

Desarrollo de equipo para el manejo de
actuadores por línea telefónica

TRABAJO FINAL DE INGENIERÍA
ELECTRÓNICA

AUTOR: Tomás ASTIZ

DIRECTOR: Ing. Miguel RABINI

CODIRECTOR: Dr. Ing. Carlos Arturo GAYOSO

Laboratorio de Componentes Electrónicos, Facultad de Ingeniería, UNMdP

27 de Marzo de 2015





RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Resumen

El presente informe, correspondiente al trabajo final de la carrera de ingeniería electrónica, trata sobre el diseño de un equipo para el manejo de actuadores, en este caso relays, utilizados para accionar distintos artefactos del hogar (luminarias, máquina de lavar, calefacción, alarma, etc.).

El comando de este equipo se realiza a distancia discando el número telefónico de la casa donde está ubicado. Se basa en la detección de tonos multifrecuentes en la línea telefónica. Se deben ingresar los códigos, utilizando el teclado telefónico, para acceder y operar los distintos actuadores. El sistema posee una plataforma vocal que dialoga con el usuario durante esta secuencia.

El dispositivo permite al interesado programar una clave numérica de cuatro dígitos para acceder al sistema, la cual podrá ser modificada todas las veces que desee y quedará registrada aún con cortes de alimentación. Del mismo modo el equipo permite la programación de la cantidad de tonos de llamada telefónicos necesarios para que la línea sea atendida. Además cuenta con un display LCD que actúa como interface con el operador durante la programación.

El núcleo del sistema está constituido por un microcontrolador.

Se armó el prototipo completo en un circuito de prueba, sobre el cual se realizó una demostración de funcionamiento.

Para el diseño del dispositivo se adoptó la siguiente metodología:

- Se estudiaron las señales presentes en la línea telefónica, lo que permitió obtener el conocimiento necesario para diseñar los distintos bloques del sistema.
- Se recopiló información sobre detectores de tonos multifrecuentes y memorias reproductoras de voz, con el fin de elegir los más apropiados. Se realizó el ensayo de los mismos.
- Se diseñó un detector de corriente de llamada que permite al microcontrolador contar la cantidad de pulsos de llamada entrantes a la línea para producir su toma.
- Se seleccionó un display LCD que cumpla los requisitos necesarios para la programación del dispositivo de manera clara y concisa para el usuario, así como también mostrar información acerca del estado de los relays.
- Se seleccionó el microcontrolador adecuado para el procesamiento de los datos de todos los bloques del sistema.
- Se desarrolló el software necesario para el correcto funcionamiento de cada uno de los bloques que componen el equipo, teniendo en consideración la simpleza y la legibilidad para el manejo por parte del usuario. Para este diseño fue necesario el aprendizaje de nuevos programas de desarrollo, lenguajes de programación, compiladores y programadores desconocidos.
- Por último se implementó el prototipo en un circuito de prueba, se solucionaron los problemas que se presentaron y se verificó su correcto funcionamiento.

Índice

Resumen	2
Capítulo 1 Introducción: Señales de la línea	5
1.1. Teléfono	5
1.2. Circuito de conversación: la híbrida telefónica	5
1.3. Transferencia de señal desde el micrófono a la línea	6
1.4. Transferencia de señal desde la línea al auricular.....	6
1.5. Marcación por tonos multifrecuentes	6
1.6. Evolución de la tensión en la línea en el curso de una llamada saliente	7
1.7. Evolución de la tensión en la línea en el curso de una llamada entrante.....	7
1.8. Los tonos PTT.....	7
Capítulo 2 Anteproyecto	8
2.1. Diagrama en bloques.....	8
2.2. Detector de tonos multifrecuentes	9
2.3. Plataforma vocal	9
2.4. Display LCD.....	10
2.5. Detector de corriente de llamada.....	10
2.6. Actuadores.....	10
2.7. Microcontrolador	11
Capítulo 3 Proyecto: Diseño y análisis de los bloques	12
3.1. Detector de tonos multifrecuentes (DTMF)	12
3.2. Plataforma vocal	14
3.3. Detector de corriente de llamada.....	17
3.4. Toma de la línea telefónica.....	20
3.5. Buffers	20
3.6. Display LCD.....	21
3.7. Microcontrolador	24
3.7.1. Esquema general del programa.....	25
3.7.2. Detector de rings	27
3.7.3. Ingreso de dígitos	29
3.7.4. Ingreso de la contraseña.....	30
3.7.5. Ingreso de los códigos.....	30
3.7.6. Modulo de cambio de contraseña y cantidad de rings	30

3.7.7.	Modulo de reproducción de audio.....	35
3.7.8.	Modulo para manejo de display	35
3.7.9.	Módulo de presentación	35
Capítulo 4	Manual de mantenimiento	37
4.1.	Especificaciones	37
4.2.	Diagrama en Bloques	37
4.2.1.	Detector de tonos multifrecuentes	38
4.2.2.	Plataforma vocal	38
4.2.3.	Detector de corriente de llamada	38
4.2.4.	Actuadores.....	39
4.2.5.	Display.....	39
4.2.6	Microcontrolador	39
4.3.	Circuito completo	39
4.4.	Diagrama de flujo.....	39
4.5.	Procedimiento de localización de fallas	39
Capítulo 5	Manual de operación	44
	Uso del manual.....	44
	Instalación.....	44
	Inserción la fuente de alimentación y el cable telefónico	45
	Cómo comenzar	46
	Programar la cantidad de tonos de llamada.	46
	Programar la clave	46
	Presentación de la pantalla.....	47
	Funcionamiento	47
	Manipulación remota del sistema	47
	Información sobre seguridad	48
	Conclusiones y recomendaciones finales	49
	Apéndice A.....	50
	Referencias	73

Capítulo 1

Introducción: Señales de la línea

En el presente capítulo se realiza un estudio de las señales presentes en la línea telefónica. La comprensión del funcionamiento del sistema telefónico fue fundamental para la realización de los distintos bloques del sistema, los cuales interactúan directamente con la línea, tanto extrayendo como inyectando datos a la misma. Fue necesario conocer los niveles y características de las señales para realizar la síntesis de los bloques y para resolver los problemas que pudieron presentarse en los mismos.

1.1. Teléfono

El teléfono es un dispositivo de telecomunicación diseñado para transmitir señales acústicas a distancia por medio de señales eléctricas.

El teléfono convencional está formado por dos circuitos que funcionan juntos: el circuito de conversación, que es la parte analógica, y el circuito de marcación, que se encarga de la marcación y llamada. Tanto las señales de voz como las de marcación y llamada (señalización), así como la alimentación, comparten el mismo par de hilos.

La impedancia característica de la línea es 600Ω . Las señales procedentes del teléfono hacia la central y las que se dirigen a él desde ella viajan por la misma línea de sólo dos hilos. Para poder combinar en una misma línea dos señales (ondas electromagnéticas) que viajen en sentidos opuestos y para luego poder separarlas se utiliza un dispositivo llamado transformador híbrido o bobina híbrida, que no es más que un acoplador de potencia (duplexor) [1].

1.2. Circuito de conversación: la híbrida telefónica

El circuito de conversación consta de cuatro componentes principales: la bobina híbrida; el auricular; el micrófono y una impedancia de 600Ω , para equilibrar la híbrida. Ésta consiste en un transformador con tres bobinados, L_1 , L_2 y L_3 , según se muestra en la figura 1.1 [2].

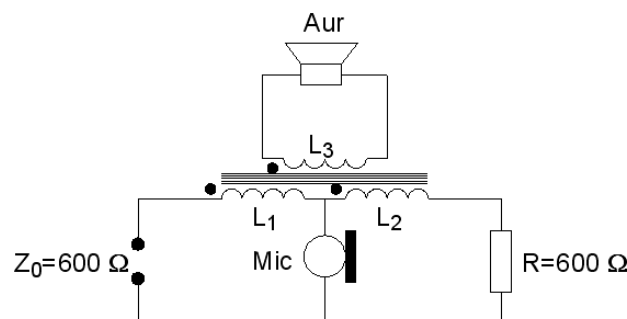


Figura 1.1. Circuito de conversación.

1.3. Transferencia de señal desde el micrófono a la línea

La señal que se origina en el micrófono se reparte entre L_1 y L_2 en partes iguales. La primera va a la línea y la segunda se pierde en la carga de 600Ω , pero L_1 y L_2 inducen corrientes iguales y de sentido contrario en L_3 , que se cancelan entre sí, evitando que la señal del micrófono alcance el auricular. En la práctica la impedancia de la carga no es exactamente igual a la impedancia de la línea, por lo que las corrientes inducidas en L_3 no se anulan completamente. Esto tiene un efecto útil, parte de la señal generada en el micrófono se escuche también en el auricular local (efecto "side tone"), lo que permite que quién habla se escuche asimismo percibiendo que el sistema está funcionando [2].

