio eléctrico es menor que la de otros sistemas de transmisión de datos, esta se presenta como una alternativa para el diseño e implementación de sistemas de monitoreo remoto que, con el uso de microcontroladores, tome energía de la red telefónica y, cuando sea necesario, utilice esta red para transmitir su información. Para esto se usó un regulador de baja pérdida o LDO por sus siglas en inglés, que toma una cantidad muy pequeña de corriente y que no altera el funcionamiento de la central telefónica, para alimentar a un microcontrolador funcionando a su mínima capacidad. Con el fin de transmitir la información se utilizó el procesamiento de voz por medio del PWM del microcontrolador y la detección de tonos DTMF con la ayuda del algoritmo de Goertzel. El resultado obtenido fue una RTU de bajo costo, implementada usando solo un microcontrolador, y que toma la energía de la red de telefonía fija para su funcionamiento.

ÍNDICE GENERAL

CONSTANCIA DE APROBACIÓN	ii
DEDICATORIA	iii
RECONOCIMIENTOS Y AGRADECIMIENTOS	iv
RESUMEN	v
ÍNDICE GENERAL	vi
LISTA DE FIGURAS	ix
LISTA DE TABLAS	xi
LISTA DE GRÁFICAS	xii
ACRÓNIMOS Y SIGLAS	xiii
INTRODUCCIÓN	1
CAPÍTULO I	4
1. PLANTEAMIENTO GENERAL	4
1.1 Planteamiento del Problema	4
1.2 Descripción General del Equipo	5
CAPÍTULO II	7
2. MARCO TEÓRICO	7
2.1 Generalidades de la Red Fija de Telefonía Pública	7
2.1.1 Marcación por Tonos Multifrecuencia DTMF	7
2.1.2 Tonos de Señalización Para el Servicio Telefónico	8
2.1.3 Interfaz Analógica Usuario-Red	10
2.2 El Algoritmo de Goertzel	12
2.3 Modulación Adaptativa Diferencial de Código de Pulsos	16
2.3.1 Compresión	16
2.3.2 Descompresión	18
2.3.3 Algoritmo de Referencia ADPCM de la IMA	19
CAPÍTULO III	20

3. ARQUITECTURA DEL HARDWARE DEL SISTEMA	20
3.1 Módulo Interfaz con la Línea Telefónica	21
3.1.1 Fuente de Poder	21
3.1.2 Diseño de la fuente e interfaz con la línea	26
3.2 Módulo de Amplificación y Filtro	29
3.3 Módulo de Entrada/Salida	33
3.4 Módulo de Control	35
CAPÍTULO IV	39
4. ARQUITECTURA DEL SOFTWARE DEL SISTEMA	39
4.1 Descripción general	39
4.2 Rutina de detección y validación de los tonos multifrecuencia	39
4.3 Software adicional para compresión de audio	46
4.3.1 Software para la grabación de voz	46
4.3.2 Software para la edición de sonido	47
4.3.3 Software para la compresión de audio	47
4.3.4 Software para el almacenaje del audio	48
4.4 Rutina para la reproducción de audio	50
4.5 Rutina Principal de control	52
4.5.1 Rutinas en estado “dormido”	53
4.5.2 Rutinas en estado “despierto”	54
4.6 Rutina Menú Interactivo de la RTU	55
4.7 Rutina Menú Alarma del RTU	62
4.8 Rutinas de Colgar y Descolgar	62
CAPÍTULO V	64
5. IMPLEMENTACION Y PRUEBAS	64
5.1 Implementación del hardware de la RTU	64
5.1.1 Resistencias	64
5.1.2 Condensadores	65
5.1.3 Ajustes por cambio de valores y tolerancias	66

5.2	Implementación del software de la RTU.....	66
5.2.1	Programación del Firmware de la RTU.....	66
5.2.2	Grabación y Compresión de audio.....	67
5.3	Resultados y análisis.....	70
5.3.1	Conexión con la línea de teléfono.....	70
5.3.2	Pruebas de transmisión y recepción de audio.....	70
5.3.3	Pruebas de las funciones de la RTU.....	74
	CONCLUSIONES	75
	RECOMENDACIONES	77
	REFERENCIAS BIBLIOGRÁFICAS	79
	BIBLIOGRAFÍA	81
	GLOSARIO	86
	ANEXOS	88

LISTA DE FIGURAS

Figura 1. Atribución de frecuencias en el teclado [1].....	8
Figura 2: Forma Directa de realización del Algoritmo de Goertzel [4]	15
Figura 3: Diagrama de Bloques Compresión ADPCM [5]	17
Figura 4: Diagrama de Bloques Descompresión ADPCM [5]	18
Figura 5: Módulos de la RTU.....	20
Figura 6: Configuración Típica para el LT3010 [8].....	26
Figura 7: Módulo de interfaz con la línea de teléfono.....	27
Figura 8: Módulo de Amplificación y Filtro	29
Figura 9: Amplificador Operacional OPA4342 [10].....	30
Figura 10: Módulo de Entrada/Salida.....	33
Figura 11: ADM3483 [11].....	35
Figura 12: Módulo de Control.....	35
Figura 13: PIC24FJ64GA002 [12]	37
Figura 14: Diagrama de flujo detección de tonos.....	41
Figura 15: Diagrama de flujos para la validación del tono de 425 Hz	43
Figura 16: Validación DTMF detectado.....	45
Figura 17: Interfaz gráfica del Winspeech	48
Figura 18: Formato de la Imagen MPFS [14]	49
Figura 19: MPFSv2 Image Generator	50
Figura 20: Rutina de reproducción de audio	51
Figura 21: Menu Principal.....	56
Figura 22: Menú Analógico Digital.....	58
Figura 23: Menú Puerto Digital.....	60
Figura 24: Programa sintetizador de audio.....	68
Figura 25: Programa de edición de audio.....	69
Figura 26: Conexión de un teléfono fijo.....	71

Figura 27: Amplificación DTMF	72
Figura 28: Señal de audio generada por el PIC24	72
Figura 29: Señal PWM antes y después del filtro	73
Figura 30: PWM sin señal de audio.....	73
Figura 31: Señal en led testigo en estado de reposo del microcontrolador	74

LISTA DE TABLAS

Tabla 1. Tonos de señalización relevantes	10
Tabla 2: Sistemas de Alimentación de Corriente Continua. [3]	11
Tabla 3: Niveles de Perdidas en Retorno a 600Ω [3]	12
Tabla 4: Formatos de Intercambio Digital de Datos [5]	19

LISTA DE GRÁFICAS

Gráfica 1: Corrientes en Reposo LM2575HV [6]	23
Gráfica 2: Eficiencia LM2575HV [6]	23
Gráfica 3: Corriente Consumida por el MC34063A [7].....	24

ACRÓNIMOS Y SIGLAS

ADPCM: Modulación Adaptativa Diferencial de Código de Pulso.

CCITT: Comité Consultivo Internacional Telegráfico y Telefónico.

DFT: Discrete Fourier Transform

DTMF: Dual-Tone Multi-Frequency.

EEPROM: Electrically-Erasable Programmable Read-Only Memory.

FB: Ferrite Beam (Núcleo de Ferrita)

FCC: Federal Communications Commission

FFT: Fast Fourier Transform

IC: Integrated Circuit

LDO: Low Dropout Regulators.

MOV: Varistor de Metal-Oxido

PCB: Printed Circuit Board.

PIC: Peripheral Interface Controller.

PWM: Pulse-Width Modulation.

ROM: Read Only Memory.

RTU: Remote Terminal Unit.

SD: Secure Digital.

SIDAC: Diodo de Silicio de Corriente Alterna

SPI: Serial Peripheral Interface.

UART: Universal Asynchronous Receiver-Transmitte.

WAV: Waveform Audio Format

INTRODUCCIÓN

Muchos de los procesos que se desarrollan actualmente requieren una respuesta inmediata y en tiempo real a posibles cambios en su estado, lo que hace necesario el diseño e implementación de sistemas que permitan monitorear diversos tipos de sensores, procesar y transmitir la información de éstos y permitir al usuario final tomar las decisiones pertinentes. Adicionalmente, la crisis que presenta el sector eléctrico introduce factores que atentan contra la disponibilidad de los sistemas de monitoreo y transmisión de información que dependen directamente del suministro público de energía. Así pues, es deseable que el diseño e implementación de estos sistemas incluyan alternativas que permitan que su operatividad sea independiente del estado de la red de distribución eléctrica.

La solución comúnmente adoptada para solventar la dependencia del sistema eléctrico nacional se basa en el uso de dispositivos que almacenan energía (UPS) para mantener el sistema en estado “activo” y de la red celular existente para transmitir la información necesaria. Sin embargo, la red celular tiende a verse afectada también por los cortes del suministro eléctrico y la inclusión de elementos almacenadores de energía trae como consecuencia el incremento en los costos de implementación.

En la búsqueda de alternativas que permitan solventar los problemas antes mencionados, se pretende estudiar la posibilidad de utilizar la red telefónica convencional como fuente de alimentación y medio de transmisión de datos para sistemas de monitoreo. La red telefónica, al contrario de la celular, tiende a mantenerse activa aun en el caso de un fallo en el suministro de electricidad pública dado que el consumo se realiza solo en el caso de que el abonado descuelga el teléfono (de lo contrario la central telefónica no presenta un consumo significativo de energía). Adicionalmente, la central telefónica mantiene una tensión constante sobre

los abonados para el funcionamiento de los teléfonos, lo que permitiría utilizar esa tensión para alimentar un dispositivo de bajo consumo de energía que monitoree múltiples actividades y, al estar directamente conectado a la red telefónica, aprovechar este mismo canal para realizar el reporte o transmitir la información necesaria.

OBJETIVOS

Objetivo general:

Diseñar e implementar una unidad terminal remota (RTU) que utilice la red de telefonía pública para obtener energía y para la transmisión de los datos

Objetivos específicos:

- Diseñar e implementar una Unidad Terminal Remota (RTU por sus siglas en ingles) que posea 2 entradas analógicas y 4 entradas y salidas digitales, con capacidad de conexión a un bus digital de campo (MODBUS)
- Diseñar e implementar una fuente de poder que tome la energía de la red de telefonía fija por medio de una fuente conmutada, disminuyendo al mínimo el consumo de corriente a la central
- Diseñar e implementar una interfaz entre el microcontrolador y la línea telefónica que permita realizar una comunicación sonora con cualquier equipo telefónico fijo o móvil
- Diseñar el programa para que el microcontrolador realice las funciones de una RTU, manejo de la línea telefónica y funciones auditivas.
- Simular, utilizando un paquete computacional, el funcionamiento del circuito y el desempeño del programa en el microcontrolador
- Diseñar e implementar un hardware para el montaje físico del circuito.
- Realizar pruebas con el hardware para comprobar su correcto funcionamiento.

CAPÍTULO I

1. PLANTEAMIENTO GENERAL

1.1 Planteamiento del Problema

Dado que en la actualidad se requiere mantener un sistema de monitoreo de información en tiempo real sobre sistemas remotos (desde el estado de un equipo en una fábrica, hasta la temperatura en unidades de refrigeración de alimentos o medicamentos), y en vista de la inestabilidad del sistema eléctrico nacional, surge la necesidad de implementar dispositivos capaces de obtener la información aportada por diversos tipos de sensores y de procesar y transmitir esta información independientemente del estado de la red de energía pública.

Puesto que la red de telefonía fija proporciona una alimentación constante y su dependencia del servicio eléctrico es menor que la de otros sistemas de transmisión de datos, ésta se presenta como una alternativa para el diseño e implementación de sistemas de monitoreo remoto que con el uso de microcontroladores tome energía de la red telefónica y, de ser necesario, utilice esta red para transmitir su información. Además, se presume que al carecer de sistemas de alimentación alternativos y al utilizar la red telefónica fija tanto para alimentar los dispositivos como para transmitir la información, los costos asociados a estos sistemas sean considerablemente menores que los de los sistemas convencionales.

Como una de las posibles aplicaciones que pueden implementarse con el uso de la red de telefonía pública como fuente de alimentación y canal de transmisión de datos, este trabajo propone el diseño de una unidad terminal remota (RTU por sus siglas en inglés) que, con la ayuda de una fuente conmutada, tome energía directamente de la red telefónica para alimentar un microcontrolador que

monitoree diversas funciones a través de las entradas y/o salidas que éste posea y dé un reporte hablado de la situación en su entorno, ya sea por solicitud (al ser llamado por la línea de teléfono) o por una emergencia o alarma que se produzca, en cuyo caso la llamada se realiza automáticamente a un número de teléfono pregrabado.

1.2 Descripción General del Equipo

Lo que se quiere conseguir con este equipo es un sistema independiente de la alimentación del sistema eléctrico nacional. Para lograrlo, se utilizará la energía suministrada por las compañías telefónicas a sus abonados. Dado que la energía que suministran las empresas de telefonía sirve solo para alimentar los teléfonos en el momento de hacer o recibir una llamada cuando se “descuelga” el teléfono, cualquier demanda de energía por parte del abonado es interpretado como “descuelgue”. La central tiene un límite en la corriente que suministra al abonado antes de interpretar un cierre de bucle o “descuelgue”. Para conseguirlo el principal reto en el diseño del equipo es conseguir trabajar por debajo de los niveles permitidos por la central, por lo que se requiere de un medio de administrar eficientemente la energía suministrada por la central y de dispositivos cuyo consumo de energía sea extremadamente pequeños y de la capacidad para inhabilitar un dispositivo si no está siendo utilizado.

En la mayoría de los teléfonos fijos, el descuelgue se produce al cerrar un circuito, lo que produce una demanda de corriente de la central que ésta interpreta como que el abonado desea realizar una llamada, por lo que se puede diseñar un dispositivo que utilice el manejo de corriente para descolgar y colgar la línea, utilizando su máxima capacidad cuando esta la línea descolgada y entrando en un modo de ahorro de energía cuando la línea está colgada, tomando el mínimo de corriente que permita la central sin que se descuelgue la línea.

Además de las características de alimentación explicadas, se requiere que el equipo sea económico, por lo que el número de dispositivos que lo conforman debe ser mínimo. Con esta premisa, se desea diseñar dicho circuito de manera que el microcontrolador realice la mayoría de las tareas incluyendo la decodificación de los tonos DTMF y la reproducción de voz y tonos DTMF para lo cual existen integrados especializados pero que aumentarían el costo final del dispositivo.

Para la decodificación de tonos DTMF se propone la implementación del Algoritmo de Goertzel Modificado, el cual examina la energía de cada uno de los tonos DTMF para determinar cuáles son los dos con mayor valor, y de allí determinar el tono discado. El Algoritmo de Goertzel Modificado no usa muchas constantes por lo que no ocupa mucho espacio en la memoria del microcontrolador y solo realizará los cálculos para las frecuencias de los siete tonos DTMF y la frecuencia de 425Hz (tono de discado), por lo que se estima que el tiempo de cálculo será menor al de la Transformada Discreta de Fourier.

La reproducción de la voz con la que el dispositivo se comunicará se realizará por medio del PWM del microcontrolador y un filtro pasa bajo Butterworth de cuarto orden implementando la Modulación Adaptativa Diferencial de Código de Pulso, o ADPCM por sus siglas en inglés, que es un método de compresión de audio de 4:1. Con este método el audio es grabado, comprimido y luego almacenado en el microcontrolador para luego ser decodificado y sacado por el PWM del microcontrolador. Con esta técnica se pueden almacenar las palabras a ser usadas y luego el microcontrolador ordenará para producir el mensaje deseado. Los tonos DTMF son grabados y guardados de igual forma que las palabras y reproducidos cuando se detecta el tono de marcado para llamar a un número previamente guardado en el microcontrolador.

CAPÍTULO II

2. MARCO TEÓRICO

2.1 Generalidades de la Red Fija de Telefonía Pública

A continuación se describen algunos aspectos fundamentales que el lector debe tener de Telefonía Pública para la mejor comprensión del presente trabajo. Los puntos que se consideran importantes son: Marcación por Tonos Multifrecuencia, Tonos de Señalización Para el Servicio Telefónico y La Interfaz Analógica Usuario-Red.

2.1.1 Marcación por Tonos Multifrecuencia DTMF

Los Tonos Duales de Multi-Frecuencia o DTMF conforman un estándar en los sistemas de telecomunicaciones que ha venido desplazando al antiguo sistema de marcación por pulsos de los inicios de la telefonía.

El sistema de marcación por tonos, como también es conocido, es un sistema de señalización, recomendado por el CCITT para los aparatos telefónicos de teclado. La señalización emplea un código multifrecuencia que transmite dos frecuencias simultáneamente al accionarse un pulsador del teclado. El sistema dispone de 10 cifras decimales y 6 señales de reserva, para un total de 16 señales. Las dos frecuencias que componen cada señal se toman de dos grupos de frecuencias, excluyentes entre sí, denominadas frecuencias inferiores y frecuencias superiores. Cada grupo contiene cuatro frecuencias en un código denominado “2(1/4)”. El grupo de frecuencias inferiores está compuesto por las siguientes: 697, 770, 852 y 941 Hz.

El de las frecuencias superiores por: 1209, 1336, 1477 y 1633 Hz. La asignación de las frecuencias a las diferentes señales en el teclado se muestra en la figura 1. [1]

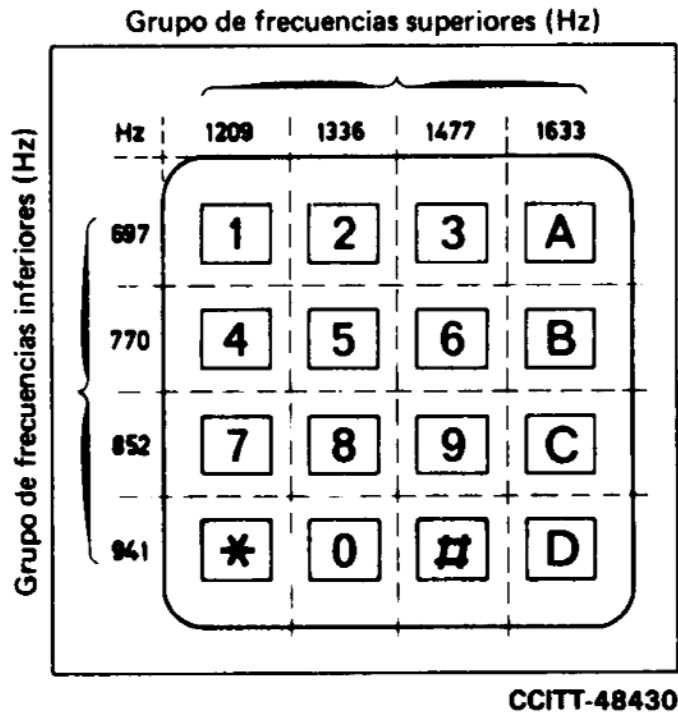


Figura 1. Atribución de frecuencias en el teclado [1]

De las 16 combinaciones asignadas a los dígitos, las redes actuales sólo admiten las correspondientes a las 10 cifras decimales, del 0 al 9, en lo que a encaminamiento de señales se refiere y los correspondientes a los caracteres no numéricos * y # para procedimientos de control de servicios complementarios.