1.4. Transferencia de señal desde la línea al auricular

La señal que viene por la línea provoca la circulación de corrientes tanto por L_1 como por L_2 . Estas corrientes se inducen en L_3 , sumándose, excitando el auricular. Si bien la señal que viene por la línea provoca la circulación de una pequeña corriente por el micrófono, este hecho no afecta la conversación telefónica.

En el circuito de conversación real se añade un varistor a la entrada, para mantener la polarización del micrófono a un nivel constante, independientemente de lo lejos que esté la central local y se mejora el efecto "side tone", conectando el auricular a la impedancia de carga, para que el usuario tenga una pequeña realimentación y pueda oír lo que dice (figura 1.2).

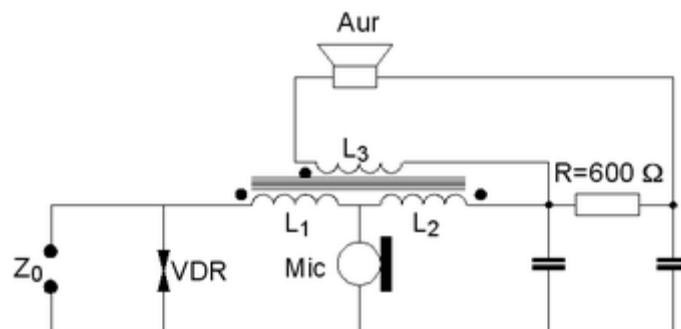


Figura 1.2. Circuito de conversación real.

En la actualidad los terminales telefónicos son construidos con híbridas de estado sólido y no en base al transformador multibobinado indicado anteriormente. Las híbridas de estado sólido, tienen una respuesta de frecuencia más plana ya que no usan bobinados. Los bobinados introducen distorsión al atenuar mucho más las señales de alta frecuencia que las de baja frecuencia [2].

1.5. Marcación por tonos multifrecuentes

Al ser pulsada en el teléfono la tecla correspondiente al dígito que se quiere marcar, se envían dos tonos, de distinta frecuencia: uno por columna y otro por fila en la que esté la tecla, que la central decodifica a través de filtros especiales, detectando qué dígito se marcó.

La tabla 1.1 que se muestra a continuación indica las frecuencias de los tonos por cada fila y columna del teclado de los teléfonos de marcación por tono [1].

	1209Hz	1336Hz	1477Hz	1633Hz
697Hz	1	2	3	A
770Hz	4	5	6	B
852Hz	7	8	9	C
941Hz	*	0	#	D

Tabla 1.1. Tonos multifrecuentes

1.6. Evolución de la tensión en la línea en el curso de una llamada saliente

En reposo, la línea presenta una tensión continua de 48 V con una polaridad determinada. La línea queda ocupada cuando se descuelga el teléfono, lo que equivale a demandar de ella un consumo superior a 5 mA a la tensión de 48 V. En ese instante, la tensión cae a un valor comprendido entre 6 y 18 V (de acuerdo a la resistencia del par telefónico entre la central y el usuario). Al colgar el teléfono, la tensión en la línea retorna al valor de 48 V quedando de nuevo la línea en reposo [3].

1.7. Evolución de la tensión en la línea en el curso de una llamada entrante

Cuando se recibe una llamada, aparece en la línea una tensión alterna de 80 V eficaces con una frecuencia comprendida entre 16 y 25 Hz que se superpone a los 48 V que hay en reposo. Esta señal es la que provoca que suene el timbre del teléfono. Cuando se descuelga el teléfono, la tensión de línea de 48 V de continua cae a un valor comprendido entre 6 y 18 V como ocurre en una llamada saliente, la central telefónica reconoce este evento y corta la corriente de llamada [3].

1.8. Los tonos PTT

Cuando se habla de tono PTT (Portes, Télégraphes, Téléphones) se hace referencia a la nota musical LA que aparece cuando se descuelga un teléfono. Ésta es una señal sinusoidal de frecuencia 440 Hz. Todas las señales de audio presentes en la línea durante la realización de una llamada son de esta naturaleza.

Cuando se descuelga un teléfono, se oye una señal permanente de 440 Hz. Se trata del tono de invitación a discar, indica que la central espera el discado de un número y desaparece después de cualquier manipulación sobre el teclado. Si el teléfono queda descolgado más de 20 segundos sin que se marque ningún número, este tono se reemplaza por el tono de ocupación, ya que la línea pasa a estado de falsa llamada.

Este tono de ocupación, aparece también cuando se intenta llamar a un abonado que no está libre o cuando uno de los interlocutores cuelga. Su significado es simplemente una invitación a colgar. Se trata de una señal de 440 Hz alternando 500 ms de actividad y 500 ms de silencio, es decir, con una cadencia de 1 Hz

Así mismo, después de realizar el discado de un número, y siempre que la línea del destinatario esté libre, se oye el tono de retorno de llamada. Al igual que las otras señales, es una señal de 440 Hz que alterna 1 segundo de actividad y 3 segundos de silencio. Aunque esta señal no es sincrónica con la del timbre del destinatario, sigue el mismo ritmo [3].

Capítulo 2

Anteproyecto

En este capítulo se detalla el análisis de soluciones alternativas a nivel de diagrama en bloques, de confiabilidad, costos, reproductibilidad, etc, y se justifican las decisiones tomadas.

2.1. Diagrama en bloques

El sistema consta de los bloques principales que se muestran en la figura 2.1.

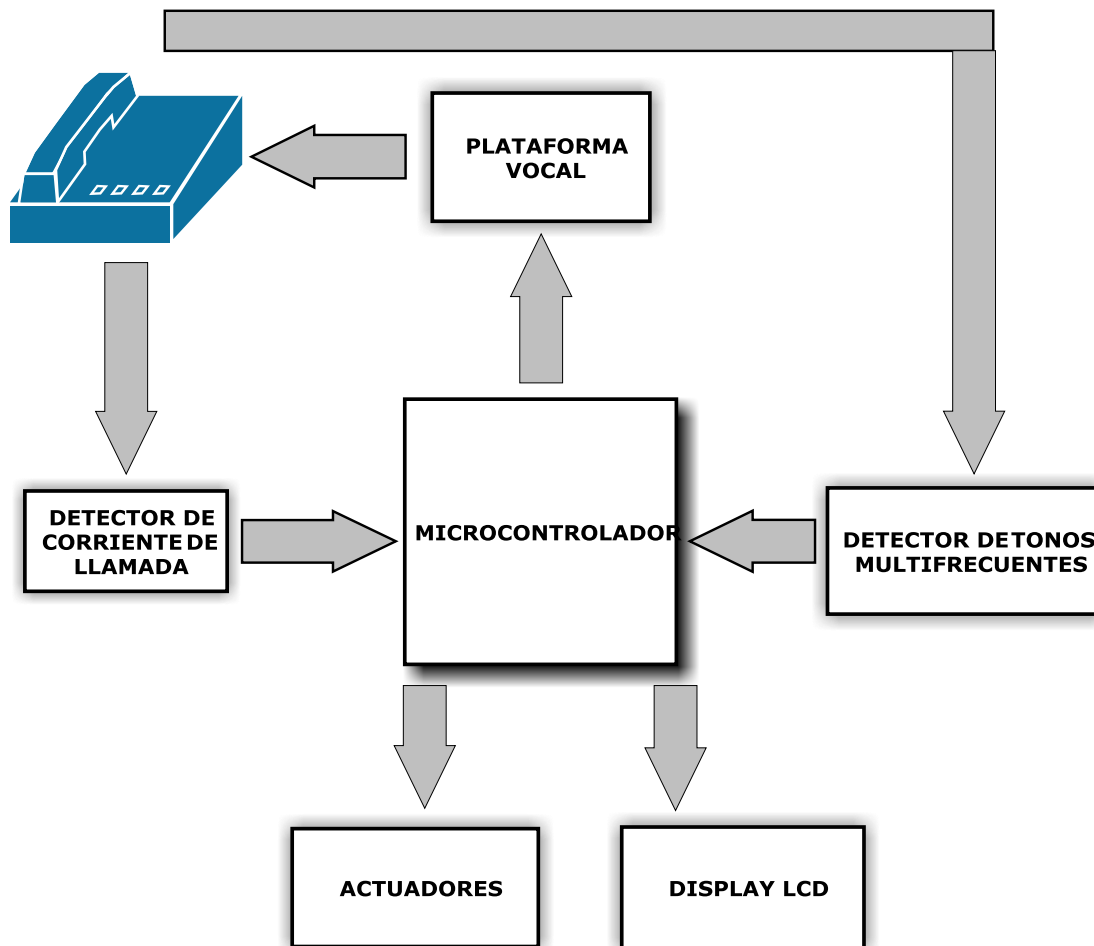


Figura 2.1. Diagrama en bloques

Se analizará cada bloque por separado detallándose las decisiones tomadas.

2.2. Detector de tonos multifrecuentes

Se necesitaba que el usuario pudiera ingresar códigos a través de la línea telefónica y que los mismos sean interpretados por el microcontrolador. Al ser pulsada en el teléfono la tecla correspondiente al dígito que se quiere marcar, se envían dos tonos, de distinta frecuencia. Esta información no puede ser interpretada por el microcontrolador que solo puede entender datos binarios. Por lo tanto se necesitó un dispositivo que pudiera transformar estos tonos duales en un código binario.

La solución a esta problemática se resolvió a través de un detector de tonos multifrecuentes. Estos detectores emplean filtros capacitivos conmutados de alta precisión para separar las señales multifrecuentes en los dos tonos que la conforman. Luego mediante técnicas de conteo digitales detectan y decodifican los 16 pares de tonos multifrecuentes cuyo resultado es una salida codificada de 4 bits.

Además incluyen un rechazo de tono de discar integrado para eliminar la necesidad de un prefiltado [4].

Se investigaron las distintas posibilidades y teniendo en cuenta disponibilidad y costos se seleccionaron dos integrados comerciales que podían realizar esta función, el HT9170 y el HT8870. Los dos integrados poseen similar funcionamiento, por disponibilidad local se optó por el HT9170.

2.3. Plataforma vocal

La plataforma vocal debe guiar al usuario en los pasos a seguir en el manejo del sistema. En este caso se analizaron distintas soluciones posibles para implementar este bloque.

Una posible solución era utilizar una memoria externa donde se pudieran grabar los mensajes, y luego mediante un microcontrolador que tuviera un convertor digital-analógico de salida, transformar el mensaje almacenado en código binario en un mensaje analógico que pudiera ser reproducido por la bocina del teléfono. La dificultad es que la mayoría de los microcontroladores no poseen el convertor mencionado, que los mensajes a almacenar eran pocos y de corta duración y además, se debe contar con hardware adicional necesario para trabajar con audio (amplificadores, osciladores, control de ganancia, filtros, memoria externa, etc).

Esta opción es más adecuada cuando se quiere realizar tratamiento digital de audio, con mensajes de larga duración que justifiquen el uso del hardware mencionado anteriormente.

Se investigó sobre el tema y se hallaron integrados de la línea ISD, los cuales proporcionan grabación y reproducción de mensajes de corta duración con alta calidad y en un sólo chip. Se trata de dispositivos CMOS que incluyen un oscilador integrado, un preamplificador de micrófono, control automático de ganancia, filtro anti-aliasing y amplificador de altavoz. Las grabaciones son almacenadas en celdas de memoria no volátiles integradas [5].

Estos integrados cumplían todos los requisitos necesarios para la implementación de la plataforma vocal en un sólo chip (mayor confiabilidad), de manera sencilla, sin utilizar hardware adicional, para mensajes de corta duración y a un precio de mercado menor que la primera opción, por lo que se decidió su implementación en el sistema.

Se utilizó el integrado ISD1420, el cual provee un tiempo de grabación de veinte segundos que se puede dividir en distintos mensajes de duración menor. En este caso se utilizaron seis mensajes de tres segundos aproximadamente.

2.4. Display LCD

Se empleó un display LCD para permitir al usuario programar la contraseña, la cantidad de tonos de llamado para que el sistema atienda la línea, y para mostrar el estado de los actuadores cuando el sistema se encuentra en reposo. Se debió definir la cantidad de líneas y caracteres por línea del display.

Para que los mensajes sean claros para el usuario se decidió utilizar un display de dos líneas y dieciséis caracteres en cada una de ellas.

Se adquirió el display WH1602B de la marca Winstar el cual cumple los requisitos mencionados [6].

2.5. Detector de corriente de llamada

Para dotar al equipo telefónico de la capacidad de contestar automáticamente una llamada, fue necesario adaptar la señal de 80 V eficaces, 16/25 Hz (corriente de llamada) presente en la línea a otra que pueda ser leída por un circuito digital (microcontrolador).

Para diseñar el detector de corriente de llamada, como se explicará en secciones posteriores, se utilizó un puente de diodos y un optoacoplador.

Se realizó la implementación del puente de diodos mediante el integrado DB157 y no con 4 diodos, de esta manera se aumenta la confiabilidad del sistema al utilizar menor cantidad de componentes, además de ahorrar espacio en el impreso.

El integrado DB157 soporta una tensión RMS máxima de 700 V, mucho mayor a la presente en la línea [7].

Para el optoacoplador se utilizó el integrado 4N27, necesario para aislar el microcontrolador de la línea telefónica y de esta manera protegerlo de sobretensiones que puedan provenir de la misma [8].

2.6. Actuadores

Los actuadores se implementaron mediante relays. Para no utilizar una fuente de tensión adicional se utilizaron mini relays para circuito impreso con tensión de devanado de 5 volts (como el resto de los bloques) y con un consumo aproximado de 60 mA en estado activo.

Estos relays en su versión más simple poseen dos contactos inversores.

Para alimentar el devanado de los relay se necesitaron buffers de corriente, debido a que el microcontrolador no es capaz de entregar 60mA en sus salidas. Para resolver este problema los buffers se podía implementar ya sea con transistores (utilizándolos como llaves) o con integrados especiales ideados para esta tarea específica.

En el caso de utilizar transistores se debían emplear resistencias de polarización y además, diodos en paralelo al devanado para absorber los picos de corriente en la comutación de carga inductiva. Esta solución sumaba una larga cantidad de componentes al circuito, lo cual provocaba que la confiabilidad del mismo bajara y aumentara el tamaño del circuito impreso además del costo.

Por lo tanto, se optó por la segunda alternativa, se utilizó el integrado ULN2803 que corresponde a un arreglo de ocho transistores Darlington de alta corriente y alta tensión.

Los ocho transistores Darlington NPN de este integrado están especialmente diseñados para la interconexión entre circuitos digitales (como TTL o CMOS) y relays con requerimientos de tensión/corriente más altos. El colector del par Darlington de cada buffer posee un diodo rápido conectado disponible para el usuario [9].

2.7. Microcontrolador

Por último se tomó la decisión acerca del microcontrolador. Por recomendación del director de tesis, la disponibilidad de un programador y el deseo de aprender a programar otro tipo de microcontroladores distintos a los ya estudiados en la carrera (Freescale), se optó por utilizar la línea Microchip.

Analizando la cantidad de entradas y salidas necesarias se llegó a la conclusión que se necesitarían aproximadamente 25, además considerando que los demás bloques del sistema están diseñados para trabajar con 5 volt y teniendo en cuenta la disponibilidad de este tipo de integrados en el mercado, se abocó a la búsqueda de un dispositivo con las características mencionadas.

Se seleccionó el microcontrolador PIC 18F4550 de 8 bits cuyas principales características son las siguientes [10]:

- 32Kbytes de memoria flash.
- 2048 bytes de memoria RAM.
- 256 bytes de memoria EEPROM
- 35 entradas/salidas.
- Tensión de operación 5 volt.
- Encapsulado DIP de 40 patas.

Para la programación del microcontrolador se utilizó el programa MPLAB X IDE, el cual es un software usado para aplicaciones de desarrollo de microcontroladores de Microchip. Esta herramienta es llamada ambiente de desarrollo integrado (IDE en inglés) porque provee múltiples funciones para el desarrollo de proyectos, tales como programación del dispositivo, modo de debugger, ciclo de desarrollo, herramientas para el manejo total del proyecto, etc [11].

El programa se escribió en el lenguaje de programación ANSI C, para lo cual se necesitó del compilador provisto por Microchip, XC8 que soporta microcontroladores PIC de 8 bits, PIC18 en este caso.

Un compilador es un programa que transforma uno o varios archivos de código fuente y genera un archivo en código de máquina llamado ejecutable, este nuevo archivo es enviado al microcontrolador para que lo ejecute [12].

Se utilizó el lenguaje de programación C y no ensamblador por las siguientes razones [13]:

- Es un lenguaje universal: mientras el lenguaje ensamblador es exclusivo para determinada máquina, marca y arquitectura, el lenguaje C es un lenguaje universal y estandarizado. La importancia de esta razón en el proyecto es que utilizando lenguaje C se pudo consultar fuentes bibliográficas para encontrar información sobre la programación.
- Es una herramienta rápida de programación: la programación en C, al ser un lenguaje de alto nivel permite soluciones más simples a nivel de código y más legibilidad del programa.