2.1.2 Tonos de Señalización Para el Servicio Telefónico

Las compañías de telefonía implementan un grupo de tonos audibles para el abonado con el objetivo de señalar el estado de la conexión de la línea telefónica. Existen varios tonos de señalización e incluso existen tonos de señalización especiales para una central telefónica o para una compañía de teléfonos. En este

punto trataremos sólo 3 de estos tonos, los más comúnmente usados por las compañías telefónicas: el Tono de Marcar, el Tono de Llamada y el Tono de Ocupado-Congestión. [2]

Tono de Marcar: El tono de invitación a marcar es un tono continuo, de 425Hz utilizado para indicar al abonado en la línea llamante que la central está lista para recibir y analizar los tonos de marcación DTMF. [2][3]

Tono de Llamada: Es un tono de 425Hz acompañado de un periodo de silencio cuyo periodo es mayor, los límites aceptados para el periodo de tono son entre 0,67 y 1,5 segundos, y para el periodo de silencio entre 3 y 5 segundos. El tono de llamada indica al abonado de la línea llamante que se está enviando la señal de llamada al abonado llamado. [2]

Tono de Ocupado – Congestión: el tono de ocupado y el tono de congestión son señales de cadencia rápida en los que el periodo de tono es, en teoría, igual al tono de silencio. La frecuencia del tono es de 425Hz y el periodo total de la señal (periodo de tono + periodo de silencio) debe estar comprendido entre 300 y 1100 milisegundos. El tono de ocupado indica al abonado de la línea llamante que el abonado al que está llamando está usando la línea. Por su parte, el tono de congestión indica al abonado de la línea llamante que la llamada no puede ser completada por encontrarse sobrecargado algún elemento de la red. [2][3]

En la Tabla 1, se indican los tonos descritos anteriormente con los valores utilizados por la mayoría de las compañías telefónicas:

TONO	FRECUENCIA F±ΔF (Hz)	NIVEL N±ΔN (dBm)	CADENCIA t±Δt (ms)
Invitación a Marcar	425 ± 15	-10 ± 2	On: Continuo
Llamada	425 ± 15	-10 ± 2	On: 1000 Off: 4000 Repetido
Ocupado	425 ± 15	-10 ± 2	On: 400 Off: 400 Repetido
Congestion	425 ± 15	-10 ± 2	On: 250 Off: 250 Repetido

Tabla 1. Tonos de señalización relevantes

Señal de Llamada:

Es la señal alterna sinusoidal que se presenta al abonado para indicar que una llamada esta entrado. La corriente de llamada se presenta superpuesta a la tensión de alimentación. Tiene una frecuencia de 25Hz y una tensión igual a 75Vrms. [3]

2.1.3 Interfaz Analógica Usuario-Red.

Alimentación de Corriente Continua desde la Red:

La alimentación de corriente continua desde la red consiste en un generador de tensión o de corriente que, a través de una determinada impedancia y un par simétrico de hilos de cobre, se aplica sobre los dos terminales de la línea del abonado. [3]

La alimentación de corriente a los abonados pueden ofrecerse mediante dos sistemas: Alimentación con puente convencional de baja resistencia y Alimentación con limitación de corriente. En la tabla 2 se resumen las principales características de los sistemas de alimentación de corriente continua. [3]

Sistema	Tensión de Alimentación en Reposo	Corriente Max. En Condición de Reposo	Corriente en Condición de Línea Tomada
Alimentación con Puente Convencional de Baja Resistencia	Valor nominal: 48V Fluctuaciones: 42 a 57V	1mA	$I = \frac{V}{R}$
Alimentación con Limitación de Corriente	Valor nominal: 48V Fluctuaciones: 42 a 57V	1mA	Limitada máx: $36mA < I < 50mA$

Tabla 2: Sistemas de Alimentación de Corriente Continua. [3]

Criterios para determinar el estado de la línea:

Interpretación de Aperturas y Cierres de Bucle:

Las corrientes de bucle para el descuelgue y cuelgue de la línea se fijan con los siguientes criterios:

- En bucle abierto, la red no interpreta un cierre de bucle mientras la corriente sea inferior a 6 mA.
- En bucle abierto, la red interpreta un cierre de bucle cuando la corriente sea mayor a 15 mA.
- En bucle cerrado, la red no interpreta una apertura de bucle mientras la corriente sea mayor a 15 mA.
- En bucle cerrado, la red interpreta una apertura de bucle si la corriente descende por debajo de 6 mA.

Interpretación de los Estados de Cuelgue y Descuelgue:

Los estados de cuelgue y descuelgue están determinados por los siguientes criterios:

- Si la línea no está siendo llamada, la red interpreta todo cierre de bucle, de duración mayor a 80 ms como descuelgue. Los cierres de duración menor a 40 ms nunca son tomados como descuelgue.
- Si la línea está siendo llamada, la red interpreta todo cierre de bucle de duración mayor a 40 ms como descuelgue.
- Si la línea está tomada (descuelgue), la red ignora aperturas de bucle de duración menor a 40 ms. Cuando la apertura de bucle es mayor a 150 ms la red lo interpreta como cuelgue. [3]

Características de Transmisión:

Banda de Frecuencias:

La banda de frecuencias en la línea del abonado está comprendida en el intervalo de 300Hz a 3400Hz. La atenuación en esta banda de frecuencias para el bucle local de usuario es menor o igual a 8dB. [3]

Impedancia de Entrada:

Es la impedancia presentada por la red entre los puntos de Tip y Ring, en condición de línea tomada y es tal que con respecto a 600Ω resistivos el nivel de perdidas en retorno es mayor o igual al que se detalla en la tabla 3. [3]

FRECUENCIAS (Hz)	REQUISITOS (dB)
300 – 500	8 – 10
500 – 1250	10 – 14
1250 – 3400	14

Tabla 3: Niveles de Perdidas en Retorno a 600Ω [3]

Nivel Máximo a la Entrada del Abonado:

Nivel Medio Máximo de Transmisión:

El nivel medio de potencia activa máximo generado a la entrada del abonado desde la red debe ser menor a -10 dB, medidos sobre una resistencia de 600Ω , en cualquier periodo de 10 s.

Nivel de Pico Máximo:

El nivel de pico máximo a la entrada del abonado desde la red debe ser menor a 1,5V, medidos sobre una resistencia de 600Ω . [3]

2.2 El Algoritmo de Goertzel

El Algoritmo de Goertzel es una técnica para el procesamiento digital de señales que puede realizar la detección de tonos y se presenta como una alternativa mucho más eficiente que la Transformada Rápida de Fourier. Mientras que la

transformada rápida de Fourier calcula la componente real e imaginaria para un número N de frecuencias en un ancho de banda, el algoritmo de Goertzel lo hace para una o varias frecuencias específicas. El principal campo de aplicación del algoritmo de Goertzel es la detección de tonos en sistemas de cómputo con baja potencia. [4]

El Algoritmo de Goertzel determina las componentes (real e imaginaria) de las frecuencias como lo haría la Transformada Discreta de Fourier (*DFT*) o la Transformada Rápida de Fourier (*FFT*). Si se necesita, la magnitud y la fase pueden ser calculadas a partir de la pareja real / imaginario.

El Algoritmo de Goertzel Modificado es aún más rápido y simple que el Algoritmo de Goertzel básico pero no determina los componentes de frecuencia real e imaginario. Por el contrario, entrega la magnitud relativa al cuadrado. Se puede tomar la raíz cuadrada de este resultado para obtener la magnitud relativa (si es necesario) pero no hay manera de obtener la fase. Para el presente trabajo, se utilizó el Algoritmo de Goertzel Modificado, por ser el más rápido y porque se puede determinar la presencia de una frecuencia en la señal por su magnitud relativa.

Para detectar una frecuencia, primero es transformada en un coeficiente de la Transformada Discreta de Fourier (*DFT*). La relación entre la frecuencia a detectar (*f_i*) y el coeficiente *DTF* (*k*) está dado por la siguiente ecuación:

$$k = N * \frac{f_i}{f_s}$$

Ecuación 1 [4]

Donde:

N = numero de muestras

f_s = Frecuencia de muestreo

f_i = Frecuencia buscada

Nótese que k es el entero más cercano al resultado de la ecuación 1. Para cada valor de k , la variable de estado, $v_k(n)$, se obtiene a partir de la siguiente ecuación en diferencia recursiva 2:

$$v_k(n) = 2 \times \cos\left(\frac{2 \times \pi \times k}{N}\right) \times v_k(n-1) - v_k(n-2) + x(n)$$

Ecuación 2 [4]

Donde:

$$n = 0, 1, \dots, N$$

Para un mismo valor de k , la ecuación 2 itera hasta que la última variable de estado, $v_k(N)$, es obtenida. A partir de entonces la salida, $y_k(N)$, se obtiene de la ecuación 3:

$$y_k(N) = v_k(N) - W_N^k \times v_k(N-1)$$

Ecuación 3 [4]

Donde:

$$W_N^k = \exp(-2 \times \pi \times k/N)$$

Este es el valor deseado de la DFT, que es, $X(k) = y_k(N)$. Las ecuaciones 2 y 3 son descritas en la realización de forma directa mostrada en la figura 2. Esta figura da un resumen del Algoritmo de Goertzel, de modo que la ecuación 3 es calculada una vez que la ecuación 2 ha sido calculada $N+1$ veces. Además, k es constante cuando las ecuaciones 2 y 3 son evaluadas.

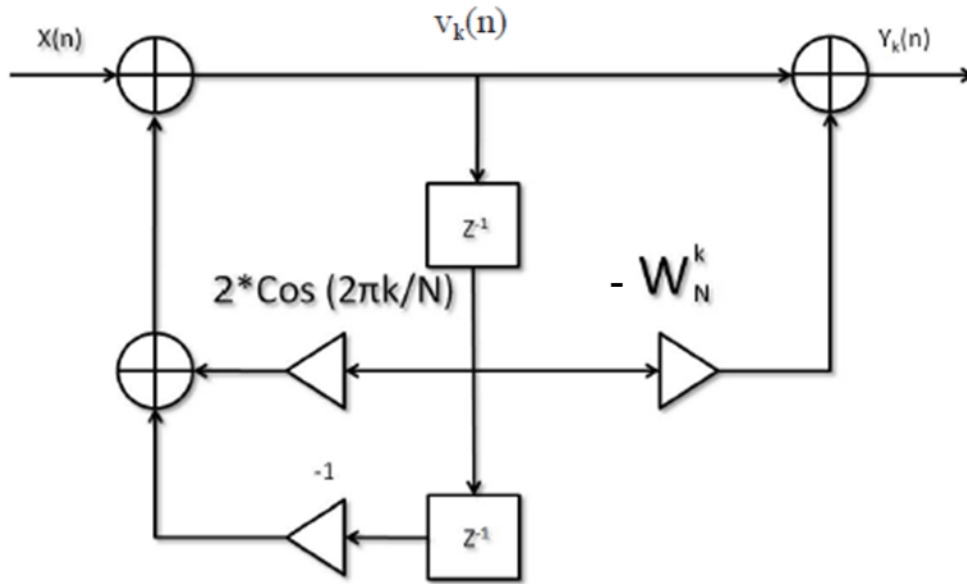


Figura 2: Forma Directa de realización del Algoritmo de Goertzel [4]

El algoritmo de Goertzel se ha modificado aún más basado en el concepto de filtro adaptado para lograr la detección de las frecuencias, en nuestro caso DTMF. La energía de la señal entrante es calculada para las ocho frecuencias DTMF. La frecuencia DTMF en la que la señal de entrada tiene un máximo de energía, es la frecuencia detectada. Esta energía se calcula con la ecuación 4.

$$\text{mag_square} = |X(k)|^2$$

Ecuación 4 [4]

$$\text{max} = \text{maximum} (\text{max. mag_square})$$

Ecuación 5 [4]

En la ecuación 5, *max* es la energía máxima que inicialmente es puesta en cero y guardada en la memoria. La energía de la ecuación 4 es usada para la comparación con la energía máxima almacenada. Cuando una nueva energía es mayor que la energía máxima almacenada para la comparación, esta nueva energía es

guardada como la energía máxima para la siguiente comparación. Así, el valor que inicialmente era cero es cambiado a un número que representa la frecuencia de esta nueva energía. La comparación se realiza para un total de 8 veces. Después de la comparación final, el índice, (un número entre 0 y 7 del resultado de la comparación), representa la frecuencia DTMF detectada.

2.3 Modulación Adaptativa Diferencial de Código de Pulsos

La Modulación Adaptativa Diferencial de Código de Pulsos o ADPCM por sus siglas en inglés, (Adaptive Differential Pulse Code Modulation), es un algoritmo que toma ventaja de la gran similitud que existe entre muestras de voz consecutivas, lo que hace que futuras muestras puedan ser predichas. En lugar de codificar una muestra de audio, la ADPCM codifica la diferencia entre la muestra de audio y una muestra predicha. Este método proporciona una mejor compresión reduciendo el número de bytes por muestra sin deteriorar la calidad del audio. La implementación del algoritmo ADPCM utilizado en el presente trabajo proviene de las Prácticas Recomendadas para la Mejora de Compatibilidad en el Audio Digital en Sistemas Multimedia, revisión 3.00, de la desaparecida Asociación Interactiva Multimedia (IMA). El Algoritmo IUT G.721 ADPCM, como es conocido actualmente, usa aritmética de punto flotante y funciones logarítmicas, lo que no es fácil de implementar en microcontroladores. Es por esto que se usa el Algoritmo que hace referencia la IMA, ya que reduce la complejidad matemática de la IUT G.721, simplificando muchas de las ecuaciones y haciendo uso de tablas de ser necesario. [5]

2.3.1 Compresión

En la figura 3 se muestra el diagrama de bloques para la compresión de audio mediante el algoritmo ADPCM.

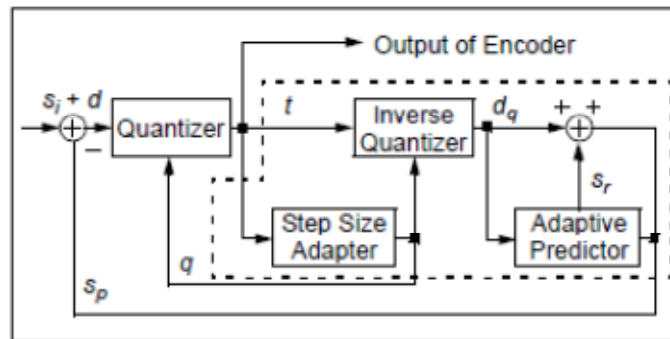


Figura 3: Diagrama de Bloques Compresión ADPCM [5]

La entrada a la rutina del codificador, (S_i), debe ser la muestra de audio de 16 bits en complemento a dos. El rango de valores aceptados para S_i es de 32767 a -32768. La muestra predicha (S_p) y el índice de tamaño del paso de cuantización (q) son guardados en una variable para la próxima iteración. Para la primera iteración, los valores de S_p y q son iguales a cero. La función de codificación toma los 16 bits en complemento a dos de la muestra de audio y entrega un número de 8 bits que contiene los 4 bits en formato signo magnitud del código ADPCM (t).

La muestra predicha S_p es restada de la muestra de audio S_i para producir una diferencia (d). Una cuantización adaptativa es realizada en la diferencia, dando como resultado un número de 4 bits (t). Ambos codificador y decodificador actualizan sus variables internas con este valor ADPCM. Un decodificador completo está integrado al codificador, lo que asegura que el codificador y el decodificador están sincronizados sin necesidad de enviar datos adicionales. El decodificador integrado es el mostrado dentro de la línea punteada de la figura 3. El decodificador integrado toma el valor ADPCM (t) para actualizar el cuantificador inverso, que produce una versión sin cuantificar (d_q) de la diferencia d . La implementación del algoritmo ADPCM para el presente trabajo usa un predictor fijo en lugar de uno adaptativo, lo cual reduce la cantidad de datos y el número de instrucciones para su ejecución.

El adaptador predictivo de la IUT G.721 se ajusta de acuerdo al valor de cada muestra a la entrada, usando el promedio ponderado de los últimos seis valores diferenciales sin cuantificar y los dos últimos valores predichos. Así, en este punto, la diferencia sin cuantificar (dq) es sumada a la muestra predicha (Sp) para producir una nueva muestra predicha (Sr) como se muestra en la figura 3. Finalmente, la nueva muestra predicha (Sr) es guardada en Sp para ser usada en la siguiente iteración. [5]

2.3.2 Descompresión

En la figura 4 se muestra el diagrama de bloques para la descompresión de audio mediante el algoritmo ADPCM.

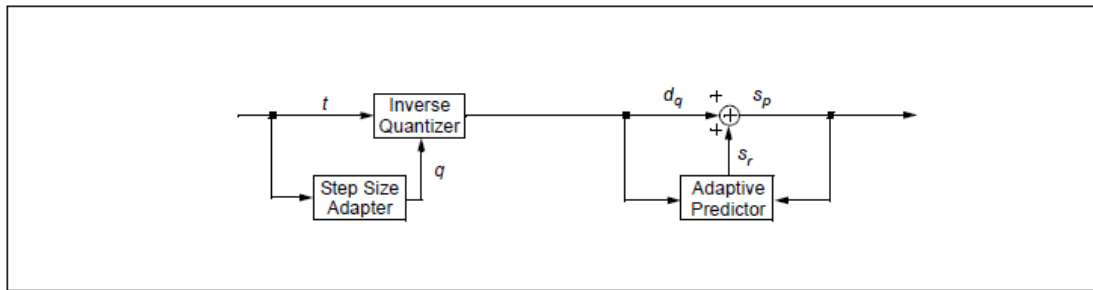


Figura 4: Diagrama de Bloques Descompresión ADPCM [5]

Para la descompresión, a la entrada del decodificador (t) debe estar un número de 8 bits, conteniendo los 4 bits del dato ADPCM en un formato signo magnitud. El rango de valores para t serán de 0 a 15, siendo 7 igual 0x07 y -7 igual a 0x0F. La muestra predicha (Sp) y el índice de tamaño del paso de cuantización (q) son guardados en una variable para la próxima iteración del decodificador. Para la primera iteración, los valores de Sp y q son iguales a cero. Esta función toma el código ADPCM de 4 bits en signo magnitud y entrega la muestra de audio de 16 bits en complemento a dos. Este codificador es el mismo usado en la rutina de codificación, usando el valor de ADPCM el cual actualiza el cuantificador inverso, con lo que produce la diferencia dq . La diferencia dq , es sumada a la muestra predicha, Sp para producir la muestra de salida, Sr . La muestra de salida Sr , es luego

guardada en la muestra predicha, S_p , para la siguiente iteración de el decodificador. [5]

2.3.3 Algoritmo de Referencia ADPCM de la IMA

La IMA, específicamente el Grupo de Trabajo para Tecnicas de Audio Digital, fue una asociación de comercio con representaciones de compañías como Compac, Apple Computers, Crystal Semiconductor, DEC, Hewlett-Packard, Intel, Microsoft, Sony y Texas Instruments. Este grupo trabajó en un estándar que definiera el intercambio de audio de alta calidad entre plataformas computacionales. El algoritmo de Intel DVI, (interface de Video Digital) fue seleccionado como el estándar por su rango dinámico de audio y su baja transferencia de datos. Los formatos de intercambio de audio digital recomendados están dados en la tabla 4. [5]

El algoritmo que se implementó en el presente trabajo deriva del algoritmo de referencia IMA ADPCM. El formato de los datos es de 8.0 kHz, mono, 4 bits ADPCM. Básicamente, la compresión y descompresión usa una cuantificación adaptativa con predicción fija. La cuantificación adaptativa está basada en una tabla de consulta desarrollada por Intel DVI para la IMA.

Frecuencia de muestreo	Mono/Estéreo	Formato	Notas
8,0 kHz	Mono	8-bit m-Law PCM	Estándar CCITT G.711
		8-bit A-Law PCM	Estándar CCITT G.711
		4-bit ADPCM	Algoritmo DVI
11,025 kHz	Mono/Estéreo	8-bit Linear PCM	Macintosh & MP-C
		4-bit ADPCM	Algoritmo DVI
22,05 kHz	Mono/Estéreo	8-bit Linear PCM	Macintosh & MP-C
		4-bit ADPCM	Algoritmo DVI
44,10 kHz	Mono/Estéreo	16-bit Linear PCM	Estándar CD-DA
		4-bit ADPCM	Algoritmo DVI

Tabla 4: Formatos de Intercambio Digital de Datos [5]

CAPÍTULO III

3. ARQUITECTURA DEL HARDWARE DEL SISTEMA

Para el mejor entendimiento del lector el hardware diseñado para esta RTU se puede dividir en cuatro (4) módulos. Separando el circuito general en sub-circuitos para describir cada uno de ellos individualmente, se espera un mejor entendimiento del circuito en su totalidad. Los módulos en los que se dividió el circuito son: un módulo de interfaz con la línea telefónica, uno de entradas/salidas, uno de amplificación y filtro, y uno de control. En la figura 5 se muestra los módulos y su interacción dentro del hardware.

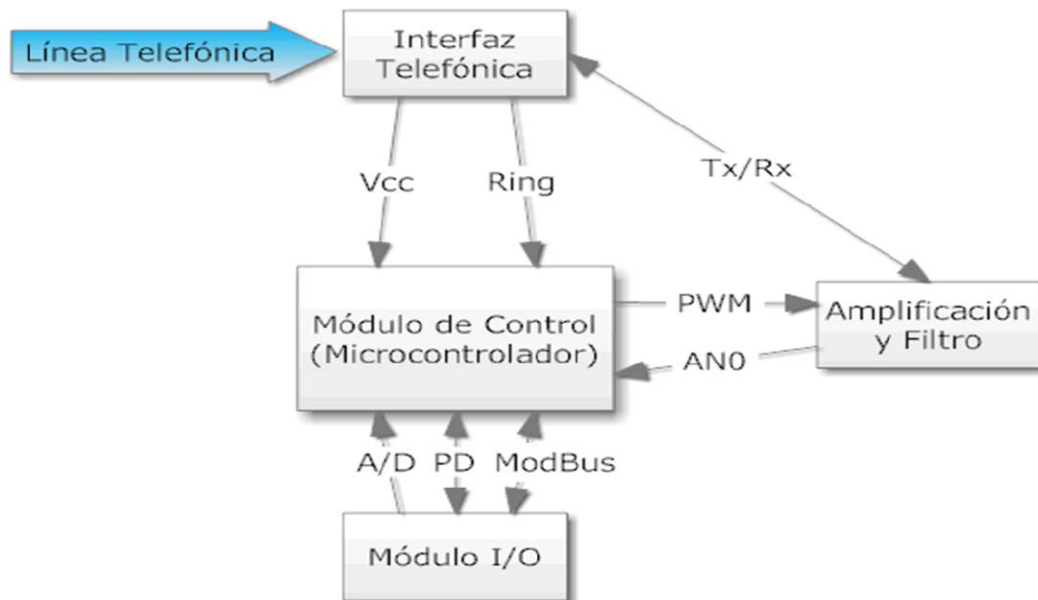


Figura 5: Módulos de la RTU

3.1 Módulo Interfaz con la Línea Telefónica

Este módulo es el que va a estar conectado a la línea de teléfono, por lo tanto, entre sus funciones esta la toma de energía de la red y el envío y recepción de voz y tonos. Se tratará estos dos puntos del módulo por separado, ya que a pesar de estar en el mismo modulo, sus funciones son distintas.

3.1.1 Fuente de Poder

La fuente de poder de la RTU es un elemento fundamental del presente trabajo. Es el encargado de entregar la corriente a todo el circuito, además de ejecutar el cuelgue y descuelgue de la línea aprovechando las interpretaciones que hace la red en las corrientes que demanda el abonado.

Para el diseño de la fuente de poder, se tuvieron presentes todas las características necesarias para su correcto funcionamiento al ser conectado a la red telefónica, esto es: soportar los voltajes nominales de la línea, manejar las corrientes necesarias para el descuelgue y no exceder la corriente que la red puede tolerar antes de interpretar una señal de descuelgue por parte del abonado. En definitiva, la fuente debería tener las siguientes características:

- Soportar el voltaje máximo que la central puede suministrar, (57VDC).
- Para una condición de reposo del microcontrolador, la corriente consumida por la fuente debe ser tal, que junto a la corriente consumida por el microcontrolador, no debe exceder la corriente máxima en condición de reposo de la red, 1mA.
- Para una condición de máximo funcionamiento del microcontrolador, la fuente debe poder suministrar la corriente que demande el microcontrolador, y los demás módulos, que no debe exceder el valor máximo que suministra la red en condición de línea tomada en un sistema de alimentación con limitación de corriente, 50mA.
- Debe entregar un voltaje de 3,3VDC para el correcto funcionamiento del microcontrolador y los demás componentes.

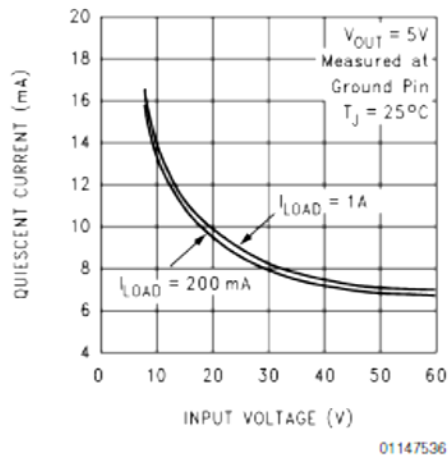
- Además, debe soportar la señal de llamada (repique), de magnitud 75Vrms en el período que dure, antes que el RTU conteste.

Fuente conmutada:

Como primera opción se consideró el uso de una fuente conmutada, por ser éstas las que mayor eficiencia tienen en la entrega de energía. Sin embargo, a la hora de implementarla resultó tener serios problemas para cumplir con las características de diseño de la fuente de poder en lo que respecta a soportar el voltaje de la línea de 57V y no consumir más de 1mA de la línea.

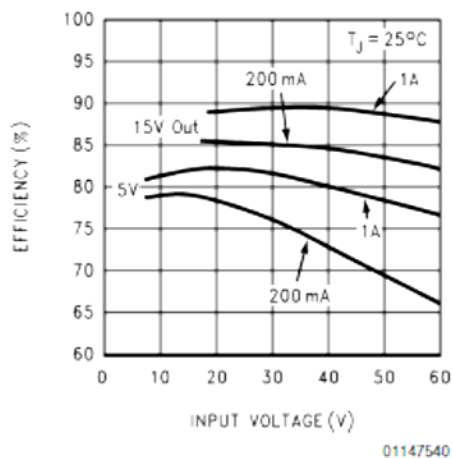
Se tomaron en consideración varios integrados de diferentes fabricantes para realizar la conmutación de la fuente, teniendo como principales opciones el MC34063A, del fabricante Texas Instruments por ser de bajo costo, alta eficiencia, (cercana al 90%), y disponibilidad; y el LM2575HV en su versión ajustable, de National Semiconductor, por ser el único integrado de conmutación disponible en el país que puede soportar los 57V de la línea (60V).

El LM2575HV se consideró por la característica que podía ser conectado directamente al voltaje de la línea, sin necesidad de un diodo zener o un divisor de voltaje, lo que a la final tendría como consecuencia un consumo adicional de corriente y un costo adicional de la RTU. Pero al estudiar la hoja de datos del dispositivo, se encontró con que el mismo tiene una corriente de reposo muy alta para este proyecto, de alrededor de 10mA, lo que significa que en condición de bajo consumo del microcontrolador, la fuente sola estaría consumiendo 10mA, lo que es demasiado para la línea en reposo. En la gráfica 1 se puede apreciar la relación entre el voltaje a la entrada, y la corriente de reposo para una salida de 5V. Allí se puede apreciar cómo la diferencia en la corriente de reposo entre 1A y 200mA es muy pequeña, por lo que no se esperarían mejores resultados para corrientes menores, por ejemplo de micro amperes y para un voltaje a la entrada de 60V, la corriente de reposo tiende a ser de entre 6 y 6,5A. [6]



Gráfica 1: Corrientes en Reposo LM2575HV [6]

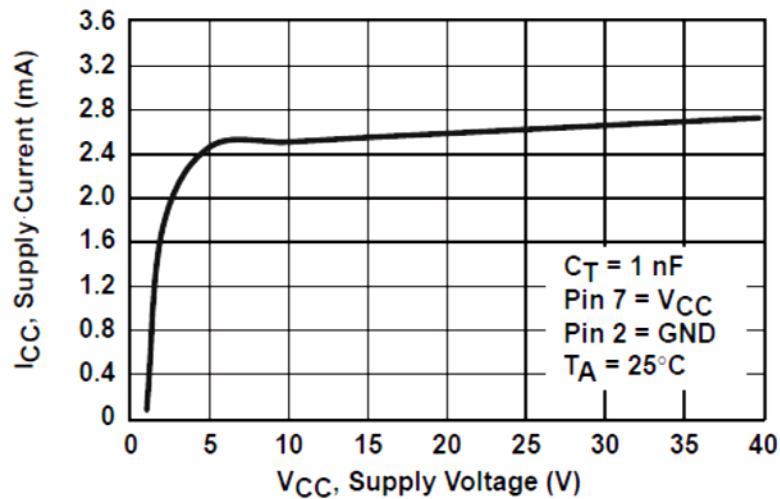
En la gráfica 2 se muestra cómo la eficiencia del LM7525HV es mayor, mientras mayor es el voltaje y corriente que entrega. Nótese que para una salida de 5V y 200mA, con una entrada de 60V, la eficiencia cae hasta casi un 65%.



Gráfica 2: Eficiencia LM2575HV [6]

De lo anterior se llegó a la conclusión de que el LM2575HV no podría nunca llegar a ser implementado para tomar corriente de la línea telefónica, a menos que la línea este en estado de descuelgue, y aun así, la mitad de la corriente suministrada por la central sería para alimentar la fuente.

El otro integrado tomado en consideración para hacer una fuente conmutada alimentado de la línea de teléfono fue el MC34063A, que a pesar de no poder soportar los 57V que puede llegar a dar la central telefónica a la entrada, tiene una eficiencia mayor que el LM2575HV y una corriente de reposo menor, 2,5mA. En la gráfica 3 se muestra la relación entre el voltaje a la entrada y la corriente que consume el MC34063A, se puede notar como el integrado consume casi la misma corriente para todo el rango de valores de voltaje que soporta a la entrada.



Gráfica 3: Corriente Consumida por el MC34063A [7]

Al igual que en el LM2575HV, esa es sólo la corriente que consume el integrado, que ya está excediendo el valor que se estableció para la corriente en condición de reposo de 1mA. Además, hace falta un circuito que disminuya el voltaje que entrega la central telefónica para que no sea mayor a los 40V que pueden ser colocados para alimentar el MC34063A.

En este punto se hizo evidente que el uso de una fuente conmutada no era la mejor opción, así que realizando una investigación de métodos de suministro de energía de bajo consumo, en especial en los dispositivos que se alimentan de baterías, se llegó a la conclusión que la mejor alternativa disponible en el mercado eran los LDO.

LDO o Regulador de Baja Pérdida:

Los LDO por sus siglas en inglés, son dispositivos de regulación de voltaje que están siendo adoptados por muchas aplicaciones en la actualidad, gracias a su bajo consumo de potencia y su bajo costo. Son comúnmente usados para regular voltaje en circuitos que están diseñados para funcionar a baterías, por lo que el diseño está orientado al mínimo consumo de energía para así garantizar la máxima durabilidad de las baterías. Por estas características el uso de LDO resultaron ser la mejor opción para el diseño de la fuente.

Al igual que con la fuente conmutada, se estudiaron muchos integrados para ser implementados, tomando en cuenta los parámetros ya establecidos para el diseño de la fuente. Los parámetros que privaron en la decisión del integrado fueron a la final, el de voltaje de 57V y corriente de 50mA, ya que los demás parámetros los LDO lo cubren en su mayoría, salvo el de la señal de repique, el cual siempre amerita una consideración especial. Dado que a mayor el rango de voltajes y corriente de salida, el precio del LDO sube significativamente, la corriente de salida máxima del LDO elegido no debería exceder los 50mA, que es lo máximo que tiene que entregar la fuente, y el voltaje de entrada debería ser el suficiente para operar con los 57Vdc que suministra la central telefónica, y los 75Vrms de la señal de repique. El integrado elegido en definitiva fue el LT3010 en su versión ajustable del fabricante Linear Technology, por su rango de voltajes a la entrada, de 3 a 80V, su corriente máxima a la salida de 50mA, y su baja corriente de reposo de 30 μ A. Este integrado tuvo que ser adquirido en el exterior, ya que en el país no hay mucha disponibilidad de LDOs. En

la figura 6 se muestra una configuración típica para el LT3010 en su versión ajustable. [8]

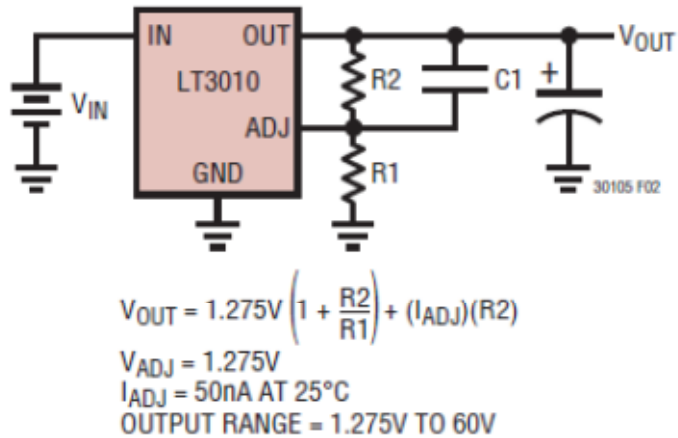


Figura 6: Configuración Típica para el LT3010 [8]

3.1.2 Diseño de la fuente e interfaz con la línea

Una vez seleccionado el integrado para la regulación del voltaje, el siguiente paso es el diseño de la fuente y de la interfaz con la línea de teléfono. En la figura 7 se puede apreciar el esquema del módulo de interfaz con la línea de teléfonos completo.

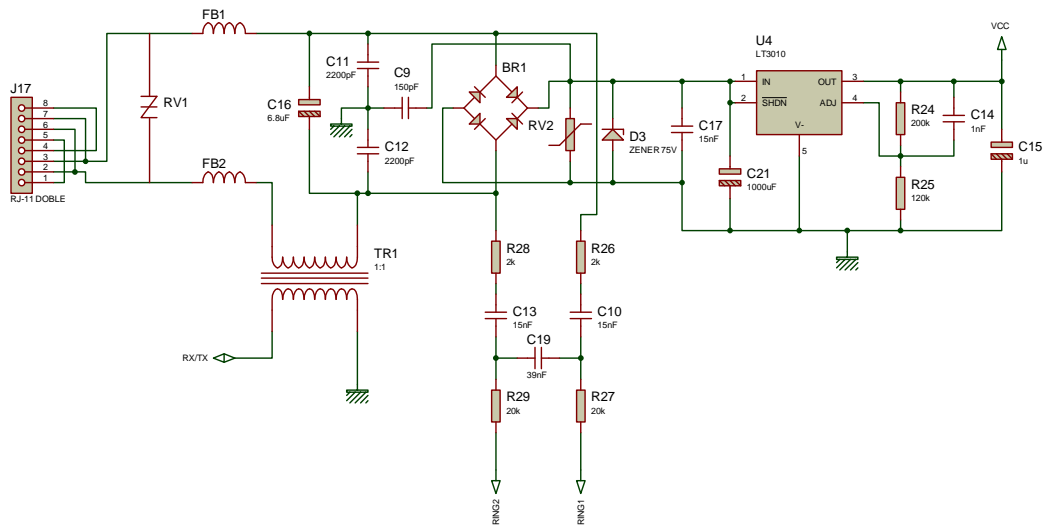


Figura 7: Módulo de interfaz con la línea de teléfono.

Para el diseño de la interfaz se estudiaron los diseños existentes para los fax-modem, por ser dispositivos electrónicos conectados directamente a la línea de teléfono con muchos años de desarrollo en circuitos de protección para los componentes de las sobretensiones y niveles de ruido que puedan presentarse en la línea, además de apegarse a las normas existentes por las compañías telefónicas y los estándares internacionales. Así, la mayor parte de la interfaz fue tomada del esquemático típico de aplicación para los integrados de acceso directo SI3035 de Silicon Laboratories. [9]

A la entrada de la interfaz se tiene un RJ-11 hembra, protegido por el Diodo de Silicio de Corriente Alterna o SIDAC por sus siglas en ingles RV1. El SIDAC es un dispositivo bilateral de disparo de alta tensión y corriente, similar a un tiristor, cuya función es proteger el circuito de sobretensiones en la línea. Los núcleos de ferrita FB1 y FB2 junto con los condensadores C11 y C12 están dispuestos para reducir las emisiones electromagnéticas. El puente de diodos fue colocado para asegurar que la polaridad en el LDO siempre sea la correcta, indiferentemente de la polaridad que tenga el conector RJ-11 macho a ser insertado en el dispositivo.

Luego tenemos RV2 que es un Varistor de Metal-Oxido o MOV por sus siglas en ingles, para protección adicional sobre el LDO. El circuito conformado por R26, R27, R28, R29, C10, C13 y C19 ayuda a disminuir y dejar pasar sólo la señal de repique, dando como salida la señal RING que será introducida en el microcontrolador para detectar una llamada entrante.

El transformador TR1 es un transformador 1:1 de impedancia 600Ω para acoplar la señal de audio de la línea al modulo de amplificación. Está conectado en serie a la línea telefónica para aprovechar la corriente consumida por el LDO, en la transmisión y recepción de audio y el condensador C16 ofrece un camino para el retorno de la señal AC de sonido, sin afectar la señal DC que va al LDO.

Por último se tiene la fuente de poder del RTU, que tiene a la entrada un diodo zener de 75V. Dado que la tensión de la señal de llamada es de $75V_{rms}$ y ésta llega al abonado sobre el voltaje DC de 57V y la tensión de la señal de llamada (repique) es rectificadas por el puente de diodo BR1, con todo lo anterior, la tensión a la entrada del LDO puede llegar a ser de hasta 95Vdc, por lo que hace falta un diodo zener para limitar el voltaje que no debe exceder los 80V que soporta el LDO. Con un zener de 75V, se limita la tensión a la entrada del LDO para protegerlo y con un umbral por encima de los 57V de la línea, este no consume corriente adicional de la línea. El condensador C21 tiene como función suministrar la corriente que demanda el microcontrolador cuando cambia de su condición de reposo a la de total funcionamiento. Como la línea de teléfono se opone a los cambios bruscos de energía y el cambio de oscilador del microcontrolador se produce en microsegundos, el condensador C21 permite que el cambio se concrete, dándole a la línea telefónica oportunidad de cambiar del modo de línea en reposo al de línea tomada. Sin este condensador, el microcontrolador no encuentra la corriente necesaria para cambiar de oscilador, lo que produce que este se reinicie y nunca logre cambiar de oscilador. Las resistencias R24 y R25 obedecen a la ecuación 6 para obtener a la salida un voltaje de 3,41V. Los valores para R24 y R25 se tomaron considerando los mayores permitidos

por el fabricante del LDO para obtener un valor cercano a los 3,3V para que la corriente que circula por las resistencias sea la mínima.

$$V_{OUT} = 1.275V \left(1 + \frac{R2}{R1} \right) + (I_{ADJ})(R2)$$

Ecuación 6 [8]

Donde:

$R2=R24=200k\Omega$

$R1=R25=120k\Omega$

$I_{adj}=50nA$

3.2 Módulo de Amplificación y Filtro

En la figura 8 se muestra el esquema del módulo de amplificación y filtro.

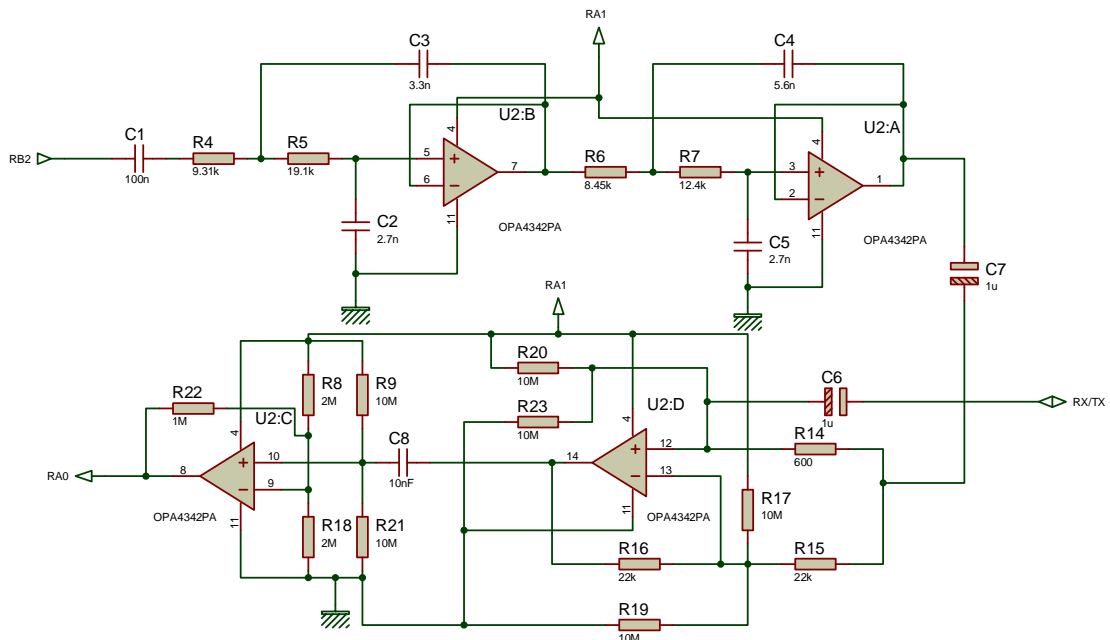


Figura 8: Módulo de Amplificación y Filtro

Como se puede apreciar el módulo está compuesto de cuatro amplificadores operacionales, dos para la transmisión de la señal y dos para la recepción. Los amplificadores operacionales U2A y U2B forman un filtro activo Butterworth pasa bajos de cuarto orden, el operacional U2D es para acoplamiento de la señal con la línea y U2C es para amplificar la señal que llega de la línea. Los cuatro operacionales pueden estar en un solo encapsulado o en varios. Para este trabajo se utilizó el integrado OPA4342 que trae los cuatro amplificadores operacionales que se necesitan. En la figura 9 se muestra el esquema interno del OPA4342 y la configuración de sus pines.

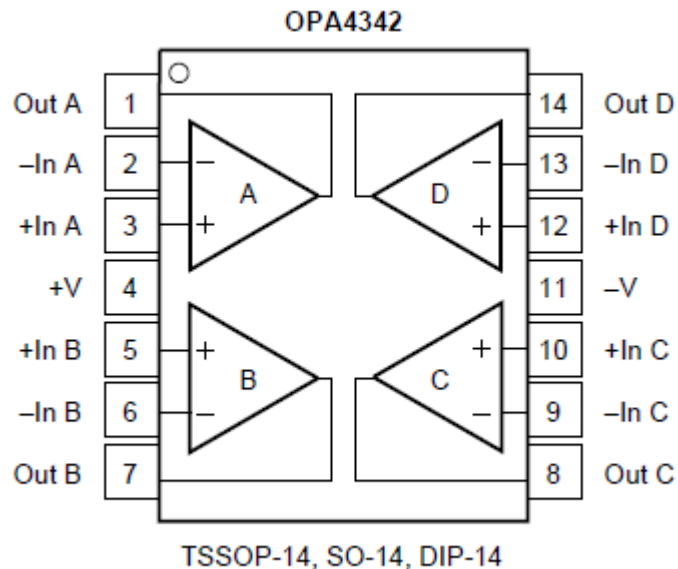


Figura 9: Amplificador Operacional OPA4342 [10]

El OPA4342 es un integrado con cuatro amplificadores operacionales, una corriente de reposo de 150uA, se puede alimentar con una fuente simple de 2,7V como mínimo, por lo que no necesita un voltaje negativo para su polarización, un ancho de banda de 1MHz, y tiene la capacidad “rail to rail” tanto a la entrada como a la salida. Esta última característica es muy importante ya que el voltaje de alimentación para el OPA4342 es de 0V a 3,3V, y una salida “rail to rail” aprovecha al máximo el voltaje que alimenta el integrado. Por ejemplo, el LM324 puede

entregar como máximo a la salida (V+)-1,5V, lo que significa que sólo puede dar como máximo 1,5V menos que el voltaje con que está alimentado, con lo que ya se estaría perdiendo casi la mitad de la señal de 3,3V, mientras el OPA4342 puede llegar a voltajes por debajo de 100mV de su tensión de alimentación. [10]

Otro aspecto importante que se puede notar en la figura 8 es la alimentación del módulo, que no proviene directamente de Vcc sino de RA1, que es el pin 1 del puerto A del microcontrolador. La razón de esto es para disminuir el consumo de corriente de la central telefónica llevando el pin a 0 cuando no se esté usando el OPA4342, como por ejemplo cuando la línea está en condición de reposo. El microcontrolador puede entregar por cada pin de salida 25mA y el OPA4342 puede consumir como máximo 15mA (corriente de corto circuito), por lo que el riesgo de que el puerto falle por exceso de corriente es muy pequeño. El microcontrolador es el único alimentado directamente por el regulador, el resto de los circuitos en los módulos de entrada y salida y de amplificación y filtro son alimentados por el pin RA1.