Para la programación del microcontrolador se utilizó el programador/debugger PICkit 3 [14] debido a la disponibilidad del mismo.

Capítulo 3

Proyecto: Diseño y análisis de los bloques

3.1. Detector de tonos multifrecuentes (DTMF)

Para la detección de los tonos multifrecuentes se utilizó el integrado HT9170 que corresponde a un receptor integrado de tonos duales multifrecuentes [4].

El diagrama en bloques del circuito integrado se muestra en la figura 3.1.

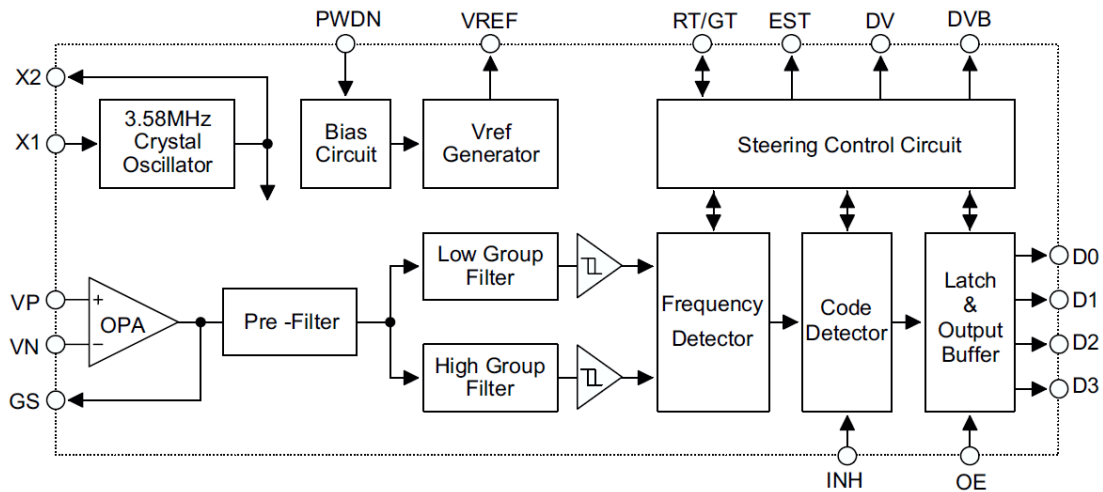


Figura 3.1. Diagrama en bloques del HT9170

Los pines VP y VN corresponden a las entradas inversora y no inversora del amplificador operacional, respectivamente, mientras que el terminal GS corresponde a su salida. Este amplificador operacional se utiliza para ajustar la señal de entrada.

La conexión de los dos hilos correspondientes a la línea telefónica se realizó mediante una entrada diferencial (línea balanceada), debido a que previamente se ensayó el circuito con una entrada común que resultó ruidosa. Se muestra el esquemático resultante en la figura 3.2.

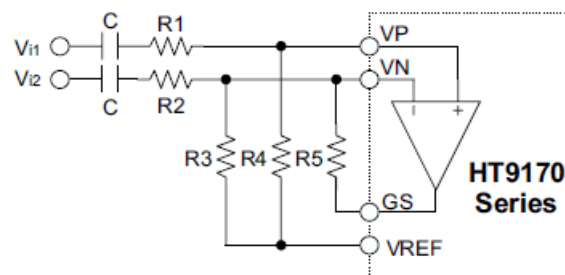


Figura 3.2. Circuito de entrada al HT9170

Los capacitores C, que bloquean el pasaje de corriente continua, son de 1µf y las resistencias R₁, R₂ y R₅, que adecuan el nivel de la señal de entrada, son de 100KΩ, valores obtenidos de la hoja de datos. La resistencia R₄ debe ser igual al paralelo entre las resistencias R₃ y R₅ para que la entrada sea balanceada y así obtener un alto rechazo a modo común.

Por lo tanto:

$$R_4 = \frac{R_3 \cdot R_5}{R_3 + R_5}$$

Resultado R₄ = 37,5KΩ y R₃ = 60KΩ, valores recomendados en la hoja de datos.

Para utilizar valores comerciales se adoptó R₄ = 39KΩ y R₃ = 68KΩ.

Se empleó un cristal de 3,58MHz especificado en la hoja de datos del integrado. Se dispuso entre las pines X₁ y X₂ (7 y 8, respectivamente).

El terminal PWDN (6) (lógica positiva) apaga el dispositivo y deshabilita el oscilador, como este pin esta internamente conectado no se realizo conexión en el mismo.

El circuito final responde al esquema presentado en la figura 3.3.

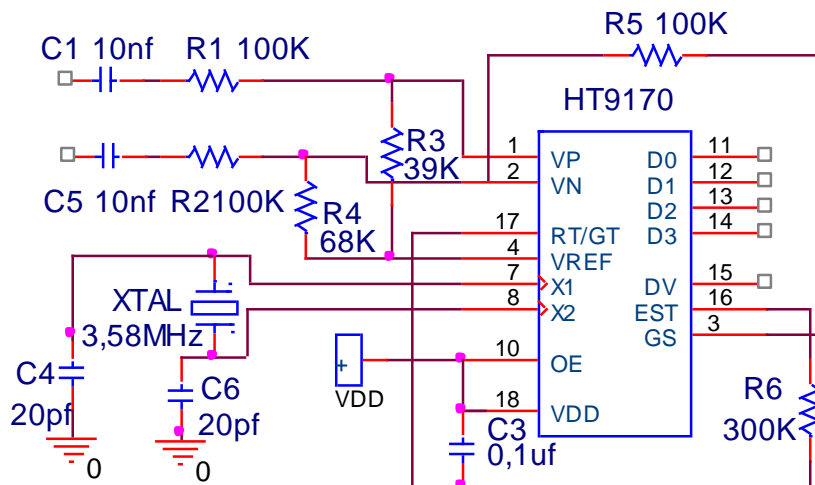


Figura 3.3. Circuito detector de tonos multifrecuentes.

Los pines D0-D1-D2-D3 son los terminales de salida de los datos, su salida corresponde a un código binario que está correlacionado con los tonos multifrecuentes entrantes por la línea telefónica según la tabla 3.1.

El pin OE (10), corresponde a una habilitación de las salidas D0 a D3, por eso se lo dispuso a Vcc, de esta manera los pines de salida están siempre activos. Las salida de datos (D0-D3) son salidas tristate. Cuando la entrada OE es cero, la salida de datos está en alta impedancia.

Cuando se ingresa la señal multifrecuente presionando un dígito, la salida DV (15) se pone en alto y se transfiere el código DTMF a la salida. Mientras que el dígito está presionado DV permanece en alto, sólo pasando a bajo cuando deja de apretarse el botón, sin embargo, las salidas D0-D3 quedan seteadas hasta que se ingresa un nuevo dígito.

DTMF data output table

Low Group (Hz)	High Group (Hz)	Digit	OE	D3	D2	D1	D0
697	1209	1	H	L	L	L	H
697	1336	2	H	L	L	H	L
697	1477	3	H	L	L	H	H
770	1209	4	H	L	H	L	L
770	1336	5	H	L	H	L	H
770	1477	6	H	L	H	H	L
852	1209	7	H	L	H	H	H
852	1336	8	H	H	L	L	L
852	1477	9	H	H	L	L	H
941	1336	0	H	H	L	H	L
941	1209	*	H	H	L	H	H
941	1477	#	H	H	H	L	L
697	1633	A	H	H	H	L	H
770	1633	B	H	H	H	H	L
852	1633	C	H	H	H	H	H
941	1633	D	H	L	L	L	L
—	—	ANY	L	Z	Z	Z	Z

Z: High impedance

Tabla 3.1. Código binario de los tonos multifrecuentes.

3.2. Plataforma vocal

Para diseñar la plataforma vocal se utilizó el integrado ISD1420 [5]. El diagrama en bloques del integrado se muestra en la figura 3.4.

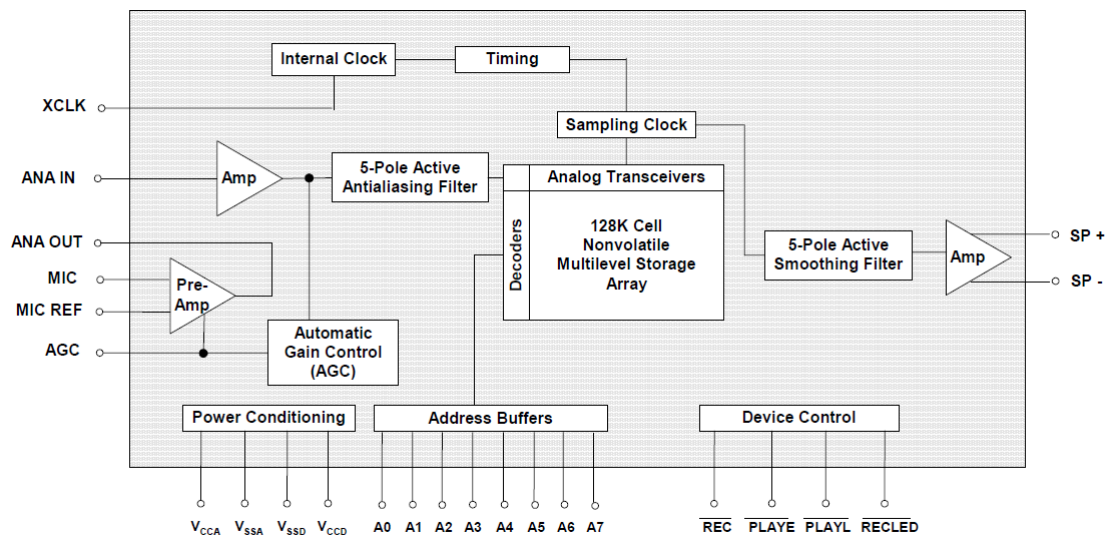


Figura 3.4. Diagrama en bloques del ISD1420

- A0-A7 (1-6, 9,10) tienen dos funciones que dependen del nivel de los dos bits más significativos de la dirección. Si cualquiera de los dos (A6 o A7) o ambos bits están en bajo, las entradas son interpretadas como bits de dirección y son usadas como la dirección inicial para el ciclo de grabación/reproducción actual. Si tanto A6 como A7 se encuentran en alto, entonces el dispositivo entra en un modo de operación especial, el cual no se utilizó para este proyecto.
- V_{SSD} , V_{SSA} (12,13) y V_{CCA} , V_{CCD} (16, 28) corresponden a la alimentación del integrado. Los circuitos internos analógicos y digitales del ISD utilizan buses de alimentación y tierra separados para minimizar el ruido. Tanto V_{SSD} y V_{SSA} como V_{CCA} , V_{CCD} se conectaron juntos respectivamente, lo más cerca posible al dispositivo, como recomendación de la hoja de datos para su correcto funcionamiento. Se utilizó una fuente de alimentación de 5V.
- SP+, SP- (14,15) proveen conexión directa con el altavoz, el cual debe poseer una impedancia de 16Ω . No se requiere capacitor de acoplamiento. Las salidas del altavoz se encuentran en estado de alta impedancia durante el ciclo de grabación.
- MIC (17), corresponde a la entrada de micrófono, la cual transfiere su señal al amplificador integrado. El control automático de ganancia (AGC) integrado controla la ganancia de este preamplificador de -15 a 24 dB. El micrófono externo debe estar acoplado en corriente alterna con este pin mediante un capacitor en serie. El valor de este capacitor, junto con la resistencia interna de $10K\Omega$ determinan la frecuencia de corte en baja del filtro pasabanda del ISD.
- MIC REF (18) corresponde a la entrada no inversora al preamplificador de micrófono. Provee cancelación de ruido y rechazo en modo común.
- AGC (19), el control automático de ganancia ajusta la ganancia del preamplificador para compensar los distintos niveles de la entrada de micrófono. El tiempo de “ataque” está determinado por la constante de tiempo dada por una resistencia interna de $5K\Omega$ y el capacitor externo C_6 , conectado entre este pin y masa, como se muestra en la figura 3.5. El tiempo de “liberación” se determina por la constante de tiempo correspondiente a una resistencia externa R_5 y al capacitor externo C_6 conectados en paralelo entre el pin AGC y masa. Los valores utilizados, fueron los sugeridos por la hoja de datos $470K\Omega$ y $4,7\mu f$.
- ANA IN (20), el pin ANA OUT debe conectarse a través de un capacitor al pin ANA IN. El valor de este capacitor junto con la impedancia de entrada de $3K\Omega$ del pin ANA IN, se selecciona para dar una frecuencia de corte en baja del filtro pasabanda de voz. En este caso se utilizó un capacitor C_3 de $0,1\mu f$ y una resistencia adicional de $5,1K\Omega$ según se aconseja en la hoja de datos.
- ANA OUT (21), provee la salida del preamplificador.
- \overline{PLAYL} (23), activado por nivel, cuando esta señal de entrada se mantiene baja, se inicia un ciclo de reproducción, que continua hasta que la línea se pone en alto, o hasta que se detecta una marca OEM (fin de mensaje). El dispositivo se apaga automáticamente y entra en modo stand-by luego que se completa el ciclo de reproducción.

- \overline{PLAYE} (24), activado por flanco. Cuando se detecta una transición a bajo en este pin comienza el ciclo de reproducción. Poner en alto \overline{PLAYE} durante este ciclo no provoca la terminación del mismo, que continua hasta que se detecta la marca OEM.
- \overline{RECLEd} (25), permanece en bajo durante un ciclo de grabación. Se utilizó para manejar un LED que indicó que el ciclo de grabado estaba en proceso. Adicionalmente, titila cuando se encuentra una marca EOM (fin de mensaje) en una operación de reproducción.
- XCLK (26), no se conectó, debido a que se utilizó el clock interno del dispositivo.
- \overline{REC} (27), es una entrada activo baja. El dispositivo graba siempre que esta entrada esta en bajo. Esta señal debe permanecer en bajo durante toda la duración de la grabación. \overline{REC} tiene prioridad frente a las señales de reproducción. Una vez que la grabación concluye una marca EOM se graba internamente, lo que permite que el ciclo de reproducción subsecuente termine apropiadamente.

Primeramente, se configuró el circuito para realizar la grabación de los mensajes necesarios para el equipo. Se muestra el esquemático en la figura 3.5.

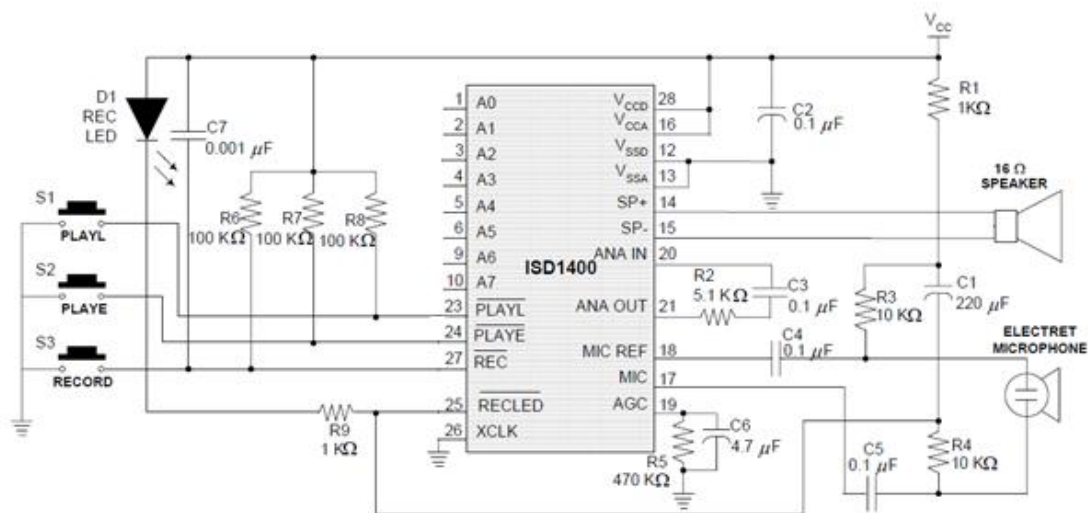


Figura 3.5. Circuito de la plataforma vocal

La plataforma vocal maneja 6 mensajes para dialogar con el usuario durante el uso del equipo, los cuales se detallan a continuación:

- “Ingrese la contraseña”
- “Contraseña incorrecta”
- “Ingrese el código”
- “Código incorrecto”
- “Actuador activado”
- “Actuador desactivado”

Se diagramó el mapa de direcciones de los mensajes. Como A_7 o A_6 debían ser 0 para que las entradas de dirección sean interpretadas como direcciones, se adoptó $A_7 = 0$. Se

necesitaron 6 mensajes, por lo tanto se utilizaron 3 bits para direccionar la memoria ($2^3 = 8 > 6$), los cuales fueron los 3 bits siguientes a A_7 lo que permitió una correcta utilización del espacio de memoria del dispositivo.