Filtro Butterworth pasa bajos de cuarto orden:

El filtro Butterworth pasa bajos de cuarto orden está conformado por amplificadores operacionales U2A y U2B, las resistencias R4 a R7 y los condensadores C2 a C5. La señal a la entrada del filtro proviene del pin RB2 del microcontrolador, el cual es una salida PWM con la señal de audio. Esta señal pasa por el filtro Butterworth pasa bajos de cuarto orden, el cual tiene una frecuencia de corte de 4 kHz, y la convierte en una señal de voltaje analógica para ser transmitida. El condensador C1 sirve para filtrar la componente DC y las frecuencias por debajo de 300 Hz.

Acoplamiento:

El operacional U2D tiene como función el acoplamiento de la línea de teléfono con el módulo. De nuevo observando la figura 8 se puede ver cómo la señal RX/TX proveniente del transformador TR1 del módulo de interfaz con la línea ingresa a través del condensador C6 a la entrada positiva del operacional U2D. A través de este punto se realiza tanto la transmisión como la recepción de las señales. El circuito funciona como un puente de Wheatstone a través de las resistencias R14, R15 y R16; para mantener las entradas del operacional en equilibrio, las resistencias R15 y R16 tienen un valor de $22\text{k}\Omega$ cada una para formar un equilibrio en la entrada negativa, R14 es de 600Ω que junto con la impedancia del transformador forma el equilibrio para la entrada positiva del operacional. Las resistencias R17, R19, R20 y R23 están dispuestas para mantener un pedestal de $V_{cc}/2$ a cada una de las entradas. Así, si una señal AC ingresa al circuito a través de C6 estará variando el voltaje que ve ese punto con respecto a tierra, y por el principio de corto virtual del operacional, la variación de voltaje se verá reflejado en la entrada negativa del mismo y por medio de la resistencia R16 llegará a la salida del operacional. Para la transmisión, la señal de voltaje analógica proveniente del filtro Butterworth pasa bajo ingresa a través de C7 variando el voltaje en las dos entradas del operacional por igual, por lo que la señal no tiene retorno sin embargo, al variar la entrada positiva estará variando el voltaje aplicado al transformador TR1, por lo que estará transmitiendo la señal por la línea de teléfono. Los condensadores C6 y C7 anulan las componentes de voltaje DC que pudieran presentarse en las entradas del operacional, mientras que las resistencias R17, R19, R20 y R23 le dan un componente DC de la mitad de V_{cc} para no tener pérdidas de la señal por cruces por cero. Los valores de las resistencias R17, R19, R20 y R23 son de $10\text{M}\Omega$ para reducir al mínimo la corriente que consumen.

Amplificación:

El último de los cuatro amplificadores operaciones permite amplificar la señal proveniente de la línea que puede tener valores máximos de 1,5Vp, 600mVp típicos. En el operacional U2C se polariza $V_{cc}/2$ en sus entradas al igual que en U2D para evitar pérdidas de la señal y tiene una ganancia de 2. A la salida entrega una señal de alrededor de 1,2Vp sobre un valor DC de 1,5V para que el convertidor analógico digital del microcontrolador en el pin RA0 lo pueda procesar sin perder información.

3.3 Módulo de Entrada/Salida

El modulo de Entrada/salida es el más sencillo de todos y en él se agrupan las distintas entradas y salidas que tendrá la RTU con el sistema a monitorear. El módulo cuenta con dos entradas analógicas, cuatro entradas y salidas digitales y un puerto para la comunicación MODBUS. En la figura 10 se muestra el esquema para el módulo.

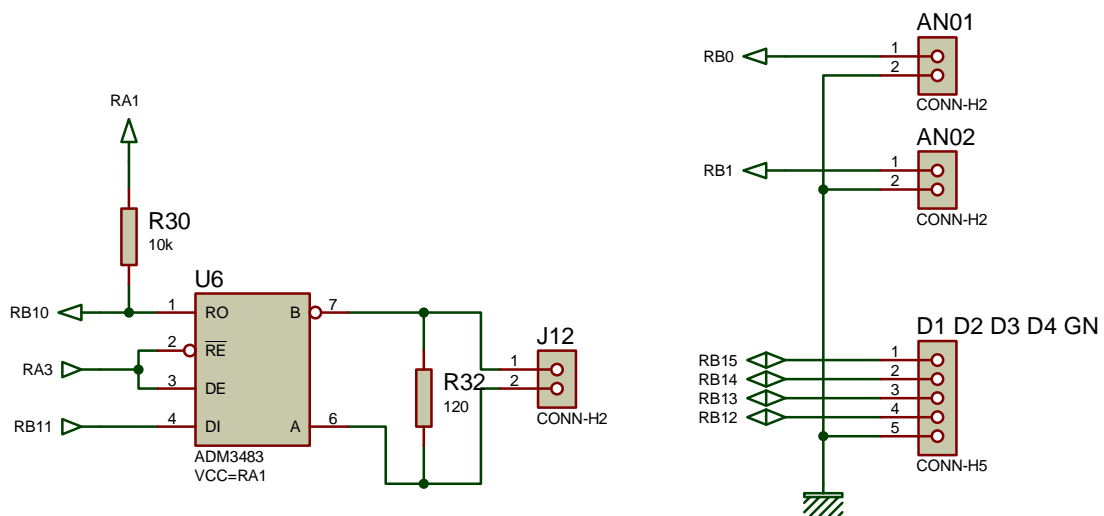


Figura 10: Módulo de Entrada/Salida

Como se puede apreciar, las entradas analógicas están conectadas directamente al microcontrolador a sus pines RB0 y RB1, además cada entrada tiene una conexión a tierra. De igual forma, los puertos digitales están conectados directamente a los pines del microcontrolador RB12 al RB15, los cuatro pines más significativos del puerto B del microcontrolador. Cualquier conexión que se haga a los puertos analógicos hay que tener en consideración que los valores aceptados en el son de 0 a 3,3V, por lo que cualquier rango de voltajes superior a este deberá pasar primero por un divisor de voltaje. Para los puertos digitales que funcionen como entrada hay que tener en consideración el valor máximo de voltaje que soportan los pines de 3,3V. Si funcionan como salida, tienen que ir a la base de un transistor BJT o a el gate de un MOSFET alimentado de manera independiente, ya que cualquier demanda del corriente del pin podría ser interpretado por la central como un cierre de bucle. Se recomienda utilizar un arreglo Darlington para esta tarea.

La comunicación MODBUS se hará por medio de un bus RS-485 y para ello hace falta un integrado que lleve las señales del UART del microcontrolador a las utilizadas en dicho protocolo. El integrado a usar es el ADM3483 del fabricante Analog Devices, que es un componente importado ya que a pesar de ser el bus RS-485 muy implementado los integrados que se consiguen en el país son para un voltaje de alimentación de 5V, y para el presente trabajo se pretende trabajar con 3,3V y el ADM3483 puede trabajar con un voltaje de 3,0V a 3,6V, además de tener un bajo consumo de corriente, de tan solo 2,2mA como máximo. Este integrado también está alimentado por el pin RA1 para ahorro de energía. La resistencia R30 está dispuesta para impedir errores cuando se transmite, ya que el integrado pone RO en alta impedancia lo que puede ser interpretado por el microcontrolador como un valor bajo e iniciar la rutina de recepción. La resistencia R32 es para acoplamiento de la línea y es opcional. En la figura 11 se muestra el diagrama del ADM3483.

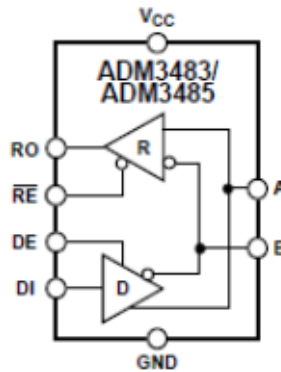


Figura 11: ADM3483 [11]

3.4 Módulo de Control

El módulo de control es el que contiene el microcontrolador y sus periféricos, que incluyen tres botones, tres leds, un conector para programación y el condensador VCap. En la figura 12 se muestra el esquema del módulo de control.

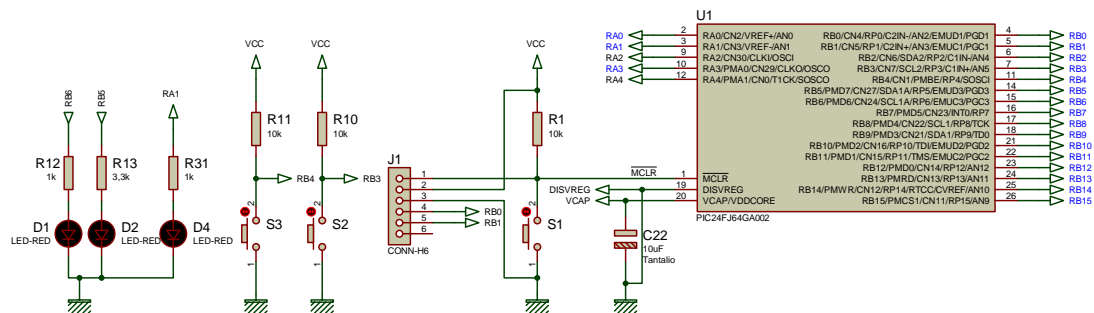


Figura 12: Módulo de Control

De los leds, uno es testigo de encendido del dispositivo, otro indica la presencia del tono de 425Hz en la línea y el tercero indica que la línea está tomada. De los botones, uno es para el reinicio del dispositivo, otro para descolgar y colgar la línea y el último es para pruebas del dispositivo. El Led D2 es el testigo de encendido y permanecerá encendido siempre que el dispositivo funcione correctamente, está conectado al pin RB5 el cual oscila para disminuir el consumo del led en estado de

reposo de la línea y esta fijo en estado de línea tomada. La resistencia R13 tiene un valor de $3,3k\Omega$ para disminuir la corriente que consume D2 y este led es de luz roja que es visible en baja corriente. El led D1 es el testigo de presencia del tono de 425Hz en la línea, es muy útil para saber si la central respondió al descuelgue con una invitación a marcar, así como las otras señalizaciones a 425Hz, (llamada, ocupado, congestión), su resistencia de $1k\Omega$ limita la corriente que consume sin impedir que encienda, este led es de luz verde y está conectado al pin RB6 del microcontrolador. El led D3 indica que el microcontrolador ha cambiado la frecuencia del oscilador y está en estado de activo, está conectado al pin RA1 que es el encargado de dar energía a los módulos de entrada/salida y amplificación y filtro, que sólo se activan cuando el dispositivo esta activo, tiene una resistencia de $1k\Omega$ y es de luz verde. El botón S1 es el de reinicio del microcontrolador, el botón S2 es para pruebas de descuelgue fuera de línea, lo que elimina las rutinas de verificación de la línea, y el botón S3 es para descolgar y hacer una llamada de forma automática al número grabado en el dispositivo. Todos los botones están conectado a Vcc por medio de una resistencia de $10k\Omega$ por lo que son pulsadores normalmente puestos en uno lógico y se activan con cero lógico.

El condensador conectado al pin VCAP del microcontrolador es un condensador de tantalio de $10\mu F$, el cual es muy importante para el consumo de corriente del microcontrolador. El pin VCAP es la entrada del voltaje del núcleo del microcontrolador, este voltaje tiene que permanecer estable para que el microcontrolador no consuma más corriente de la debida para mantener un voltaje del núcleo estable, por eso el condensador conectado a VCAP es muy importante, estabilizando el voltaje del microcontrolador a alrededor de $2,2V$ que es lo nominal para el núcleo del microcontrolador. Otra técnica para disminuir aun más el consumo del microcontrolador, es colocar un LDO de $2,2V$ alimentando el pin VCAP, esta opción no se consideró por el costo que representaría el uso de un LDO adicional y que la corriente con el condensador de tantalio se mantiene dentro de los parámetros aceptados por la central telefónica.

Microcontrolador:

El microcontrolador es el encargado de todas las operaciones en el RTU, mientras en otros diseños se recurre al uso de integrados adicionales para funciones tales como la detección DTMF o la generación de audio. En este trabajo el microcontrolador desempeña esas funciones para disminuir el costo usando menos ICs dedicados. El microcontrolador usado para el RTU es el PIC24FJ64GA002 del fabricante Microchip Technology Inc. que es un microcontrolador con arquitectura de 16bit, de 28 pines con una velocidad máxima del oscilador interno de 32MHz, 64kbytes de memoria de programa y manejo de nanowatts power.

En la figura 12 se puede observar al microcontrolador bajo la identificación de U1. Los pines en azul indican los que se están usando en el RTU, y los que están en negro los que no se usaron. En la figura 13 se muestra la disposición de pines del PIC24FJ64GA002.

28-Pin SPDIP, SSOP, SOIC

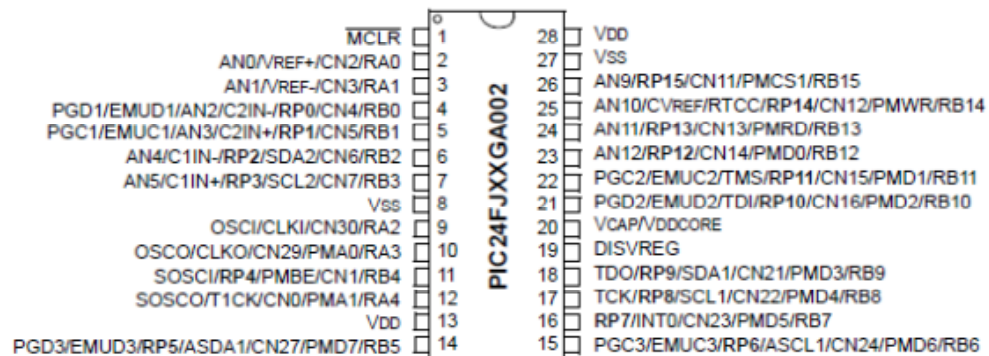


Figura 13: PIC24FJ64GA002 [12]

El PIC24FJ64GA002 tiene muchas características significativas, pero a continuación resaltaremos las que privaron en su elección:

- Oscilador interno de 32 MHz
- Arquitectura optimizada para compiladores en lenguaje C
- Voltaje de alimentación entre 2V a 3,6V
- Manejo de energía por medio de cambios en el oscilador interno
- Entradas analógicas de 10 bits
- Cinco Timers de 16 bits
- Cinco PWM de 16 bits
- Dos UARTs para comunicación serial

El resto de las características se pueden consultar en la hoja de datos del PIC24 [12] en la página del fabricante www.microchip.com.

CAPÍTULO IV

4. ARQUITECTURA DEL SOFTWARE DEL SISTEMA

4.1 Descripción general

El software o firmware del RTU va a ser el encargado de realizar por medio del hardware todas las funciones necesarias para cada una de las aplicaciones del dispositivo. El lenguaje de programación usado es el ANSI C, implementado por medio del MPLAB IDE v8.63 con el toolsuite Microchip C30 v3.23 en su versión estudiantil, el cual es gratis y de descarga directa por la pagina de michochip.

El software debe tomar el valor del analógico digital y procesarlo con el algoritmo de Goertzel para obtener los valores de las frecuencias de tonos DTMF para luego aplicar una rutina de validación y devolver un digito DTMF, con la ayuda del Microchip File System o MPFS debe tomar los valores de una tabla previamente almacenada con las palabras que se desean transmitir para luego aplicar el algoritmo ADPCM de descompresión y sacar el resultado por PWM. Además tiene que monitorear las entradas, realizar rutinas para descolgar, colgar, responder, contestar y remarcar, entre otras aplicaciones que debe realizar el microcontrolador y que el software indicara como proceder.

4.2 Rutina de detección y validación de los tonos multifrecuencia

Esta parte del software usa el algoritmo de Groetzel para detectar los tonos DTMF y luego una rutina para validar los tonos con los criterios de la recomendaciones de la IUT-T Q.24 [13]. También detecta el tono de 425Hz usado por las centrales telefónicas para la señalización.

Como se vio en el capítulo anterior, el módulo de amplificación y filtro entrega a la entrada del analógico digital AN0 la señal proveniente de la línea telefónica que devolverá el valor de la señal al registro ADC1BUF0. Este valor es tomado con la interrupción del Timer 5 que se configura a un valor de 8kHz, valor recomendado para tomar muestras en sistemas de telefonía. Así cada 125 μ s se toma una muestra del AN0 y es almacenada en un arreglo de nombre “*sample*”, esto se repite hasta que se toman el número de muestras especificado en la constante “*GOERTZEL_N*”, que está fijado en 180 muestras. Una vez obtenidas todas las muestras, se deshabilita la interrupción del Timer 5 y se procede al cálculo del algoritmo de Goertzel. Los valores de los primeros coeficientes de la serie $V_k(n)$ para cada uno de los tonos DTMF y el tono de 425Hz se calculan al iniciar el programa y son almacenados en un arreglo de nombre *coeffs[n]* ya que estos valores son constantes. Se calculan todas las muestras para cada uno de los valores de las frecuencias a detectar y a la final se obtienen valores estimados de energía para cada una de las frecuencias y se almacenan en el arreglo *magnitude*, se habilita la interrupción para el Timer 5 para empezar a recoger muestras nuevamente y procede a validar los tonos. En la figura 14 se muestra el diagrama de flujo para la rutina del cálculo de las energías de las 8 frecuencias a muestrear.

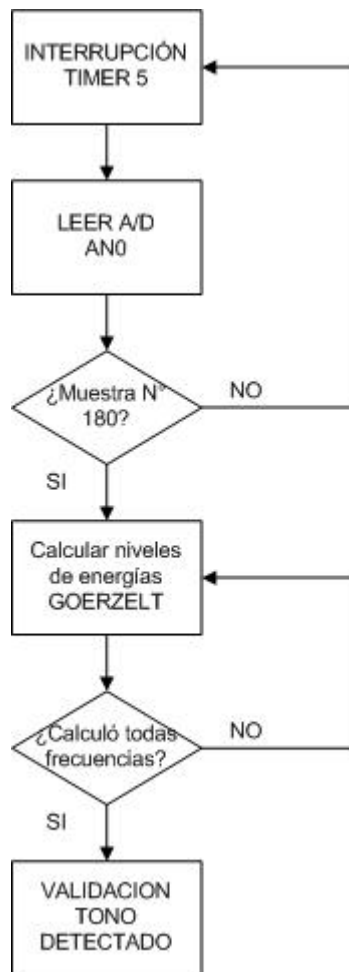


Figura 14: Diagrama de flujo detección de tonos

La validación se lleva a cabo en varios procesos distintos. En el primero se comparan los valores del arreglo *magnitude* con el valor de la constante *threshold*, la constante *threshold* es un valor esperado para la energía de la frecuencia a detectar, por lo tanto los valores en *magnitude* que sean mayor a *threshold* representan una frecuencia detectada. Los valores de *magnitude* tienen el mismo orden de las frecuencias en el arreglo *freqs*, cuyo orden es 697, 770, 852, 941, 1209, 1336, 1477 y 425. Por lo que si el valor en *magnitude*[7] es mayor a *threshold* (en este caso 1×10^6) significa que se ha detectado un tono de 425Hz. Aquí se toman caminos distintos de validación para los tonos DTMF y para los 425Hz de señalización.

El tono de 425Hz se detecta solo en el proceso de descuelgue para interpretar las señalizaciones de la central, por lo que se cuenta el tiempo en que se detecta el tono de 425Hz y el tiempo en que no hay tono o de silencio. Lo primero que se hace es esperar un tono de invitación a marcar, por un tiempo de aproximadamente 3 segundos, si en este tiempo no se detecta un tono de 425Hz el dispositivo cuelga, espera por 10 segundo y vuelve a marcar, a este ultimo procedimiento se le ha llamado “remarcar”. Al detectar un tono de 425Hz, se cuenta el tiempo que se esté detectando el tono, si este tiempo es mayor a 2 segundos el programa precede a discar el numero almacenado en el arreglo *numtel*. Luego espera 4 segundos como máximo el tono de llamada, si no detecta un tono de 425Hz en ese periodo el programa procede a remarcar. Si se detecta un tono de 425Hz se procede a contar el tiempo de silencio, si el tiempo de silencio es igual al tiempo del tono la señal de la central es la ocupado o de congestión de la línea, por lo que se procede a remarcar, si el tiempo de silencio es mayor que el tiempo de tono es una señal de llamada por lo que el programa espera a que el tiempo de silencio sea mayor a 4 segundos para contestar y empezar con la rutina de menú del dispositivo. Si al tercer repique no se responde el dispositivo procede a remarcar. La rutina de descuelgue se produce sólo al apretar el pulsador S2 o al producirse una alarma en el dispositivo. En la figura 15 se muestra el diagrama de flujos para la validación del tono de 425Hz.

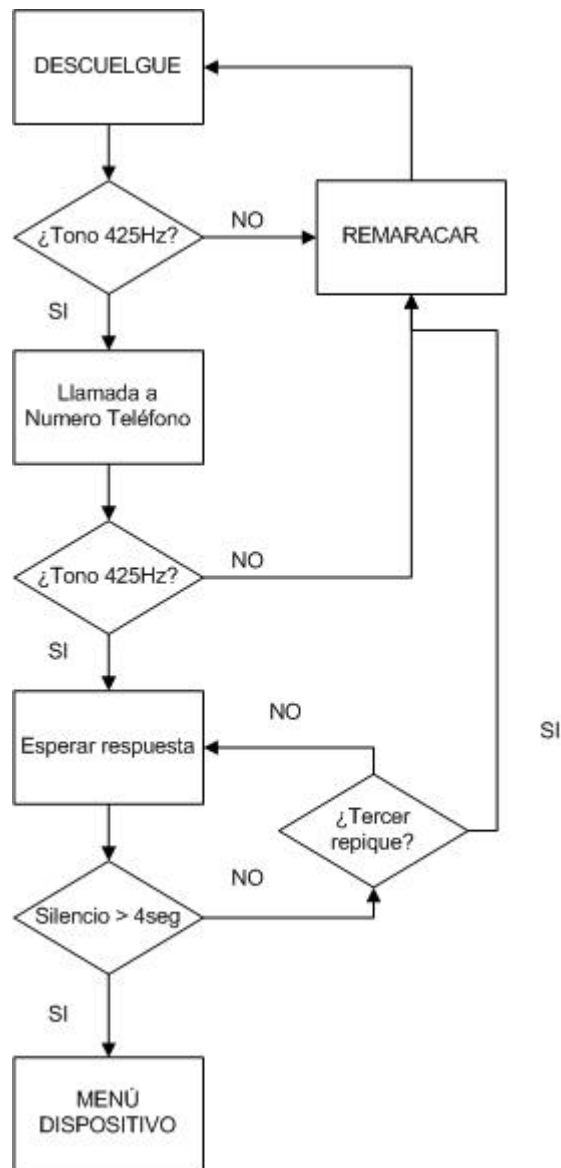


Figura 15: Diagrama de flujos para la validación del tono de 425 Hz

Cuando se entra en la rutina de menú interactivo la detección de tonos se centra en las siete frecuencias DTMF. Con los valores de *magnitude* se hacen dos comparaciones por separado, para el grupo de frecuencias inferiores y para el grupo de frecuencias superiores, en cada grupo se evalúa que el valor de *magnitude* sea mayor que *threshold*, de ser así el valor *magnitude* correspondiente es almacenado en la variable *max*, para las siguientes comparaciones. Si el valor en *magnitude* es mayor a *threshold* se compara con el almacenado en *max*, el mayor de los dos valores será el

nuevo valor de *max* y así hasta que a la final se tendrá el valor máximo de todos los niveles de energía que superaron el valor de *threshold* para el grupo de frecuencias inferiores y superiores. Las dos frecuencias equivalentes serán almacenadas en las variables PreL y PreH para ser comparadas con el siguiente par de frecuencias detectadas. Si son las mismas se toma como un periodo de duración del tono, de lo contrario se considera que se inicio el período de silencio. Cuando se inicia el período de silencio se comprueba que se hayan detectado por lo menos tres periodos de tono para considerar que se ha detectado un dígito valido, por lo que se procede a asignar el valor del digito detectado a la variable *DTMFin*. Los valores para *DTMFin* van de 0 a 9 y para * y #, 0x0A y 0x0B respectivamente. En la figura 16 se muestra el diagrama de flujos para la rutina de validación de DTMF.

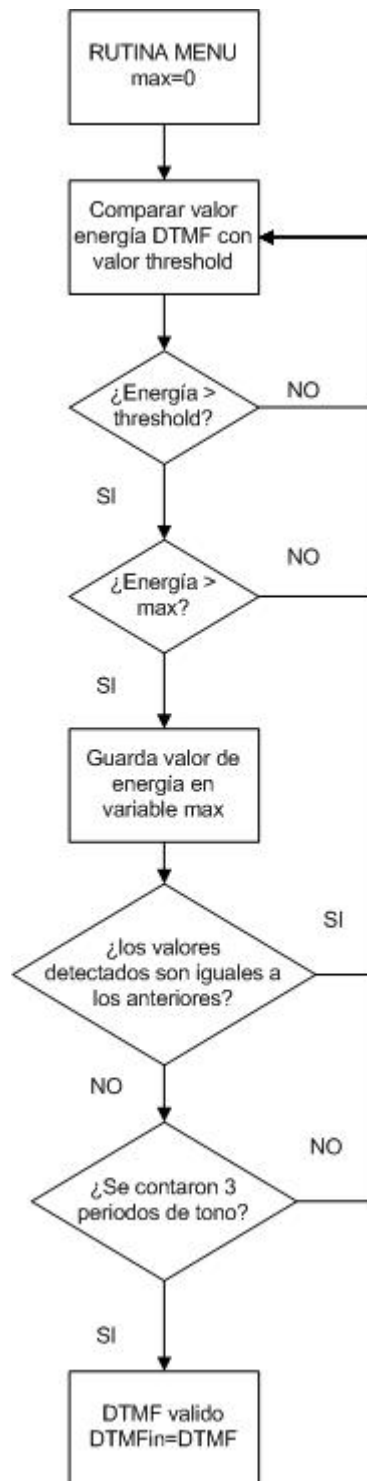


Figura 16: Validación DTMF detectado

4.3 Software adicional para compresión de audio

Todos los archivos de sonido que se van a reproducir en el microcontrolador primero deben pasar por un proceso de compresión y adecuación con la ayuda de algunos software dedicados para tal fin. Además hay que crear los archivos que se reproducirán, por lo que hace falta software para la grabación y edición de audio, a continuación se nombrarán y describirán todos los programas de computación adicional para poder reproducir audio por medio del microcontrolador.

4.3.1 Software para la grabación de voz

Un método que comúnmente se usa para la grabación de voz, es de disponer de un estudio de grabación donde se reduce el ruido del ambiente para que la voz a grabar quede lo mas entendible posible. Este método supone un problema adicional para quien no disponga de una sala de grabación. Por lo que se utilizó en lugar de una voz humana un emulador de voz para grabar cada uno de los mensajes que se reproducirán en el microcontrolador. Al utilizar un emulador de voz se obtiene un mensaje audible sin ningún tipo de ruido de fondo y sin problemas de pronunciación o tono de voz. Existe una gran variedad de software dedicado a la emulación de voz, por lo que a la hora de elegir uno se debe tener en cuenta que pueda grabar un archivo en formato WAV, que se pueda variar la velocidad de reproducción y que posea el idioma que se desea para los mensajes en el microcontrolador. Al guardar el archivo en formato WAV se puede abrir con la gran mayoría de los editores de audio existentes y al poder variar la velocidad con que se reproduce la voz se puede obtener un mejor resultado y a mayor velocidad, menor el tiempo de reproducción por palabra y menor el tamaño del archivo a guardar en el microcontrolador, lo cual es muy importante ya que todas las palabras serán almacenadas en la memoria de programa del microcontrolador (64kbytes). El software utilizado para la emulación de voz fue SAPI5 TTSApp de Microsoft Corporation.

4.3.2 Software para la edición de sonido

El software para la edición de sonido es necesario para manipular el archivo de audio en un formato compatible para ser reproducido por el microcontrolador corriendo el algoritmo ADPCM. Existen muchos editores de audio que se pueden conseguir pero el que se desee usar debe tener la posibilidad de adaptar el archivo de audio en formato WAV al formato compatible para la compresión ADPCM. Las características que debe tener el editor de audio son las siguientes:

- Re-muestrear al audio a 8kHz
- Convertir un archivo WAV a un formato RAW mono unsigned 16bit little endian PCM.

El editor que se uso fue el Goldwave que es fácil de usar con una versión de muestra descargable desde su página web y que puede transformar el formato de audio al necesario para la compresión IMA-ADPCM que es un RAW 8kHz mono unsigned 16bit little endian PCM. Los archivos RAW pueden tener extensión de .raw o .snd dependiendo del editor.

4.3.3 Software para la compresión de audio

Una vez convertido el archivo de audio al formato RAW, es necesario transformarlo en un archivo adecuado para el algoritmo ADPCM implementado en el microcontrolador. El fabricante del microcontrolador, Microchip, provee una herramienta para la compresión del archivo .raw, de nombre Winspeech.

Winspeech es una utilidad para convertir un archivo RAW unsigned 16bit little-endian en un archivo de audio comprimido ADPCM compatible con el algoritmo IMA-ADPCM implementado en el microcontrolador. El programa también puede descomprimir archivos que pudiesen ser recuperados de un microcontrolador

con el algoritmo de compresión IMA-ADPCM implementado. En la figura 17 se muestra la interfaz gráfica del Winspeech.

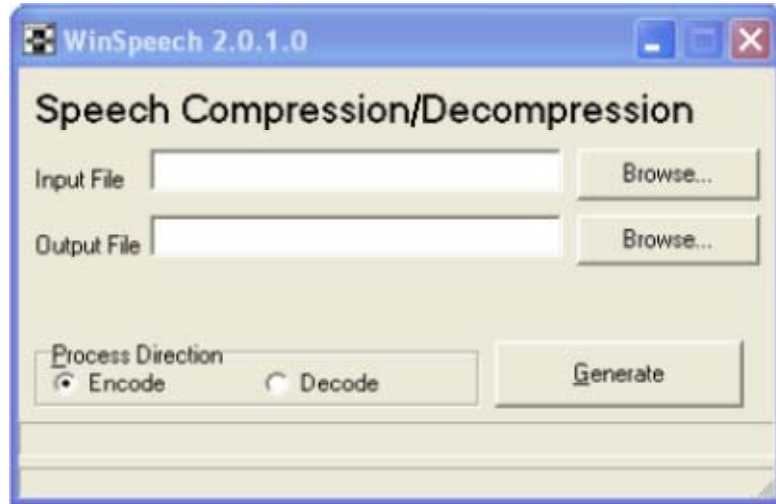


Figura 17: Interfaz gráfica del Winspeech

4.3.4 Software para el almacenaje del audio

Para aplicar el algoritmo ADPCM se usaron las librerías que provee Microchip para sus microcontroladores, en ellas utilizan un sistema de almacenaje de archivos llamado Microchip File System o MPFS. El MPFS es un sistema de archivos que ocupa poco espacio en memoria del microcontrolador y es de uso exclusivo para los microcontroladores PIC y dsPIC de Microchip y viene integrado en el paquete de almacenaje de páginas web en microcontroladores TCP/IP Stack. El Microchip TCP/IP Stack es un conjunto de programas que ofrece servicio a los programas basados en el estándar TCP/IP como por ejemplo servidores HTTP, clientes de correo, etc.

Originalmente para almacenar páginas web, el MPFS puede almacenar archivos que usen el estándar 8.3 para nombres cortos de archivos (8 caracteres para el nombre, 3 para la extensión) en la memoria de programación del microcontrolador o en una EEPROM serial externa con un formato especial de almacenamiento. El

formato de almacenamiento empieza con una o más entradas MPFS FAT (File Allocation Table) seguido de los datos de los archivos. En la figura 18 se muestra el formato de la imagen MPFS a almacenar en el microcontrolador. [14]

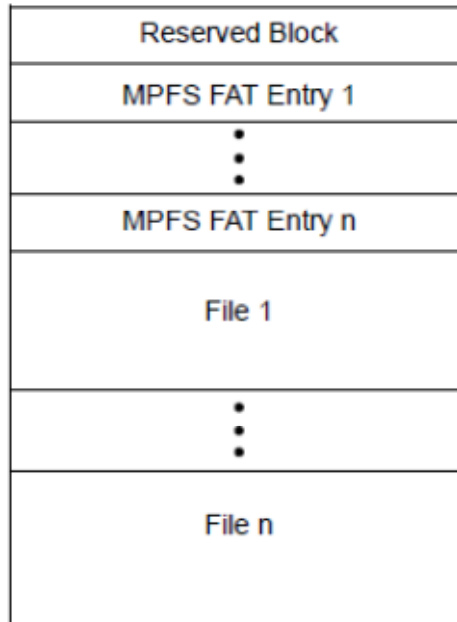


Figura 18: Formato de la Imagen MPFS [14]

Herramienta para la creación de Imágenes MPFS

El MPFS es un programa para PC que viene con el paquete TCP/IP Stack. Existen varias versiones de este programa, la usada para el presente trabajo fue la MPFSv2. El MPFSv2 crea un archivo assembler para MPLAB ASM30 con la imagen FAT de todos los archivos que se desean almacenar, los archivos deben ser reunidos en una sola carpeta de la PC y luego decir la dirección de esa carpeta al MPFSv2. En la figura 19 se muestra la interfaz de usuario del MPFSv2.

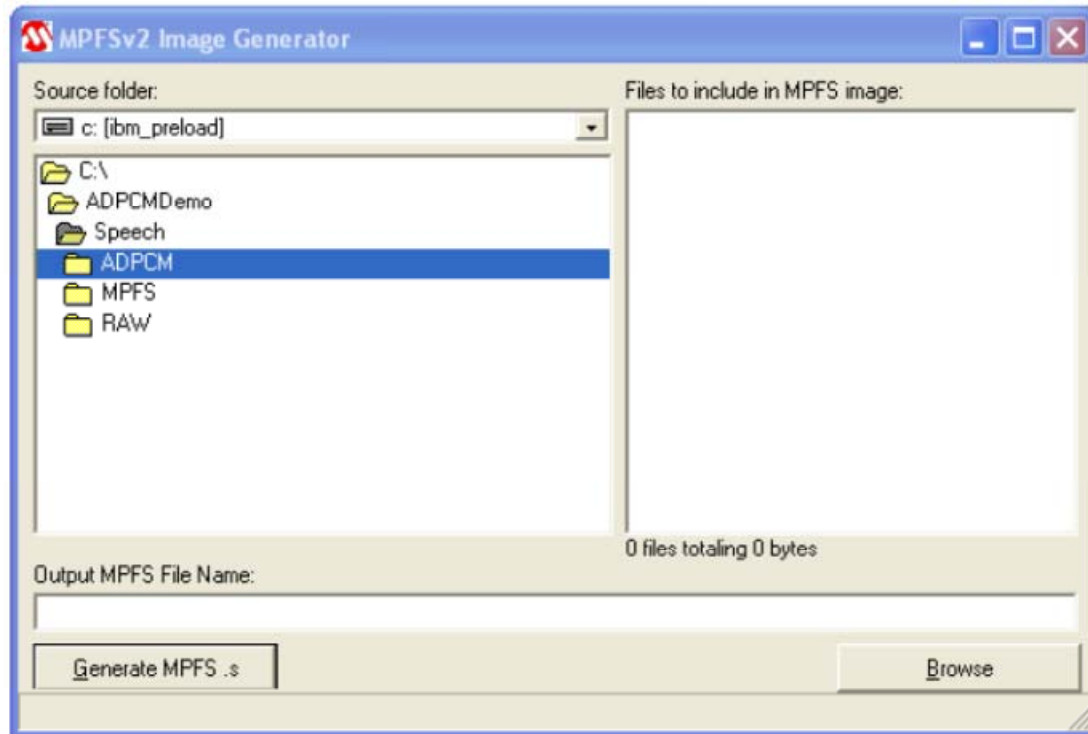


Figura 19: MPFSv2 Image Generator

4.4 Rutina para la reproducción de audio.

La rutina para la reproducción de audio son las que proporciona el fabricante de los microcontroladores PIC y dsPIC, Microchip, en su Nota de Aplicación AN643 “*Adaptive Differential Pulse Code Modulation Using PIC Microcontrollers*”. Allí se proporcionan las rutinas necesarias para la descompresión del archivo de audio con el algoritmo ADPCM y su integración a las rutinas de lectura de imagen MPFS de las librerías TCP/IP Stack. [5]

La rutina para la toma de archivos de la imagen MPFS viene en el paquete de librerías que distribuye Microchip en el TCP/IP Stack. La versión usada en el presente trabajo fue rev. 2.21. La rutina recibe un arreglo con cada uno de los caracteres del nombre FAT del archivo en código ASCII para ubicar la posición en memoria del archivo. Los datos del archivo están en una tabla donde el primer dato está en la en la dirección de memoria que hace referencia el nombre FAT. La rutina

toma cada uno de los datos de 8 bits de esta tabla y los asigna a la variable *hFile* que es el dato a introducir en la rutina de descompresión ADPCM. Se hará esto con cada dato contenido en la tabla hasta llegar al “Fin de Archivo” (EOF) de la tabla.

La rutina de descompresión ADPCM aplica el algoritmo IMA-ADPCM y transforma cada uno de los datos de 8 bits de la tabla en datos de 16 bits, luego le asigna este valor al registro de 16 bits OC1RS que lo coloca a la salida del PWM del microcontrolador. Esto debe ocurrir a una frecuencia de 16kHz, por lo que se sincroniza la rutina con el Timer 2 del microcontrolador. En la figura 20 se muestra el diagrama de flujo para la rutina de reproducción de audio.

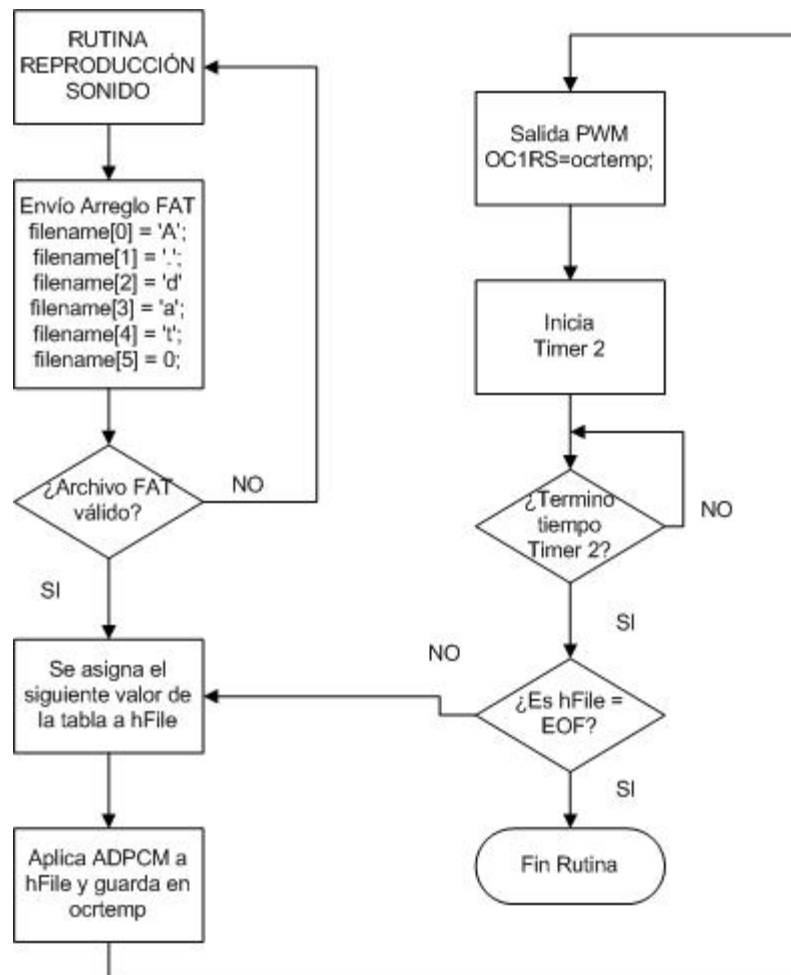


Figura 20: Rutina de reproducción de audio

4.5 Rutina Principal de control.

La Rutina Principal de control es la que estará corriendo todo el tiempo en un bucle infinito mientras el microcontrolador esté encendido. De él partirán las demás rutinas dependiendo de la situación que se presente. La rutina principal tendrá dos estados primarios, uno de “dormido” y otro de “despierto”. Cuando el microcontrolador este dormido el oscilador interno estará configurado como “Low Power Internal RC” (LPRC) lo que pone el microcontrolador a funcionar a 31kHz y cuándo este despierto el oscilador interno estará configurado como “Fast Internal RC” (FRC) con PLL interno (FRCPLL) lo que pone el microcontrolador a funcionar a 32MHz.