A A_3 , A_2 , A_1 y A_0 se las conectó a masa permanentemente, lo que resultó en el mapa de direcciones que se muestra en la tabla 3.2.

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Mensaje
0	0	0	0	0	0	0	0	"Ingrese la contraseña"
0	0	0	1	0	0	0	0	"Contraseña incorrecta"
0	0	1	0	0	0	0	0	"Ingrese el código"
0	0	1	1	0	0	0	0	"Código incorrecto"
0	1	0	0	0	0	0	0	"Actuador Activado"
0	1	0	1	0	0	0	0	"Actuador desactivado"
0	1	1	0	0	0	0	0	
0	1	1	1	0	0	0	0	

Tabla 3.2. Mapa de direcciones de mensajes.

Se calculó la duración de cada mensaje la cual resultó: $D = \frac{20 \text{ seg}}{8 \text{ msj}} = 2,5 \frac{\text{seg}}{\text{msj}}$. Se ensayó empíricamente el tiempo de estas frases mediante un cronómetro, utilizando una cadencia de voz normal y se concluyó que este tiempo era suficiente para grabar correctamente cada mensaje.

3.3. Detector de corriente de llamada

Se incluyó en la interfaz de línea un montaje como el de la figura 3.6. Este circuito se diseñó para estar permanentemente conectado a la línea, por lo tanto no debía consumir ninguna corriente continua sobre la misma, ya que esto equivaldría a ocuparla. Por este motivo se necesitó la inclusión de los condensadores C_1 y C_2 (en serie) que bloquean la componente continua. Así mismo, la red RC serie limita la corriente alterna consumida.

Un puente rectificador asociado a un condensador de filtrado C_3 transforma cada tren de impulsos de llamada en una corriente continua capaz de alimentar el LED del optoacoplador.

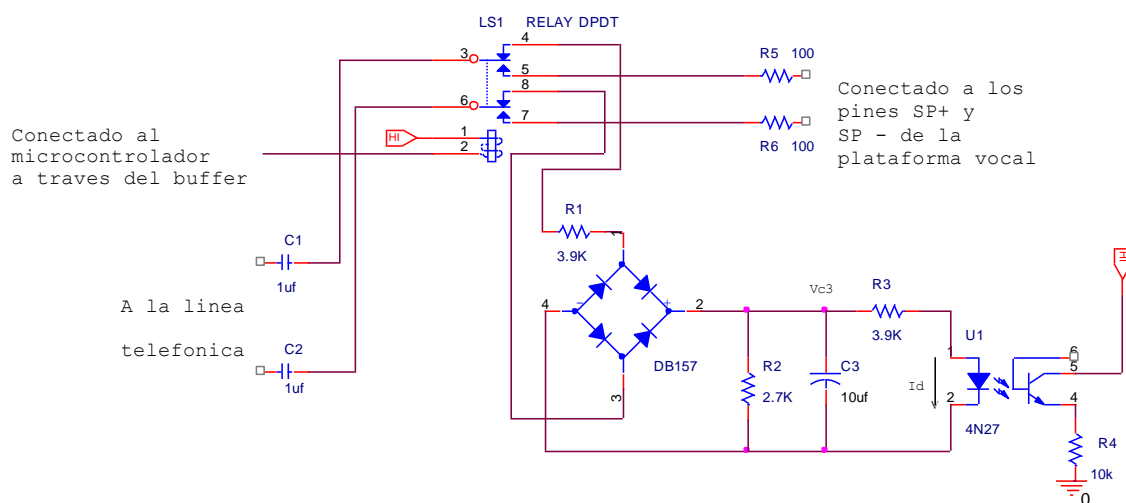


Figura 3.6. Circuito del detector de corriente de llamada.

La salida del montaje se conectó directamente a la entrada del microcontrolador ya que el optoacoplador proporcionó un aislamiento de seguridad suficiente.

Para implementar el puente de diodos se utilizó el integrado DB157 cuya disposición de pines se muestra en la figura 3.7, posee dos pines correspondientes a la corriente alterna y dos a la corriente rectificada de salida [7].

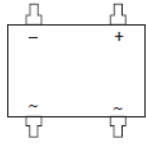


Figura 3.7. Encapsulado del DB157.

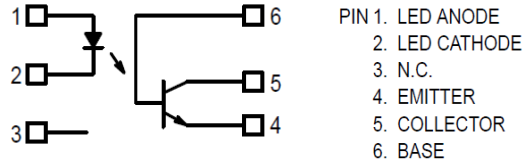


Figura 3.8. Disposición de pines del 4N27.

El optoacoplador se implementó mediante el integrado 4N27, cuya disposición de pines se presenta en la figura 3.8 [8].

El cálculo de los componentes C_1 , C_2 , R_1 , R_2 y R_3 fue realizado basándose en un modelo matemático aproximado, pero utilizando la práctica empírica se afinaron los valores. La corriente de colector para saturar el transistor del optoacoplador 4N27 debe ser aproximadamente $5V/10K\Omega = 500\mu A$. Pudiendo deducirse entonces, de la hoja de datos, que será necesaria una corriente por el LED mayor a 1 mA para mantener saturado el transistor. La tensión en el capacitor C_3 , suponiendo una caída de tensión en el led de 1,15V (tensión típica extraída de la hoja de datos), con una corriente de polarización propuesta de 2mA suficiente para saturar el transistor del optoacoplador, siendo $R_3 = 3,9K\Omega$ (valor adoptado), resulta:

$$V_{c3} = 3,9K\Omega \cdot 2mA + 1,15V = 8,95V$$

Los capacitores C_1 y C_2 y la resistencia R_1 se adoptaron de $1\mu f$ y $3,9K\Omega$ respectivamente. Con estos valores se limita la corriente consumida por el equipo debido a su alta impedancia, la cual resultó:

$$X_c = \frac{1}{2 \cdot \pi \cdot f \cdot C} = \frac{1}{2 \cdot \pi \cdot 16Hz \cdot 0,5\mu f} = 19894 \Omega$$

Suponiendo una corriente por el LED de 2mA, la resistencia reflejada entonces es:

$$R = \frac{8,95V}{2mA} = 4475 \Omega$$

La tensión en el capacitor C_3 , resulta aproximadamente:

$$V_{c3} = \frac{80V \cdot \sqrt{2} \cdot 4475\Omega // 2,7K\Omega}{|4475\Omega // 2,7K\Omega + 3,9K\Omega - j19894 \Omega|} = 9,22V$$

Recalculando la corriente por el LED:

$$9,22V = 3,9K\Omega \cdot I + 1,15V \quad I = 2,06mA$$

La cual es una corriente que asegura la saturación del transistor de salida del optoacoplador.

La resistencia $R_2 = 2,7K$ se dispuso para descargar prácticamente a cero el capacitor C_3 de manera que pueda seguir las pausas entre cada pulso de corriente de llamada. El tiempo mínimo entre pulsos, dependiendo del tipo de central, es de 250 ms, como es el caso de algunas centrales privadas que envían dos ring consecutivos y luego hacen una pausa de 3 segundos. La constante de tiempo inicial involucrada para este caso resulta:

$$\tau_0 = 10 \mu f \cdot (2,7K\Omega // \frac{9,22V}{2,06mA}) = 16,84mseg$$

En el osciloscopio se observaron, a la salida del optoacoplador, los pulsos perfectamente conformados y bien diferenciados unos de otros, con los valores adoptados, como se muestra en la figura 3.9.

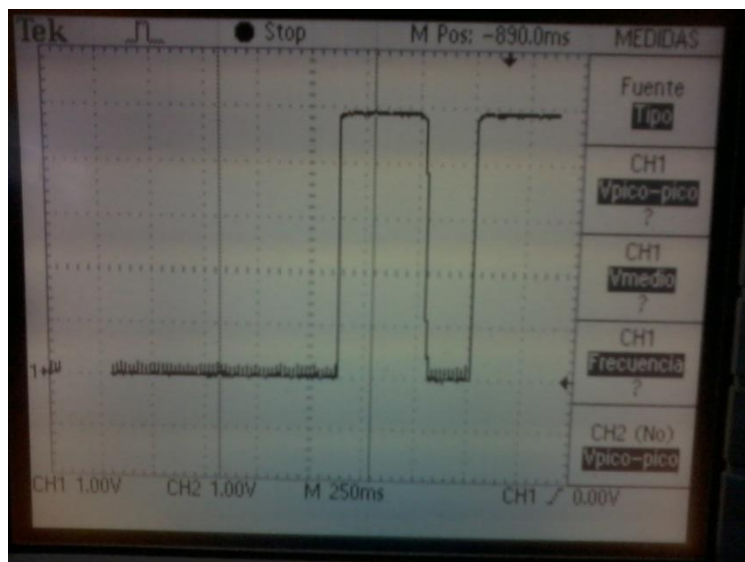


Figura 3.9. Salida del detector de corriente de llamada.