Las rutinas que ejecutara la Rutina Principal cuando esté en estado de dormido serán las siguientes:

- Las rutinas de monitoreo de las entradas, siempre que estén activadas las alarmar.
- Encender y apagar el led testigo de encendido en cada bucle del programa
- Monitorear si se oprime uno de los pulsadores
- Esperar la interrupción INT0 para responder una llamada

Las rutinas que ejecutara la Rutina Principal cuando esté en estado de despierto serán las siguientes:

- Las rutinas de monitoreo de las entradas, siempre que estén activadas las alarmas.
- Mantener prendido el led testigo de encendido.
- Poner en 3,3V el pin RA1 para alimentar los demás ICs de la RTU
- Monitorear si se oprime uno de los pulsadores
- La rutina de Menú Interactivo de la RTU

4.5.1 Rutinas en estado “dormido”

Cuando la variable *sleep* este en 1 el microcontrolador estará en un estado de dormido. En este modo se ejecutaran un mínimo de rutinas dado que la velocidad del oscilador es la más baja y el consumo de energía de la RTU a la central telefónica debe ser la menor posible.

Rutina de monitoreo de entradas.

Las entradas del RTU analógicas y los puertos digitales se pueden configurar como alarmas. Dentro del Menú Interactivo de la RTU se podrá elegir como una alarma al analógico digital 1 y a los 4 puertos digitales siempre y cuando éstos estén como puertos de entrada o lectura.

La rutina de monitoreo de entradas verifica que se haya configurado alguna de las posibles alarmas como tal y de ser cierto verifica que el valor a la entrada sea motivo de una alarma, de ser así la rutina llamará al número grabado en memoria para notificar la alarma.

Led testigo de encendido de la RTU

El led testigo de encendido es un indicador visual de que la RTU está funcionando, en estado de dormido el led oscilará con cada bucle que haga la Rutina Principal. De esta forma el led consume menos corriente y se aprecia la luz que emite para verificar que el RTU está funcionando.

Monitoreo de los pulsadores.

En cada ciclo de la Rutina Principal se verificarán que los pulsadores hayan sido oprimidos leyendo los pines a los que están conectados. Para impedir ruidos y chispas de los pulsadores, se programó la rutina para que reconozca una pulsación solo después de verificar que el pin ha estado en un nivel bajo (nivel de los pulsadores) por un periodo de tiempo continuo.

Interrupción de llamada con el INT0

La interrupción externa 0 del microcontrolador (INT0) está conectada a la salida de tono de llamada RING del módulo de interfaz con la línea telefónica. Cuando entra una llamada por la línea de teléfono llega a través de RING a la entrada INT0 del microcontrolador, lo que dispara la interrupción y descuelga la línea para recibir la llamada.

4.5.2 Rutinas en estado “despierto”

Cuando la variable *sleep* esté en 0 el microcontrolador estará en un estado de despierto. Para este estado la línea estará descolgada por lo que se podrá tomar toda la energía necesaria para el funcionamiento de todas las funciones de la RTU. Se podrá acceder a cada una de las funciones del RTU por medio del Menú Interactivo que estará corriendo en el microcontrolador.

Rutina de monitoreo de entradas.

Correrá de la misma manera que en el modo dormido con la salvedad de que no realizará la llamada telefónica, sólo interrumpirá el Menú Interactivo para dar a conocer la alarma.

Led testigo de encendido de la RTU

Como la RTU tiene suficiente corriente para su funcionamiento, el led testigo permanecerá encendido sin oscilar.

Encendido de RA1.

Cuando el microcontrolador pasa al estado de despierto se coloca en 1 lógico el pin RA1 con lo que se alimenta el amplificador operacional y el IC para la comunicación RS-485 junto con los condensadores y resistencias que estos necesitan. También se encenderá el led D3 como testigo de que el microcontrolador está despierto. Cuando el microcontrolador pasa al estado de dormido pone el pin RA1 en 0 lógico, con lo que se elimina el suministro de corriente.

Monitoreo de los pulsadores.

Correrá de la misma manera que en estado dormido.

4.6 Rutina Menú Interactivo de la RTU.

La rutina Menú Interactivo es la que transmite el mensaje hablado que guiará al usuario del otro lado de la línea telefónica a acceder a todas las funciones de la RTU. Básicamente, es una maquina de estados que cambiará dependiendo del dígito marcado en el teclado del teléfono con el que mantiene la llamada. El Menú Interactivo se transmitirá una y otra vez si no se recibe un tono DTMF valido por un periodo de 15 segundos colgara de forma permanente a menos que el menú esté en una alarma, lo que hará que después de 15 segundos sin recibir un DTMF cuelgue y vuelva a marcar al número almacenado en memoria. En la figura 21 se muestra el menú principal, de donde el usuario deberá partir para acceder a la RTU vía telefónica con cuatro opciones a elegir.

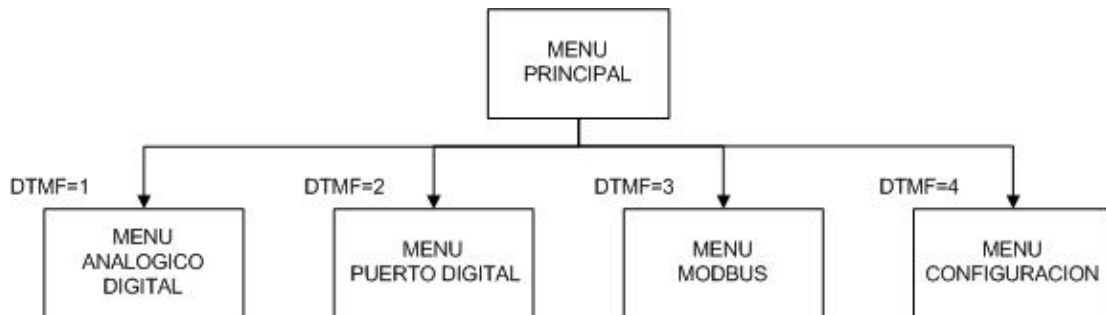


Figura 21: Menu Principal

Menú Analógico Digital:

El menú analógico digital presenta al usuario todas las opciones disponibles en la RTU para sus puertos de lectura analógico digital. Las opciones son:

- Leer analógico digital 1

Para leer el valor presente a la entrada del A/D 1. La respuesta será un número entre 0 y 1024 pronunciado dígito por dígito con el valor del AN2 del microcontrolador.

- Leer analógico digital 2

Para leer el valor presente a la entrada del A/D 2. La respuesta será un número entre 0 y 1024 pronunciado número por número con el valor del AN2 del microcontrolador.

- Configuración Analógico Digital

Con esta opción se configura la alarma para el convertidor analógico digital. Para este RTU sólo se tiene alarma en el analógico digital 1. Al marcar la opción 1 del menú configuración lee el estado actual de la alarma junto con el

valor actual de alarma, luego da la opción para activar la alarma o desactivarla, dependiendo con el estado actual de la alarma, presionando 1, y presionando 2 repite el estado de la alarma, si esta activo o desactivo junto al valor actual de alarma y si se avisa de una alarma para valores mayores o menores al número introducido como de alarma. Si se presiona 1 y la alarma está activa, se desactiva y lee el estado de la alarma para confirmarlo, si la alarma esta desactiva, se procede a activarla y a pedir el valor para la alarma, un número entre 0 y 1024 que se introduce digito por digito, luego preguntara si es la alarma es para valores mayores o menores al introducido. Si se introduce un valor de 512 y se le indica que la alarma es para mayores, cuando el microcontrolador detecte algún valor mayor o igual a 512 se producirá una alarma, si la opción fue menor, la alarma será con valores de 512 para abajo. En la figura 22 se muestra el diagrama para el menú analógico digital.

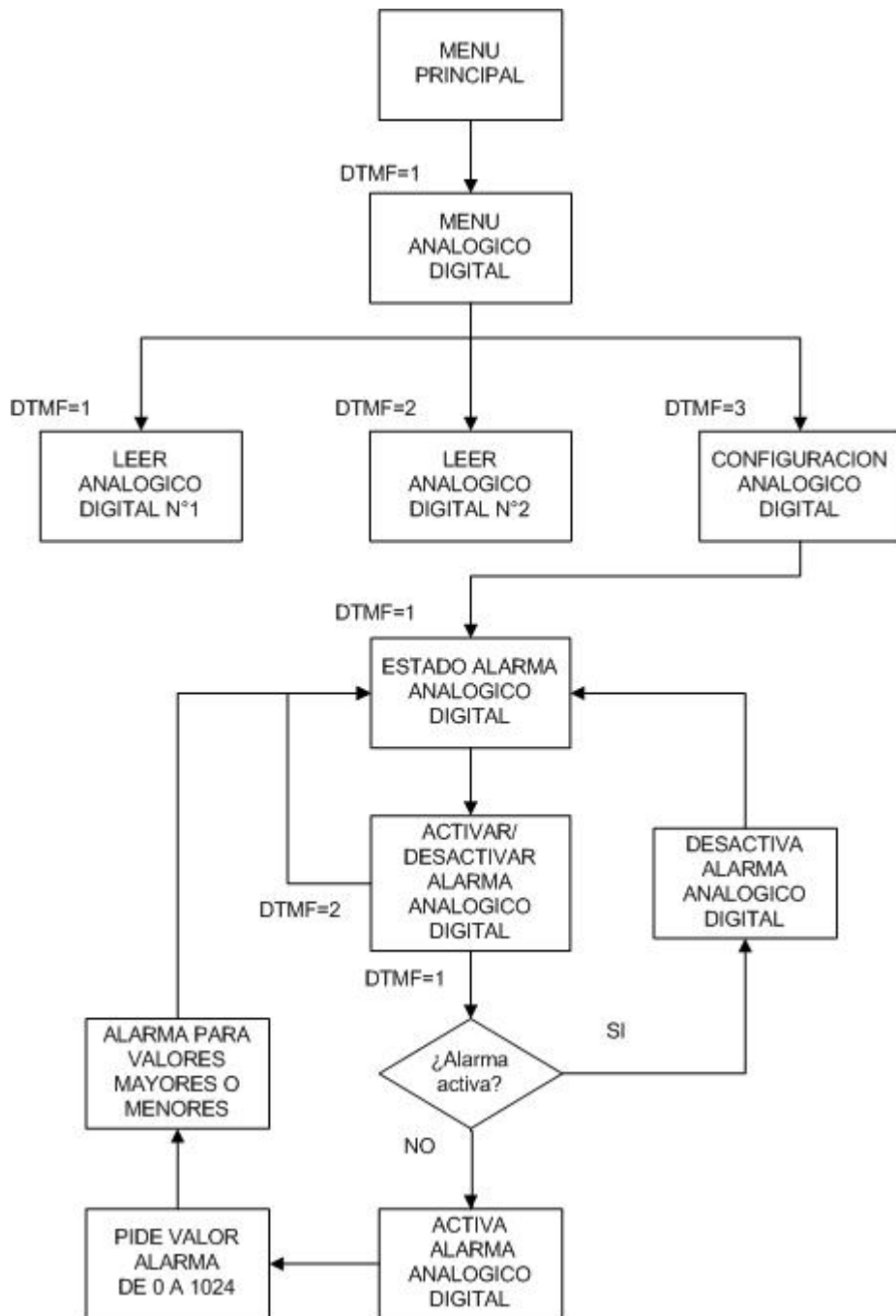


Figura 22: Menú Analógico Digital

Menú Puerto Digital

El menú puerto digital presenta al usuario todas las opciones disponibles en la RTU para sus puertos de lectura y escritura digitales. Las opciones son:

- Leer puerto digital

Para leer el valor de los puertos digitales, se selecciona el puerto que se desea leer y el RTU responderá pronunciando el número 0 ó 1 dependiendo del valor en el puerto seleccionado.

- Escribir puerto digital

Para escribir el valor de los puertos digitales, se selecciona el puerto que se desea escribir, se marca un número 0 ó 1 y el RTU colocará en el puerto digital seleccionado un nivel alto si se marcó 1 o un nivel bajo si se marcó 0. Para poder escribir un puerto digital primero tiene que configurarse como salida en el menú configuración puerto digital.

- Configuración Puerto Digital

En este submenú se configuran todas las opciones disponibles para los puertos digitales. Al presionar 1 se puede elegir cuáles de los puertos digitales estarán configurados como entradas, al presionar 2 se selecciona los puertos configurados como salida, presionando 3 el RTU dirá la configuración actual de los puertos, (si son de entrada o de salida). Presionando 4 se entra en un submenú de configuración para la alarma de los puertos digitales, el RTU nombrará cada uno de los puertos indicando si están activados o desactivados, al seleccionar un puerto éste se desactivará o activará según sea el caso. Al activarse la alarma de un puerto digital, el RTU solicitará que se marque el

valor para el cual se debe producir una alarma, 0 ó 1. En la figura 23 se muestra el diagrama el menú puerto digital.

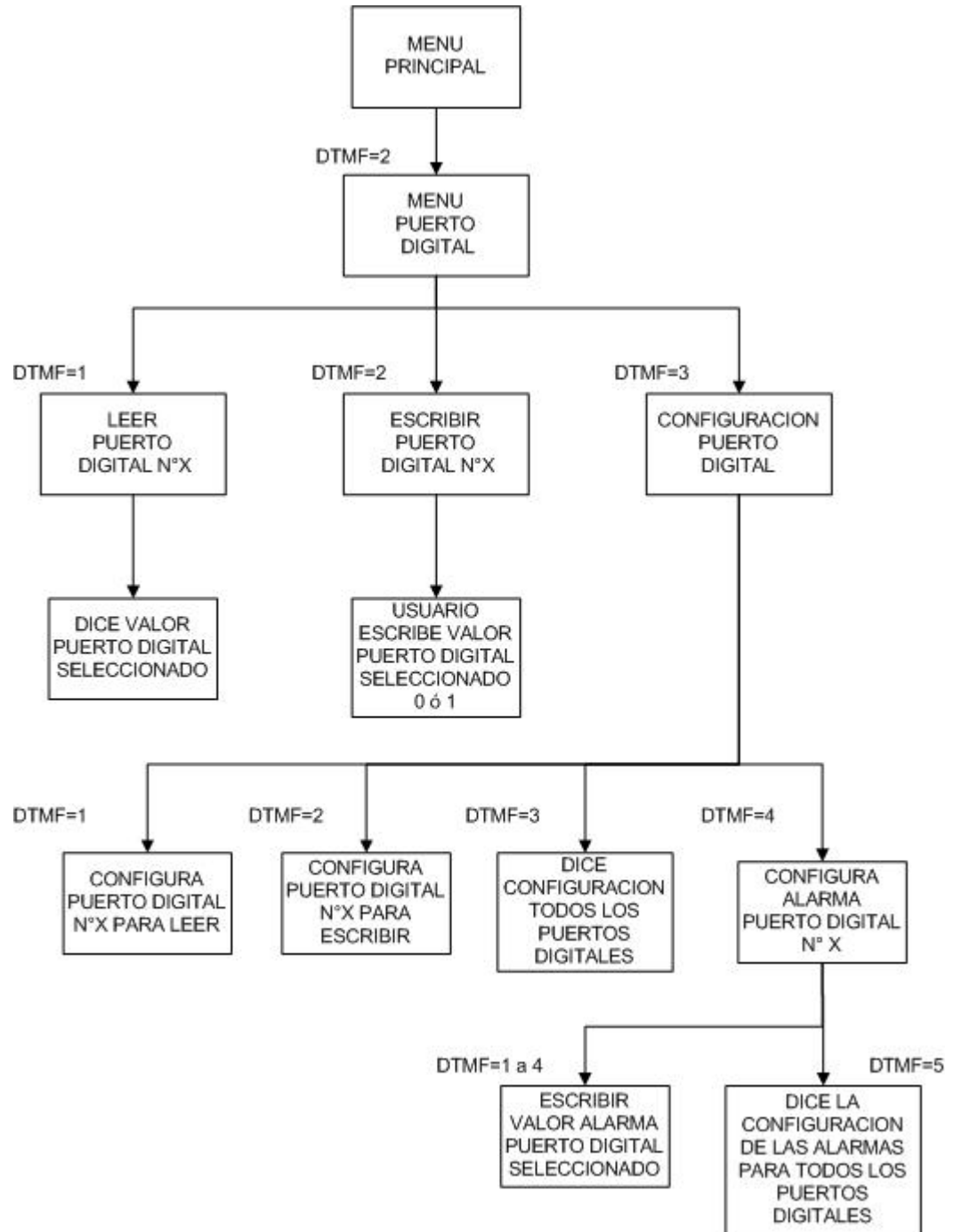


Figura 23: Menú Puerto Digital

Menú MODBUS.

Al presionar 3 en el menú principal se accede al menú MODBUS. En este menú se puede enviar y recibir paquetes bajo el protocolo MODBUS. Al entrar al menú el RTU pedirá los datos para formar el paquete que enviara vía RS-485 a las posibles remotas que estén por debajo de ella. El RTU será siempre un amo, por lo que no acepta mensajes de petición de sus esclavos. El modulo MODBUS del RTU sirve sólo para leer información de las remotas y no tiene opción para activación de alarmas. El primer dato que pedirá la RTU es el número del esclavo MODBUS al que se quiere enviar el paquete (un número de 0 a 254), seguido de la función (otro número de 0 a 254), la dirección de inicio y el número de registros, cada uno de 0 a 65534. Una vez ingresado el último valor la RTU transmitirá el paquete y esperará la respuesta. Al recibirla la RTU dirá los valores del paquete en el siguiente orden: primero el número del dispositivo, seguido por la función, y después dirá los siguientes paquetes de 16 bits. La RTU identificará el paquete y seguidamente pronunciará cada uno de los números del paquete en decimal.

Menú Configuración.

Este menú es la opción 4 del menú principal y en él se pueden hacer configuraciones generales del RTU, como asignarle un número que identifique a la RTU y guardar el número telefónico al que debe discar en caso de una alarma. La opción 1 permitirá almacenar un número para que la RTU se identifique cuándo haga una llamada. Esto será muy útil para saber cual RTU es la que presenta la alarma en el caso de que existan más de una. La opción 2 permitirá asignarle un número de teléfono distinto al que viene por defecto a la RTU, opción muy importante ya que no sería muy útil un dispositivo que solo pueda llamar a un solo teléfono.

Consideraciones Generales para el Menú Interactivo:

- En todos los menús, la tecla asterisco regresa al menú principal, excepto en el menú de configuración de alarma del puerto digital, que la tecla asterisco regresa al menú configuración de puerto digital.
- Pasados 15 segundos el RTU colgara la línea telefónica si no detecta ningún tono DTMF

4.7 Rutina Menú Alarma del RTU

La rutina de menú alarma se ejecutará cuando el teléfono llamado para comunicarle la alarma conteste. La rutina responde diciendo que ha ocurrido una alarma y continúa diciendo el número de la RTU que llama y la alarma que se ha producido. El menú da tres opciones para elegir al producirse una alarma, el usuario puede leer el valor que produjo la alarma, puede leer el valor que está configurado para producir la alarma y puede desactivar la alarma. Si el usuario no responde la llamada telefónica, el RTU intentará hasta que sea atendido. Si el usuario atiende pero no selecciona ninguna de las opciones en un lapso de 15 segundos, el RTU colgará y volverá a llamar hasta que el usuario desactive la alarma.

4.8 Rutinas de Colgar y Descolgar

Las rutinas de colgar y descolgar son las que cambian el oscilador interno del microcontrolador de LPRC a FRCPLL y viceversa. Esto es posible gracias a la posibilidad de los microcontroladores PIC24 y dsPIC de cambiar su oscilador interno por software cambiando el registro OSCCONH y OSCCONL. La rutina para el cambio de oscilador está diseñada para ser implementado en assembler, dado que los pasos a seguir para el cambio tienen que ser muy precisos. Por eso el conjunto de herramientas de programación Microchip C30 tiene un comando precompilado para ejecutar el cambio de oscilador donde solo hay que colocar el comando con el valor para el nuevo oscilador y el compilador se encargará del resto. A continuación un ejemplo de código para cambio de oscilador de FRCPLL a LPRC con C30:

```
__builtin_write_OSCCONH(0x05);  
__builtin_write_OSCCONL(0x05);  
OSCCONbits.OSWEN =1;  
while(OSCCONbits.OSWEN);
```

El bit OSWEM se coloca en cero cuando el cambio del oscilador esta completo. Es recomendable deshabilitar todas las interrupciones durante el cambio de oscilador. [15]

CAPÍTULO V

5. IMPLEMENTACION Y PRUEBAS

5.1 Implementación del hardware de la RTU

En el capítulo 3 se describe con detalle el hardware diseñado para la RTU. Ahora se describirán las consideraciones y técnicas empleadas para su implementación.

5.1.1 Resistencias

Todas las resistencias empleadas para la implementación de la RTU fueron en su mayoría de carbón aglomerado de 1/4W, de la serie del 5%, a excepción de las empleadas en el módulo de amplificación y filtro, en cual se emplearon resistencias del 1%.

La resistencia R14 en el módulo de amplificación y control se diseño para un valor de 600 Ω . Dado que este no es un valor comercial se eligió el siguiente disponible en el mercado, se consiguió una resistencia de 604 Ω de película de metal de la serie del 1% y de 1/4W.

La resistencia R4 diseñada para un valor de 9,31k Ω es un valor comercial de la serie del 1%, pero no se encontró disponible en el país, por lo que se utilizó una resistencia de valor 9,53k Ω de tolerancia $\pm 1\%$ de película de metal de 1/4W.