Se configuró una salida por emisor en el transistor de manera de no tener inversión en la polarización de los pulsos de llamada colocándose una resistencia $R_4 = 10K\Omega$.

El relay LS1 se dispuso de manera que en su estado normalmente cerrado la línea telefónica esté conectada al detector de corriente de llamada. Al ser tomada, este relay debe conmutar (a través de una orden del microcontrolador), de manera de enlazar los capacitores C_1 y C_2 con las resistencias R_5 y R_6 y acoplar la salida de la plataforma vocal a la línea telefónica. De esta manera, se evita que la misma (que maneja tensiones altas) esté conectada todo el tiempo a la plataforma vocal.

El acople del audio a la línea telefónica realizado mediante esta configuración balanceada resultó exitoso, el audio es percibido sin ruidos y a un volumen óptimo por el emisor de la llamada.

Cuando el sistema libera la línea telefónica el relay conmuta nuevamente de manera de dejar conectada la misma al detector de corriente de llamada para esperar una nueva llamada entrante.

3.4. Toma de la línea telefónica

Para tomar la línea telefónica se la debe cargar con aproximadamente 600Ω . Se utilizó una resistencia de $560\Omega/3W$ en paralelo con la línea, de esta manera la central interpreta que el teléfono ha sido descolgado. Esto se realizó mediante el circuito de la figura 3.10. Cuando el relay conmuta, por una orden del microcontrolador, la resistencia se conecta y se toma la línea. Para manejar el devanado del relay se utilizó un buffer que se detallará en la siguiente sección.

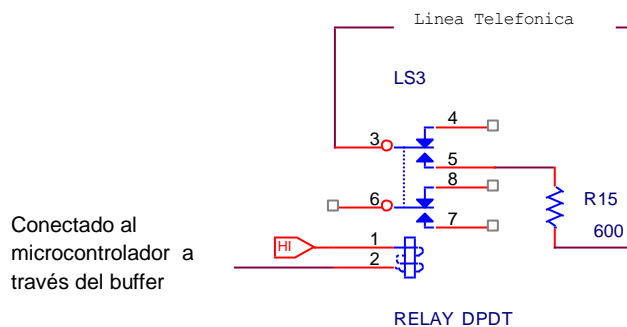


Figura 3.10. Circuito para toma de línea.

3.5. Buffers

Se adoptó el integrado ULN 2803 cuya disposición de pines y circuito esquemático se muestra en la figura 3.11 [9].

Se conectó el devanado de los relays entre los pines correspondientes al colector de cada transistor y la fuente de 5 volt. Los diodos se dispusieron en paralelo a los relays, por lo tanto el terminal 10 se conectó directamente a la fuente.

Los pines 1 a 8 se ligaron directamente a cada una de las salidas del microcontrolador que manejan los relays.

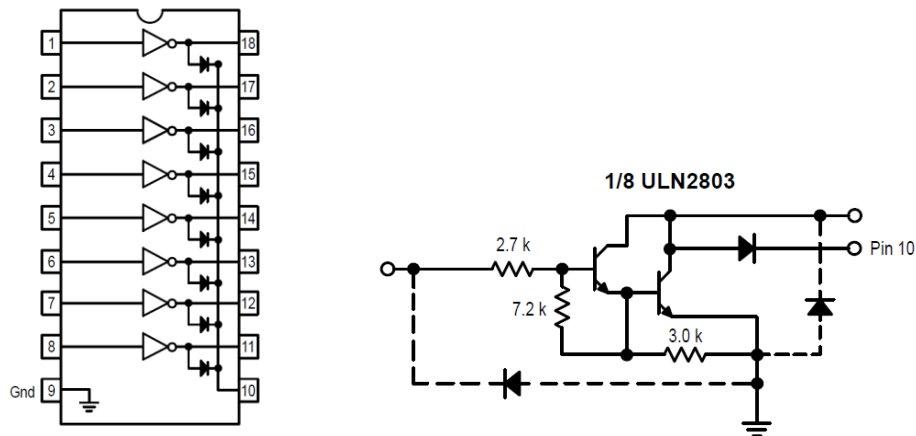


Figura 3.11. Esquemático y disposición de pines del ULN2803.

3.6. Display LCD

Se utilizó un display LCD Winstar modelo WH1602B [6]. Este display posee dos registros de 8 bits, uno de instrucciones (IR) y otro de datos (DR). El registro IR almacena códigos de instrucción, tales como el borrado de la pantalla o el desplazamiento del cursor. El registro DR almacena temporalmente los datos a ser escritos o leídos de la DDRAM. La selección de los mismos se realiza a través del registro de selección RS.

Upper 4 bit Lower 4 bit	LLLL	LLLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH
LLLL	CG RAM (1)				0	1	2	3			4	5	6	7	8	9
LLLH	CG RAM (2)				:	;	<	>			[]	{	}	~	.
LLHL	CG RAM (3)				"	#	\$	%			^	_	~	+	=	-
LLHH	CG RAM (4)				*	+	=	-			~	+	=	-	~	+
LHLL	CG RAM (5)				@	A	B	C			E	F	G	H	I	J
LHLH	CG RAM (6)				K	L	M	N			O	P	Q	R	S	T
LHHL	CG RAM (7)				U	V	W	X			Z	[]	{	}	~
LHHH	CG RAM (8)				7	8	9				4	5	6	7	8	9
HLLL	CG RAM (1)				C	D	E	F			H	I	J	K	L	M
HLLH	CG RAM (2)				N	O	P	Q			S	T	U	V	W	X
HLHL	CG RAM (3)				Y	Z	[]			{	}	~	+	=	-
HLHH	CG RAM (4)				+	=	-	~			~	+	=	-	~	+
HHLL	CG RAM (5)				.	~	+	=			~	+	=	-	~	+
HHLH	CG RAM (6)				-	~	+	=			~	+	=	-	~	+
HHHL	CG RAM (7)				~	+	=	-			~	+	=	-	~	+
HHHH	CG RAM (8)				~	+	=	-			~	+	=	-	~	+

Tabla 3.3. Patrones de caracteres.

La memoria DDRAM es usada para almacenar los datos del display representados en códigos de caracteres de 8 bits. Su capacidad extendida es de 80 x 8 bits o sea 80 caracteres.

La memoria CGROM genera patrones de caracteres de 5 x 8 o 5 x 10 puntos a través de códigos de 8 bits que se muestran en la tabla 3.3.

Cuando el busy flag es 1, quiere decir que el controlador se encuentra en medio de una operación, y la próxima instrucción no será aceptada. Cuando RS = 0 y R/W = 1, el busy flag se muestra en DB7 (Tabla 3.4). Se debe escribir la próxima instrucción luego de asegurarse que este flag se encuentre en 0. En este proyecto la entrada R/W se conectó a masa, lo que equivale a no leer el busy flag, ya que se utilizaron retardos suficientes entre las instrucciones para evitar escribir mientras el controlador está ocupado.

RS	R/W	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB7)
1	0	Write data to DDRAM or CGRAM (DR to DDRAM or CGRAM)
1	1	Read data from DDRAM or CGRAM (DDRAM or CGRAM to DR)

Tabla 3.4. Registro de selección.

Pin No.	Symbol	Level	Description
1	V _{SS}	0V	Ground
2	V _{DD}	5.0V	Supply Voltage for logic
3	VO	(Variable)	Operating voltage for LCD
4	RS	H/L	H: DATA, L: Instruction code
5	R/W	H/L	H: Read(MPU→ Module) L: Write(MPU→ Module)
6	E	H,H→ L	Chip enable signal
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB7	H/L	Data bit 7
15	A	-	LED +
16	K	-	LED -

Tabla 3.5. Disposición de pines.

Los pines correspondientes a este modelo de display se muestran en la tabla 3.5. Funciona con una alimentación de 5V, lo que lo hace directamente compatible con el microcontrolador que trabaja con la misma tensión.

El pin V0 (3) se conectó al punto medio de un potenciómetro, de esta manera se reguló el contraste de la pantalla.

La pata E (6), se utilizó para habilitar el bus de datos de display y la pata RS para seleccionar entre el registro de instrucción o de datos. Estos dos pines se manejaron a través de dos líneas del microcontrolador.