La resistencia R5 se diseño con el valor comercial de 19,1k Ω , cuyo valor se encontró y empleo. Es una resistencia de película de metal de la serie del 1% y 1/4W.

La resistencia R6 diseñada para un valor de 8,45k Ω fue implementada con una de 8,06k Ω de película de metal del 1% y 1/4W disponible en el país.

La resistencia R7 diseñada para un valor de 12,4k Ω se implementó con una de 12,7k Ω de película de metal de la serie del 1% y 1/4W.

El resto de las resistencias tienen disponibilidad en el país, por lo que no hubo cambio en sus valores.

5.1.2 Condensadores

Los condensadores diseñados para el RTU están disponibles en el mercado nacional por lo que no hubo mayores inconvenientes a la hora de su implementación.

Los condensadores usados para la implementación fueron:

- C1 = 100nF \pm 20% 100V polyester
- C2 y C5 = 2,7nF \pm 20% 25V polyester
- C3 = 3,3nF \pm 20% 25V polyester
- C4 = 5.6nF \pm 20% 63V polyester
- C6, C7 y C15 = 1uF \pm 10% 50V electrolítico
- C8 = 10nF \pm 20% 63V polyester
- C9 = 150pF \pm 20% 3kV cerámico
- C10, C13 y C17 = 15nF \pm 20% 250V polyester
- C11 y C12 = 1000pF \pm 10% 3kV cerámico
- C14 = 1nF \pm 20% 25V polyester
- C16 = 6,8uF \pm 10% 100V electrolítico
- C18 = 220uF \pm 10% 25V electrolítico
- C19 = 39nF \pm 10% 16V cerámico
- C21 = 1000uF \pm 10% 100V electrolítico
- C22 = 10uF \pm 10% 16V tantalio

5.1.3 Ajustes por cambio de valores y tolerancias

Para el presente trabajo no se consideró necesario un ajuste por los cambios en las resistencias y las tolerancias, ya que el cambio de la frecuencia de corte del filtro pasa bajo fue diseñado para una frecuencia de corte de 4kHz y el audio puede ser transmitido hasta con un ancho de banda de 3,3kHz sin alterar su calidad.

5.1.4 Consideraciones térmicas

El equipo fue diseñado para trabajar a temperatura ambiente (25°C), su funcionamiento en condiciones extremas no está garantizado. La disipación de potencia por parte de los componentes utilizados no se considera significativa, dado que el consumo de corriente es el menor posible dada las condiciones planteadas para el presente trabajo.

5.2 Implementación del software de la RTU

En el capítulo 4 se describió con detalle el software que controlara la RTU. Ahora se describirá la forma en que se implementó. Todos los programas utilizados para la programación corrieron bajo el sistema operativo Windows 7 Ultimate.

5.2.1 Programación del Firmware de la RTU

La programación de la RTU fue implementada en el lenguaje ANSI C, bajo la plataforma MPLAB IDE v8.63 con el Tool suite MPLAB C30 v3.23. Se utilizó las librerías TCP/IP Stack versión 2.21 y las librerías propias del MPLAB, el resto de las rutinas fueron de programación propia para este trabajo. El programa final compilado ocupó el 99% de la memoria de programación del microcontrolador PIC24FJ64GA002, siendo la imagen MPFS la que ocupó más espacio en el microcontrolador, 49836 bytes de 64305 bytes (77% de la memoria de

programación), esto a pesar de grabar el audio a su máxima velocidad para ocupar el menor espacio posible.

El firmware fue transferido al microcontrolador directamente en el RTU aprovechando la capacidad de Programación en Circuito que poseen los microcontroladores. Esta habilidad permite al microcontrolador ser programado estando instalado dentro de un circuito, por lo que no es necesario sacar el microcontrolador de su lugar en el circuito impreso para tener que instalarle un nuevo programa. El programador usado fue el PICKIT 3 del fabricante Microchip que transfiere el programa directamente del MPLAB IDE al microcontrolador por vía de un puerto USB de la PC.

5.2.2 Grabación y Compresión de audio

En el capítulo 4 se explico el procedimiento a seguir para la grabación y posterior compresión del audio a reproducir en la RTU. Ahora se describirá la forma en que se implemento para que el RTU pronunciará palabras.

La grabación del audio se realizó con el sintetizador de voz SAPI5 TTSApp de Microsoft Corporation en la figura 24 se muestra el programa SAPI5 TTSApp.

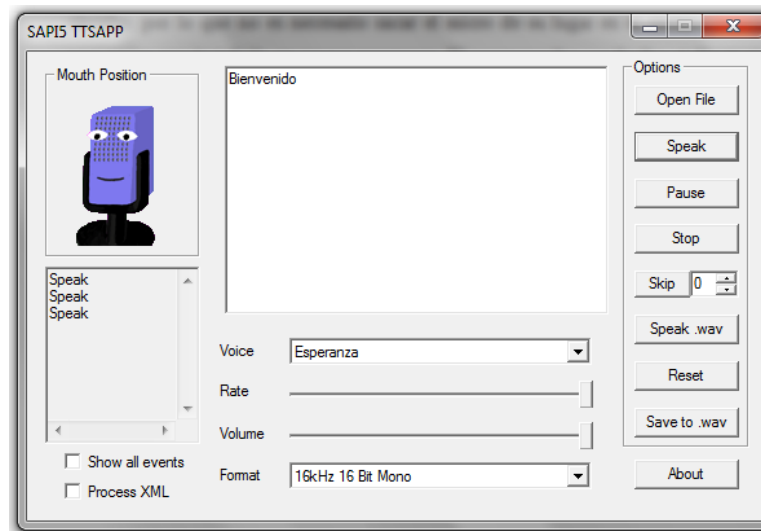


Figura 24: Programa sintetizador de audio

Este es un programa de fácil manejo en el cual se escribe la palabra que se desea grabar, se selecciona una voz para que lo pronuncie, la velocidad y el volumen y este programa puede reproducir la palabra por los altavoces de la PC o grabarlo en un archivo WAV. La velocidad usada fue la máxima permitida, esto por razones de espacio de la memoria del microcontrolador, ya que mientras más rápido se pronuncien las palabras, más palabras se podrán grabar en un mismo espacio. Cada segundo de audio es aproximadamente 4kbytes de memoria de microcontrolador. Se grabaron un total de 34 palabras, 10 tonos DTMF y un tono de silencio de 0,25 segundos de duración.

Luego se modifico el formato de los .wav para adaptarlos al requerido por la ADPCM. Para esto se uso, como ya se menciona en el capítulo 4, el programa Gold Wave versión 5.55, mostrado en la figura 25.



Figura 25: Programa de edición de audio

Con este programa se remuestrearon los 45 archivos .wav a 8kHz y se guardaron como Raw PCM unsigned 16 bit, Little endian, 8kHz, Mono.

El siguiente paso es la compresión de los archivos con la herramienta Winspeech. Para ello es necesario seleccionar cada archivo y guardar el resultado comprimido en una sola carpeta y en formato 8.3 para la creación de la imagen MPFS. Los archivos comprimidos se nombraron con letras mayúsculas para las palabras, números para las palabras correspondientes a 0 a 9, números precedidos por la letra “d” para los tonos DTMF y “@” para el tono de silencio. Todos los archivos se guardaron con extensión .dat para ser reconocidos por el MPFSv2.

Con todos los archivos comprimidos en la misma carpeta, esta se selecciona con el MPFSv2 para ser comprimidos. El resultado es un archivo con extensión .s que deberá ser incluido en el proyecto dentro del MPLAB IDE.

5.3 Resultados y análisis.

5.3.1 Conexión con la línea de teléfono.

La RTU se conectó a una línea de teléfono fija de la compañía CANTV. El voltaje nominal de la línea medido en bucle abierto fue de 51,6Vdc. Al conectar la RTU a la línea telefónica el condensador C21 de 1000uF comienza a cargarse, por lo que produce un pico de corriente en la línea de 15mA que la central interpreta como descuelgue, en menos de un segundo la corriente baja 2mA por lo que la central cuelga la línea. La corriente llega a 710uA y sigue bajando hasta después de pasados varios minutos la corriente se estabiliza a 410uA, y el voltaje en la línea es de 43,9V.

Cuando la RTU cambia al modo activo, la corriente consumida de la línea llega a 29,9mA y el voltaje medido a la entrada de la línea es 14,99V, la central detecta un descuelgue sin problemas. El voltaje suministrado por el LDO al microcontrolador se mantiene estable en 3,383V.

5.3.2 Pruebas de transmisión y recepción de audio.

Para las pruebas de audio se colocó un teléfono fijo conectado a la entrada Rx/Tx y el RTU alimentado por una fuente de 3,3V externa. La conexión del teléfono fijo se realizó como se muestra en la figura 26.

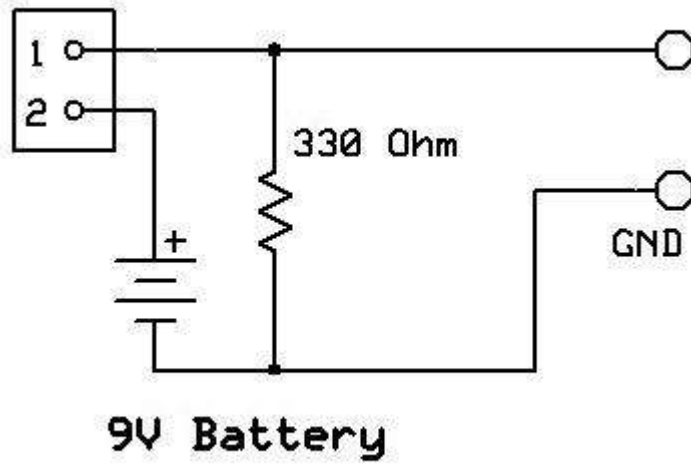


Figura 26: Conexión de un teléfono fijo

El teléfono se conecta a los terminales 1 y 2, la batería de 9V provee la corriente necesaria para el funcionamiento del teléfono, la resistencia de 330Ω sirve para el acoplamiento.

Se observó la forma de onda de una señal DTMF marcada en el teléfono a la entrada del operacional U2D y a la salida del operacional U2C para corroborar el paso del tono y la amplificación de la misma. En la figura 27 se muestra el resultado observado en el osciloscopio.

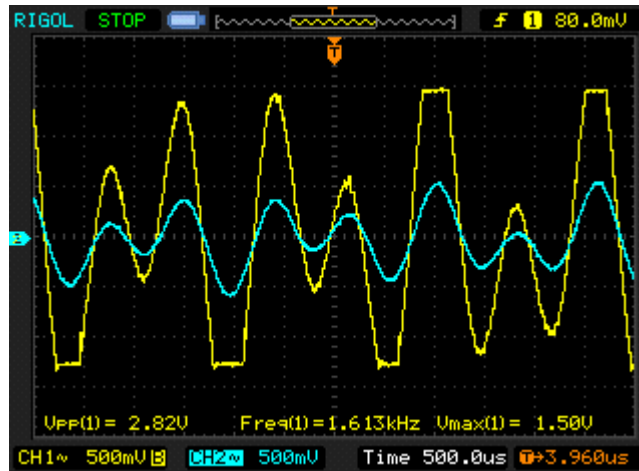


Figura 27: Amplificación DTMF

Como se observa en la figura 27, la amplificación es suficiente para la entrada del analógico digital. Cabe destacar que al usar el teléfono directamente conectado a Rx/Tx la amplitud de la señal es mayor al esperado de la línea telefónica.

Se conectó el osciloscopio a la salida de la señal de audio para observar la señal PWM filtrada, el resultado se presenta en la figura 28.

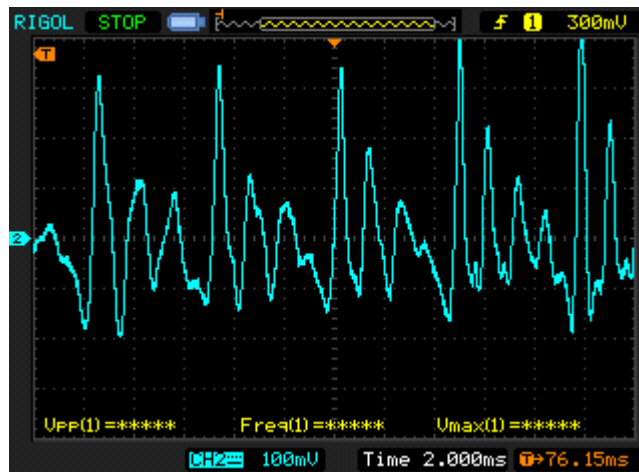


Figura 28: Señal de audio generada por el PIC24

El audio resulta entendible a través del auricular del teléfono, sólo que acelerado, dado al hecho de que se grabó a máxima velocidad para ahorrar espacio en

memoria de programación. En la figura 29 se puede observar la señal PWM antes de entrar en el filtro Butterworth y la señal de audio a la salida del filtro. Se puede apreciar cómo el ancho del pulso PWM modifica el audio a la salida.

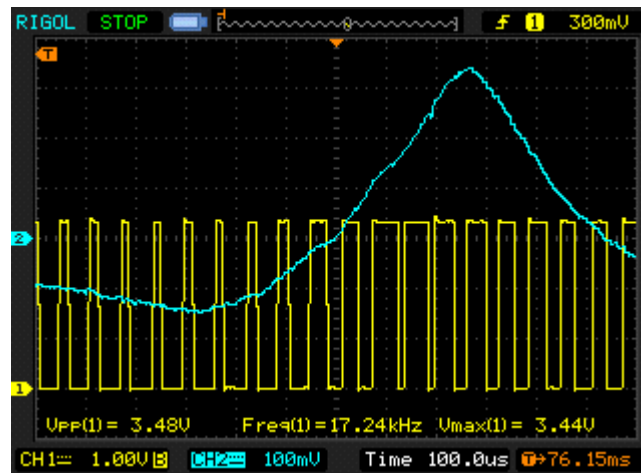


Figura 29: Señal PWM antes y después del filtro

Cuando no se está transmitiendo audio, el PWM se configura para trabajar a 50%, al conectar el osciloscopio y ver la señal se aprecia el duty cycle de 50% y la frecuencia de 16kHz. En la figura 30 se puede apreciar la señal PWM sin señal de audio.

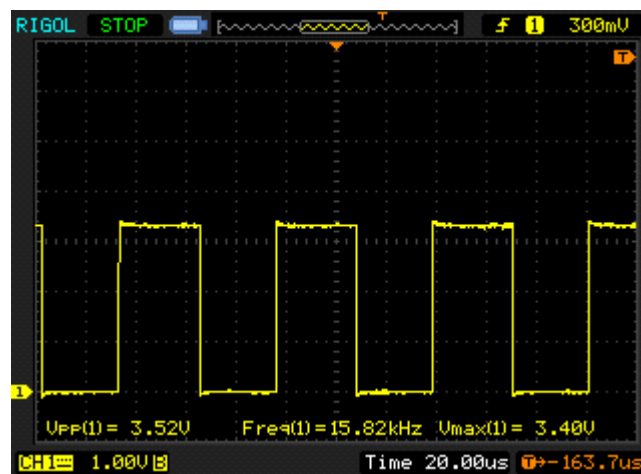


Figura 30: PWM sin señal de audio.

Al conectar la RTU a la línea de teléfono la comunicación se realizó sin ningún problema, la central reconoció los tonos DTMF almacenados en la memoria del microcontrolador y estableció la comunicación con el teléfono llamado. Al introducir los tonos DTMF desde el teléfono llamado la RTU los reconoció, solo se tiene que mantener presionado la tecla por un periodo de no menos de 1 segundo para que el microcontrolador lo pueda detectar correctamente.

5.3.3 Pruebas de las funciones de la RTU.

Se conectó el osciloscopio para ver el led testigo de encendido en el modo de reposo, el diodo oscilaba tal cual fue diseñado, la oscilación se puede apreciar en la figura 31.

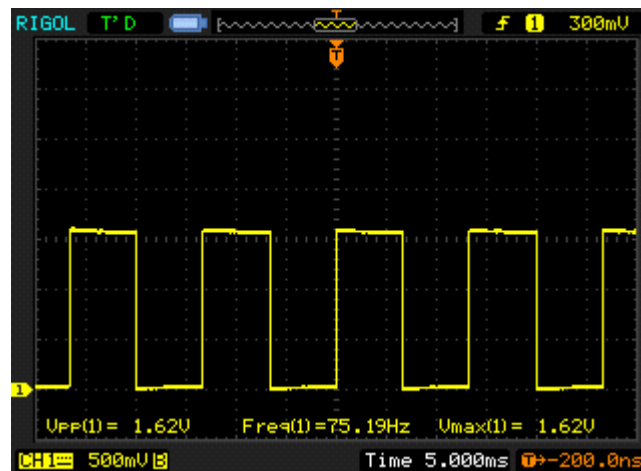


Figura 31: Señal en led testigo en estado de reposo del microcontrolador

Se probaron todas las configuraciones del RTU, se cambió el número del dispositivo y el número de teléfono pregrabado, se leyó valores de los puertos analógicos y de los puertos digitales, se configuraron las alarmas y se probaron para constatar la activación de la alarma. Todas las opciones de la RTU funcionaron dentro de los parámetros esperados.

CONCLUSIONES

El uso de la energía que suministran las centrales telefónicas a sus abonados para aplicaciones electrónicas es una opción bastante viable en la implementación de equipos de monitoreo remotos. El concepto de tomar energía de la línea telefónica no es nada nuevo, es lo que hacen los teléfonos fijos desde los inicios de la telefonía, y esa es la razón por la cual las centrales disponen de esa fuente de corriente para sus abonados, por lo que su uso no presume una violación de los términos bajo los cuales las compañías de teléfono prestan el servicio. Como se describió en el capítulo 2, las centrales tienen parámetros bajo los cuales suministran energía a sus abonados, por lo que el diseño e implementación de equipos que cumplan con estos criterios no deberían afectar el funcionamiento de la central telefónica ni de la línea de teléfono. Las fuentes conmutadas no son la mejor opción para la regulación de voltaje en sistemas de bajo consumo de energía. Las fuentes conmutadas presentan una mayor eficiencia a mayor energía manejada, y en sistemas donde se manejan corrientes por debajo de 1mA las fuentes conmutadas no obtienen suficiente corriente para su funcionamiento, por lo que el uso de los LDOs es la opción indicada para estos casos. Los LDO tienen años en la regulación de voltaje en sistemas cuya alimentación es a baterías, por lo que han sido desarrollados para consumir el mínimo de corriente que sea posible.

El algoritmo de Goertzel es una excelente opción para la detección de tonos con microcontroladores, ya que su tiempo de cálculo es mucho menor que la FFT por lo que la carga al microcontrolador es mínima, dejando al microcontrolador tiempo para ejecutar múltiples aplicaciones adicionales a la detección de tonos. En el presente trabajo la detección de tonos fue un elemento crucial y sin el algoritmo de Goertzel se habría tenido que recurrir a circuitos integrados adicionales, que consumirían más corriente de la central y supondría un costo adicional al equipo.

La reproducción de audio con el microcontrolador fue una pieza clave del presente trabajo, el uso de una interfaz hablada que guía a los usuarios a través del menú interactivo de la RTU proporciona un método de configuración a distancia muy útil para controlar múltiples procesos en varios lugares a la vez. Además se puede consultar el estado de todos los procesos que monitorea la RTU en tiempo real y desde cualquier teléfono, no necesariamente uno de última generación con acceso a internet y capacidad de mensajes de texto, sino con cualquier teléfono capaz de hacer y recibir una llamada. Las técnicas de reproducción de audio para microcontroladores es un tópico muy extenso, cada fabricante tiene sus propias herramientas y librerías para las aplicaciones “speech” como se le conocen, que significa “discurso” en inglés.

La implementación de una RTU como la descrita en el presente trabajo sería muy útil para el monitoreo en sitios remotos en lo que existen alta probabilidad de fallas de energía o que no allí cobertura de la red celular, sólo se necesita una línea de teléfono fija y el RTU hará su trabajo. Además el dispositivo solo requiere de un solo cable para su funcionamiento, al igual que los dispositivos PoE de red LAN, sólo se necesita instalar un cable para el suministro de energía y la transferencia de datos por lo que se podrían disponer de una central telefónica de oficina con líneas libres, y sólo haría falta instalar un único cable para monitorear procesos en varios puntos.

El bajo costo del RTU también es una gran ventaja. Al disponer de un mínimo de componentes y al mantener el funcionamiento en una falla de corriente sin necesitar de baterías hace que el dispositivo sea accesible y el uso de la energía de la línea telefónica significaría un ahorro en la factura de electricidad. Además, el diseño de poco consumo de energía del RTU representa una ventaja del punto de vista ecológico, además de no haber grandes pérdidas en fuentes de tensión que tomen la energía de la línea para reducirla al voltaje de funcionamiento del RTU.

RECOMENDACIONES

Con base en la experiencia vivida en el presente trabajo se presentan una serie de recomendaciones para futuras aplicaciones de este tipo o la implementación de manera comercial del presente RTU.

Se recomienda el uso de una memoria externa para el almacenaje de la imagen MPFS. Ya que cada segundo de audio es aproximadamente 4kbytes de memoria, el uso de una memoria externa daría mayor flexibilidad para grabar mensajes más largos. En el presente trabajo se guardaron palabras individuales y luego se formaron frases con dichas palabras, lo que en ocasiones genera mensajes no muy entendibles, el uso de una memoria externa permitiría el uso de frases completas lo que daría mensajes más claros y mucho más agradables que palabras por separado. Una memoria EEPROM I2C o SPI (se recomienda SPI para PIC24 o dsPIC) de 512kbytes se considera suficiente para cualquier aplicación, de necesitar más memoria se podría utilizar memorias SD.

Para el uso del puerto digital como salida, se recomienda el uso de un circuito adicional alimentado por una fuente distinta a la de la línea de teléfono, como se aconseja en el capítulo 3. Sin este circuito no es recomendado el uso del puerto digital como salida ya que consumiría corriente adicional de la línea telefónica.

Si se desea emplear el RTU en una aplicación comercial, se recomienda someterla a las validaciones que pueda pedir la compañía de teléfonos a la cual se desea conectar. La RTU trató de adaptarse a las exigencias de la FCC (Federal Communications Commission) pero no se garantiza su total cumplimiento, por lo que se recomienda someterlo a las pruebas permitentes y hacer las modificaciones necesarias para cumplir con los requerimientos de las operadoras.

Se recomienda la inclusión de más opciones en el menú interactivo, como opciones para la configuración de la transmisión UART del microcontrolador, números alternativos de marcado en caso de alarma y cualquier otra que pudiera requerirse. Al sacar la imagen MPFS de la memoria interna de programación del microcontrolador se podrían configurar muchas más opciones que las que presenta el presente trabajo y al usar microcontroladores con mas entradas y salidas, se puede aumentar el número de puertos.

Para disminuir el tiempo de reconocimiento de los tonos DTMF se recomienda el uso de interrupciones en el PWM. Así, al estarce ejecutando la rutina de reproducción de audio se podrá calcular el algoritmo de Goertzel sin disminuir la calidad del audio y disminuyendo el tiempo para la detección de los tonos DTMF.

REFERENCIAS BIBLIOGRÁFICAS

- [1] UIT-T (Q.23:1993). Recomendación UIT-T Q.23 Unión Internacional de Telecomunicaciones. Suiza, 1988,1993 __4p.
- [2] UIT-T (E.180/Q35:1998). Recomendación UIT-T E.180/Q35 Unión Internacional de Telecomunicaciones. Suiza, 1998 __19p.
- [3] Características Técnicas de las Interfaces de TELEFONICA ESPAÑA, S.A.U: Versión 5ª. Red Telefónica Pública con Conmutación Interfaz de Línea Analógica. TELEFONICA España, 2004 __127p.
- [4] Chiouguey J. Chen. Modified Goertzel Algorithm in DTMF Detection Using the TMS320C80. Reporte de Aplicación. Texas Instrument, 1996 __19p.
- [5] Rodger Richey. Adaptive Differential Pulse Code Modulation Using PIC® Microcontrollers. Nota de Aplicación AN643. Microchip Technology Inc. 2007 __18p.
- [6] Hoja de Datos Técnicos. LM1575/LM2575/LM2575HV. National Semiconductor Corporation. EEUU, 2004 __28p.
- [7] Hoja de Datos Técnicos. MC34063A/E. STMicroelectronics. EEUU, 2004 __16p.
- [8] Hoja de Datos Técnicos. LT3010/LT3010-5. Linear Technology Corporation. EEUU, 2003 __18p.

[9] Hoja de Datos Técnicos. Si3035 rev 1.2. Silicon Laboratories. EEUU, 2000__54p.

[10] Hoja de Datos Técnicos. OPA342/OPA2342/OPA4342. Texas Instruments Incorporated. EEUU, 2005__12p.

[11] Hoja de Datos Técnicos. ADM3483/ADM3485/ADM3488/ADM3490/ADM3491 rev D. Analog Devices, Inc. EEUU, 2005-2010__20p.

[12] Hoja de Datos Técnicos – PIC24FJ64GA002. Microchip Technology Inc. EEUU, 2010__268p.

[13] UIT-T (Q.24:1993). Recomendación UIT-T Q.24 Unión Internacional de Telecomunicaciones. Suiza, 1988,1993__5p.

[14] Nilesh Rajbharti. The Microchip TCP/IP Stack, Nota de Aplicación AN833. Microchip Technology Inc. EEUU, 2008__94p.

[15] Guía de Usuario: Versión (1.2). MPLAB® C Compiler for PIC24 MCUs and dsPIC® DSCs. Microchip Technology Inc. EEUU, 2009__260p.

BIBLIOGRAFÍA

UIT-T (Q.23:1993). Recomendación UIT-T Q.23 Unión Internacional de Telecomunicaciones. Suiza, 1988,1993__4p.

UIT-T (Q.24:1993). Recomendación UIT-T Q.24 Unión Internacional de Telecomunicaciones. Suiza, 1988,1993__5p.

UIT-T (E.180/Q35:1998). Recomendación UIT-T E.180/Q35 Unión Internacional de Telecomunicaciones. Suiza, 1998__19p.

UIT-T (Q552:1994). Recomendación UIT-T Q552 Unión Internacional de Telecomunicaciones. Suiza, 1994__32p.

Características Técnicas de las Interfaces de TELEFONICA ESPAÑA, S.A.U: Versión 5ª. Red Telefónica Pública con Conmutación Interfaz de Línea Analógica. TELEFONICA España, 2004__127p.

Guía de Usuario: Versión (1.2). MPLAB® C Compiler for PIC24 MCUs and dsPIC® DSCs. Microchip Technology Inc. EEUU, 2009__260p.

Guía de Usuario: dsPIC® DSC DTMF DETECTION LIBRARY. Microchip Technology Inc. EEUU, 2010__36p.

Guía de Usuario: dsPIC® DSC DTMF Generation Library User's Guide. Microchip Technology Inc. EEUU, 2010__35p.

Guía de Usuario: dsPIC dsPICDEM.net™ 1 and dsPICDEM.net™ 2 Connectivity Development Board. Microchip Technology Inc. EEUU, 2004__114p.

Fotis E. Andritsopoulos, Newton Bomeisel Cardoso, Gregory A. Doumenis, Yannis M. Mitsos, Lambros E. Sarakis. An accurate Dual Tone Multiple Frequency Detector based on the low-complexity Goertzel algorithm. 2001__7p.

Matthew D. Felder, James C. Mason, and Brian L. Evans. Efficient Dual-Tone Multifrequency Detection Using the Nonuniform Discrete Fourier Transform. IEEE SIGNAL PROCESSING LETTERS, VOL. 5, NO. 7, JULY 1998, 4p.

Edxar. F. Velez. Detection of Multi-tone Signals Based on Energy Operators. Telecom Microelectronics Center, Northern Telecom Limited. Canada. 4p

Chiouguey J. Chen. Modified Goertzel Algorithm in DTMF Detection Using the TMS320C80. Reporte de Aplicación. Texas Instrument, 1996__19p.

Bonnie C. Baker. What Does “Rail-to-Rail” Operation Really Mean?. Nota de Diseño Analógico ADN009. Microchip Technology Inc. 2004__2p.

Amar Palacherla. Using PWM to Generate Analog Output. Nota de Aplicación AN538. Microchip Technology Inc. 1997__4p.

Sumit Mitra, Stan D’Souza, y Russ Cooper. Using the Analog-to-Digital (A/D) Converter. Nota de Aplicación AN546. Microchip Technology Inc. 1997__21p.

Kim Peck, Consultor. PICmicro™ Microcontroller Oscillator Design Guide. Nota de Aplicación AN588. Microchip Technology Inc. 1997__20p.

Rodger Richey. Low Power Design Using PICmicro™ Microcontrollers. Nota de Aplicación AN606. Microchip Technology Inc. 1997__16p.

Rodger Richey. Adaptive Differential Pulse Code Modulation Using PIC® Microcontrollers. Nota de Aplicación AN643. Microchip Technology Inc. 2007__18p.

Rob Stein y John Day. The D/A Conversion Using PWM and R-2R Ladders to Generate Sine and DTMF Waveforms. Nota de Aplicación AN655. Microchip Technology Inc. EEUU, 1997__19p.

Nilesh Rajbharti. The Microchip TCP/IP Stack. Nota de Aplicación AN833. Microchip Technology Inc. EEUU, 2008__94p.

David Hedley. Connecting a PICmicro® Microcontroller to a Standard Analog Telephone Line. Nota de Aplicación AN854. Microchip Technology Inc. EEUU, 2002__4p.

Ruan Lourens, Jose Benevides y Jonathan Dillon. Using the Microchip Ultra Low-Power Wake-Up Module. Nota de Aplicación AN879. Microchip Technology Inc. EEUU, 2008__10p.

Brant Ivey. nanoWatt and nanoWatt XLPT™ Technologies: An Introduction to Microchip's Low-Power Devices. Nota de Aplicación AN1267. Microchip Technology Inc. EEUU, 2009__10p.

Draw 150mW Of Isolated Power From Off-Hook Phone Line. Nota de Aplicación 1923. Maxim Integrated Products. EEUU, 1998__2p.

MODBUS Application Protocol Specification V1.1b. Modbus-IDA. December 28, 2006. 51p

Ian Pearson. Creating and Adding Sound Files to a PIC24F. Microchip Technology Inc. Europa, 2007__17p.

Hoja de Datos Técnicos. LM1575/LM2575/LM2575HV. National Semiconductor Corporation. EEUU, 2004__28p.

Hoja de Datos Técnicos. MC34063A/E. STMicroelectronics. EEUU, 2004__16p.

Hoja de Datos Técnicos. LT3010/LT3010-5. Linear Technology Corporation. EEUU, 2003__18p.

Hoja de Datos Técnicos. Si3035 rev 1.2. Silicon Laboratories. EEUU, 2000__54p.

Hoja de Datos Técnicos. OPA342/OPA2342/OPA4342. Texas Instruments Incorporated. EEUU, 2005__12p.

Hoja de Datos Técnicos. ADM3483/ADM3485/ADM3488/ADM3490/ ADM3491 rev D. Analog Devices, Inc. EEUU, 2005-2010__20p.

Hoja de Datos Técnicos. PIC24FJ64GA002. Microchip Technology Inc. EEUU, 2010__268p.

Hoja de Datos Técnicos. LM111/LM211/LM311 Voltage Comparator. National Semiconductor Corporation. EEUU, 1994__19p.

Hoja de Datos Técnicos. LM117/LM317A/LM317 3-Terminal Adjustable Regulator. National Semiconductor Corporation. EEUU, 1996__17p.

Hoja de Datos Técnicos. LM124/LM224/LM324/LM2902 Low Power Quad Operational Amplifiers. National Semiconductor Corporation. EEUU, 1994__17p.

Hoja de Datos Técnicos. LM34910C High Voltage (50V, 1.25A) Step Down Switching Regulator. National Semiconductor Corporation. EEUU, 2007__12p.

Hoja de Datos Técnicos. MCP1252/3. Microchip Technology Inc. EEUU, 2002__18p.

Hoja de Datos Técnicos. MCP1701A. Microchip Technology Inc. EEUU, 2007__22p.

Hoja de Datos Técnicos. MCP6021/1R/2/3/4. Microchip Technology Inc. EEUU, 2009__42p.

Hoja de Datos Técnicos. TC1014/TC1015/TC1185. Microchip Technology Inc. EEUU, 2007__22p.

Hoja de Datos Técnicos. TCM1705A. Texas Instruments Incorporated. EEUU, 1983__13p.

Hoja de Referencia. CHAPTER 2 PIC® Microcontroller Low Power Tips ‘n Tricks. Microchip Technology Inc. EEUU, 2009__21p.

GLOSARIO

Cadencia: serie de sonidos o movimientos que se suceden de un modo regular o medido.

Chip Enable: entrada que habilita el dispositivo.

DTMF (Dual-Tone Multi-Frequency): utilizado en los sistemas de telefonía actual de marcación por tonos.

EEPROM (Electrically-Erasable Programmable Read-Only Memory): es un tipo de memoria ROM que puede ser programado, borrado y reprogramado eléctricamente.

Firmware: es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria de tipo no volátil, que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo.

Full Duplex: cualidad de los elementos que permiten la entrada y salida de datos de forma simultánea.

I²C (I²C): es un bus de comunicaciones en serie, que utiliza dos líneas para transmitir la información: una para los datos y por otra la señal de reloj. También es necesaria una tercera línea, pero esta sólo es la referencia.

Low Dropout: Regulador de voltaje que trabaja con una pequeña diferencia entre la entrada y salida, generalmente menor a 0,8V.

PCB (Printed Circuit Board): tarjeta de circuito impreso.

PIC (Peripheral Interface Controller): son una familia de microcontroladores cuya principal característica es pertenecer al reglón de computadoras con un conjunto de instrucciones reducidas.

Pull Up: se refiere a la elevación de tensión a la salida de un circuito lógico sea por medio de una resistencia (pasivo) o un transistor (activo).

ROM (Read Only Memory): es un medio de almacenamiento utilizado en los dispositivos electrónicos. Los datos almacenados en la ROM no se pueden modificar (al menos no de manera rápida o fácil) y se utiliza principalmente para contener el firmware.

SD (Secure Digital): formato de tarjeta de memoria flash desarrollada de la memoria EEPROM que permite que múltiples posiciones de memoria sean escritas o borradas en una misma operación de programación mediante impulsos eléctricos, frente a las anteriores que sólo permite escribir o borrar una única celda cada vez.

SPI (Serial Peripheral Interface): es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.

Stand-by: es un estado del firmware donde se espera órdenes para establecer el funcionamiento completo o parcial del hardware asociado.

ANEXOS

Anexo N° 1. Código para la compresión ADPCM.

Anexo N°2. Código para la descompresión ADPCM.

Anexo N°3. Código para el cálculo del algoritmo de Groetzel.

Anexo N°4. Lista de palabras almacenadas en el microcontrolador.

Anexo N°5. Ejemplo de concatenación de palabras.

[ANEXO N° 1]

Código para la compresión ADPCM.

```
/* Tabla de índices de cambio */
const int IndexTable[16] = {
    0xff, 0xff, 0xff, 0xff, 2, 4, 6, 8,
    0xff, 0xff, 0xff, 0xff, 2, 4, 6, 8
};

/* Tabla de cuantificación de paso */
const long StepSizeTable[89] = {
    7, 8, 9, 10, 11, 12, 13, 14, 16, 17,
    19, 21, 23, 25, 28, 31, 34, 37, 41, 45,
    50, 55, 60, 66, 73, 80, 88, 97, 107, 118,
    130, 143, 157, 173, 190, 209, 230, 253, 279, 307,
    337, 371, 408, 449, 494, 544, 598, 658, 724, 796,
    876, 963, 1060, 1166, 1282, 1411, 1552, 1707, 1878, 2066,
    2272, 2499, 2749, 3024, 3327, 3660, 4026, 4428, 4871, 5358,
    5894, 6484, 7132, 7845, 8630, 9493, 10442, 11487, 12635, 13899,
    15289, 16818, 18500, 20350, 22385, 24623, 27086, 29794, 32767
};

signed long diff;          /* Diferencia entre la muestra y la muestra predicha */
long step;                /* cuantificación de paso */
signed long predsamp;     /* salida del predictor ADPCM */
signed long diffq;        /* Diferencia predicha */
int index;                /* Índice dentro de la tabla de cuantificación de pasos */
/*****
*   ADPCMEncoder - ADPCM encoder routine                               *
*****/
```

```

*   Input Variables:                                     *
*       signed long sample - 16-bit signed speech sample *
*   Return Variable:                                     *
*       char - 8-bit number containing the 4-bit ADPCM code *
*****/
char ADPCMEncoder( signed long sample )
{
int code;          /* ADPCM output value */
int tempstep;     /* Temporary step size */
/* Restore previous values of predicted sample and quantizer step size index*/
predsample = state.prevsample;
index = state.previndex;
step = StepSizeTable[index];
/* Compute the difference between the actual sample (sample) and the predicted
sample (predsample)*/
diff = sample - predsamp;
if(diff >= 0)
code = 0;
else
{
code = 8;
diff = -diff;
}
/* Quantize the difference into the 4-bit ADPCM code using the quantizer step size*/
tempstep = step;
if( diff >= tempstep )
{
code |= 4;
diff -= tempstep;
}
}

```

```

tempstep >>= 1;
if( diff >= tempstep )
{
code |= 2;
diff -= tempstep;
}
tempstep >>= 1;
if( diff >= tempstep )
code |= 1;
/* Inverse quantize the ADPCM code into a predicted difference using the quantizer
step size*/
diffq = step >> 3;
if( code & 4 )
diffq += step;
if( code & 2 )
diffq += step >> 1;
if( code & 1 )
diffq += step >> 2;
/* Fixed predictor computes new predicted sample by adding the old predicted
sample to predicted difference*/
if( code & 8 )
predsample -= diffq;
else
predsample += diffq;
/* Check for overflow of the new predicted sample*/
if( predsamp > 32767 )
predsample = 32767;
else if( predsamp < -32768 )
predsample = -32768;

```

```
/* Find new quantizer stepsize index by adding the old index to a table lookup using
the ADPCM code*/
index += IndexTable[code];
/* Check for overflow of the new quantizer step size index*/
if( index < 0 )
index = 0;
if( index > 88 )
index = 88;
/* Save the predicted sample and quantizer step size index for next iteration*/
state.prevsample = predsampl;
state.previndex = index;
/* Return the new ADPCM code */
return ( code & 0x0f );
}
```



```

*   Return Variable:
*   signed long - 16-bit signed speech sample
*****/
signed long ADPCMDecoder(char code )
{
/* Restore previous values of predicted sample and quantizer step size index*/
predsample = state.prevsample;
index = state.previndex;
/* Find quantizer step size from lookup table using index*/
step = StepSizeTable[index];
/* Inverse quantize the ADPCM code into a difference using the quantizer step size*/
diffq = step >> 3;
if( code & 4 )
diffq += step;
if( code & 2 )
diffq += step >> 1;
if( code & 1 )
diffq += step >> 2;
/* Add the difference to the predicted sample*/
if( code & 8 )
predsample -= diffq;
else
predsample += diffq;
/* Check for overflow of the new predicted sample*/
if( predsamp > 32767 )
predsample = 32767;
else if( predsamp < -32768 )
predsample = -32768;
/* Find new quantizer step size by adding the old index and a table lookup using the
ADPCM code*/

```

```
index += IndexTable[code];
/* Check for overflow of the new quantizer step size index*/
if( index < 0 )
index = 0;
if( index > 88 )
index = 88;
/* Save predicted sample and quantizer step size index for next iteration*/
state.prevsample = predsampl;
state.previndex = index;
/* Return the new speech sample */
return( predsampl );
}
```

[ANEXO N° 3]

Código para el cálculo del algoritmo de Goertzel.

```
if(sample_complete == true && Sleep == 0)
{
    for(i = 0; i < MAX_BINS; i++)
    {
        prev2[i] = prev1[i] = 0.0;
    }
for(sample_count=0; sample_count < GOERTZEL_N; sample_count++)
{
    //Goertzel Algorithm
    for ( i = 0; i < MAX_BINS; i++ )
    {
val = coeffs[i] * prev1[i] - prev2[i] + (double)samples[sample_count];
        prev2[i] = prev1[i];
        prev1[i] = val;
    }
}
    for ( i=0; i<MAX_BINS; i++ ) {
magnitude[i] = (prev1[i] * prev1[i]) + (prev2[i] * prev2[i]) - (coeffs[i] * prev1[i] *
prev2[i]);
    }
sample_complete = false;
sample_count = 0;
IEC1bits.T5IE = 1;      // enable TMR0 interrupt, start storing samples
```

[ANEXO N° 4]

Lista de palabras almacenadas en el microcontrolador

A: Bienvenido	Y: Mayor
B: Marque	@: Tono de silencio de 0,25 segundos
C: Marque	de duración
D: Analógico	0: Cero
E: Digital	1: Uno
F: Mod-Bus	2: Dos
G: Configuración	3: Tres
H: Leer	4: Cuatro
I: Valor	5: Cinco
J: Puerto	6: Seis
K: Número	7: Siete
L: Igual	8: Ocho
M: Regresar	9: Nueve
N: Asterisco	D0: Tono DTMF 0
O: Numeral	d1: Tono DTMF 1
P: Función	d2: Tono DTMF 2
Q: Teléfono	d3: Tono DTMF 3
R: Dispositivo	d4: Tono DTMF 4
S: Alarma	d5: Tono DTMF 5
T: Escribir	d6: Tono DTMF 6
U: Activar	d7: Tono DTMF 7
V: Des	d8: Tono DTMF 8
W: Activado	d9: Tono DTMF 9
X: Menor	

[ANEXO N° 5]

Ejemplo de concatenación de palabras

```
unsigned char const cont1[] = {'@','B','1','C','D','E','@',      //MENU  PRINCIPAL
                               'B','2','C','J','E','@',
                               'B','3','C','J','F','@',
                               'B','4','C','G','@',0};

filename[0] = cont1[ic];

ic++;
if(cont1[ic]==0)
{
    ic=0;
}

filename[1] = '.';
filename[2] = 'd';
filename[3] = 'a';
filename[4] = 't';
filename[5] = 0;
hFile = MPFSOpen(filename);    // Open file
PlayClip(hFile);              // Play audio clip
for(time=0;time<65534;time++); // Delay
```