Las instrucciones disponibles se muestran en la tabla 3.6. Se programaron instrucciones a través de funciones en C, como clear display (borrado de display), return home (regresa el cursor a su posición original) y line 2 (línea 2).

Los pines A (15) y K (16), son las terminales correspondientes al ánodo y cátodo de un led para la iluminación del display. El pin 15 se conectó a través de una resistencia de 10Ω a Vcc y el 16 a masa.

Las líneas de datos, DB0-DB7 (7-14) se manejaron directamente mediante un puerto del microcontrolador.

Instruction	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	Description Time (540KHz)	
Read display data	1	1	Read data								Read data into DDRAM/CGRAM/SEGRAM	18.5us	
Write display data	1	0	Write data								Write data into DDRAM/CGRAM/SEGRAM	18.5us	
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM, and set DDRAM address to "00H" from AC	0.76ms	
Return Home	0	0	0	0	0	0	0	0	0	1	X	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	0.76ms
Entry Mode Set	0	0	0	0	0	0	0	0	1	I/D	S	Assign cursor moving direction and specify display shift. These operations are performed during data read and write. I/D="1": increment I/D="0": decrement	18.5us
Display ON/OFF	0	0	0	0	0	0	0	1	D	C	B	Set Display /Cursor/Blink On/OFF D="1": display on D="0": display off C="1": cursor on C="0": cursor off B="1": blink on B="0": blink off	18.5us
Cursor or Display shift	0	0	0	0	0	0	1	S/C	R/L	X	X	Cursor or display shift S/C="1": display shift S/C="0": cursor shift R/L="1": shift to right R/L="0": shift to left	18.5us
Function Set	0	0	0	0	0	1	DL	N	F	X	X	Set Interface Data Length DL= 8-bit interface/ 4-bit interface N= 2-line/1-line display F= 5x8 Font Size / 5x11 Font Size	18.5us
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0		Set CGRAM address in address counter	18.5us
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0		Set DDRAM address in address counter	18.5us
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0		Can know internal operation is ready or not by reading BF. The contents of address counter can also be read. BF="1": busy state BF="0": ready state	0us

Tabla 3.6. Instrucciones.

El display se inicializó en el modo de dos líneas y debido a la disponibilidad de pines de entrada salida del microcontrolador, se decidió utilizar la interfaz de 8 bits en lugar de la de 4.

Como se muestra en la figura 3.12, antes de poder utilizar el módulo LCD, este debe inicializarse en el modo de 8 bits ingresando la siguiente secuencia de comandos de la tabla 3.6:

00110000 }
00110000 } Se ingresa tres veces este comando debido a que no se utilizó la línea
00110000 } R/W, de esta manera se asegura que el dispositivo entre en el modo de
operación de 8 bits

00111100 : Los bits N y F, corresponden a la elección de 2 líneas y a un tamaño de fuente de 5x8 (función "Function Set").

00001100 : Corresponde a la instrucción "Display ON/OFF", el bit D se seteo en 1 (display prendido), el bit C y B en 0 (sin cursor y sin parpadeo).

00000001 : Corresponde a la instrucción "Display clear" (borrado de display).

00000110: Corresponde a la instrucción "Entry mode set". El bit I/D en 1 corresponde a un incremento del cursor cada vez que se escribe un caracter. El bit S en 0 hace que los caracteres en el display no se muevan.

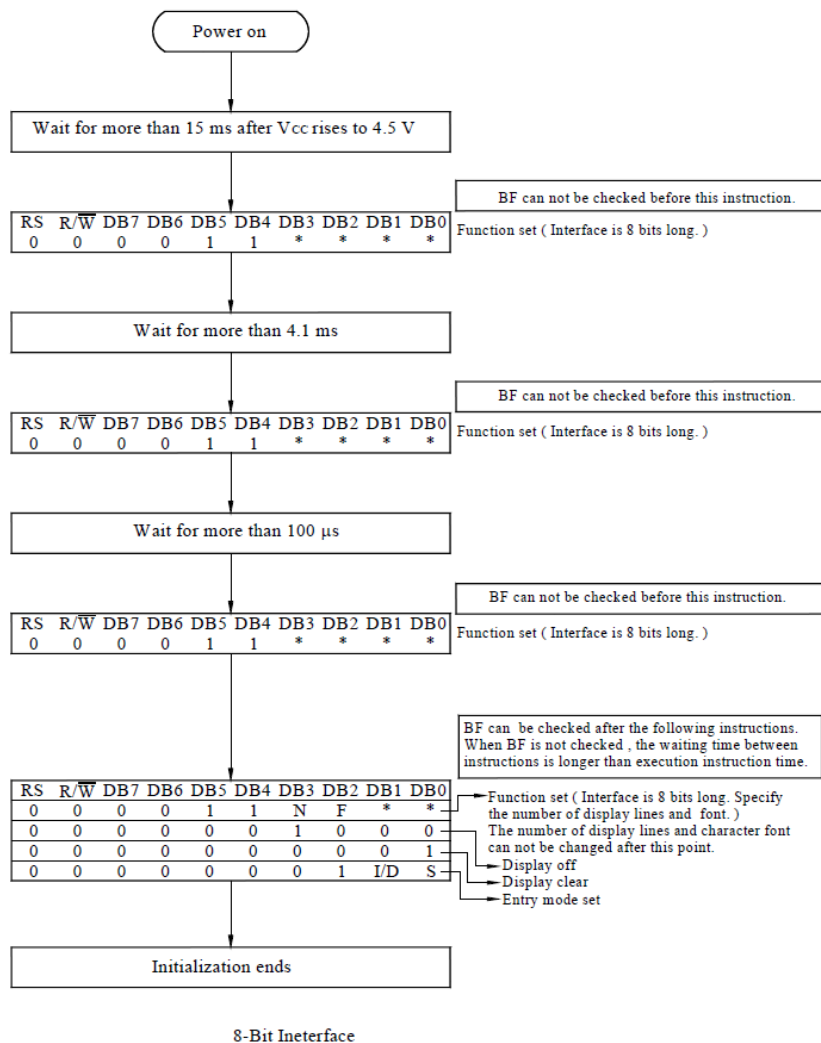


Figura 3.12. Inicialización.

3.7. Microcontrolador

Se utilizó el microcontrolador 18F4550 de la marca Microchip [10].

Se configuró un oscilador externo, para ello se dispuso de un cristal de 8 MHz conectado entre las patas OSC1 y OSC2 (13 y 14).

La disposición de pines de entrada y salida del microcontrolador se muestran en la tabla 3.7. Se utilizaron 27 pines de entradas/salidas.

Puerto	Pin	Entrada/Salida	Descripción
Puerto D Completo	Bus de datos del display	Salida	Manejo de la entrada de los datos hacia el display.
Puerto E0	RS	Salida	Selector de registro del display.
Puerto E1	E	Salida	Habilitación de bus de datos del display.
Puerto E2	Sensado de línea	Entrada	Sensa la línea en búsqueda de pulsos de corriente de llamada.
Puerto B5	LINEA	Salida	Toma la línea a través de la conexión de una resistencia de 560Ω al par telefónico.
Puerto B0	D0	Entrada	Entrada de Datos del DTMF
Puerto B1	D1	Entrada	
Puerto B2	D2	Entrada	
Puerto B3	D3	Entrada	
Puerto B4	DV	Entrada	Línea de dato valido del DTMF
Puerto A0	PV0	Salida	Líneas de direcciones de la plataforma vocal
Puerto A1	PV1	Salida	
Puerto A2	PV2	Salida	
Puerto A3	\overline{PLAYE}	Salida	Línea de manejo del ciclo de reproducción de la plataforma vocal
Puerto A4	Relay 1	Salida	Líneas para el manejo del Estado de los actuadores
Puerto A5	Relay 2	Salida	
Puerto C0	Relay 3	Salida	
Puerto C1	Relay 4	Salida	
Puerto C2	Habilita el audio	Salida	Conmuta el relay que conecta el par telefónico a la plataforma vocal/Detector de corriente de llamada
Puerto C7	BOTÓN	Entrada	Botón que permite la ejecución del módulo de programación del display

Tabla 3.7. Disposición de pines de entrada/salida.

3.7.1. Esquema general del programa.

En la figura 3.13 se muestra un esquema general del funcionamiento del programa a través de un diagrama de flujo.

Considerando el teléfono “colgado”, el sistema debe permanecer en espera de un llamado. Como existe la posibilidad que el teléfono sea atendido, el sistema debe contar cierta cantidad de tonos de llamada antes de tomar la línea, valor programable por el usuario. Una vez hecho esto, el sistema emite el mensaje “Ingrese la contraseña”. No se utilizó ninguna frase de reconocimiento debido a que no quiere darse información sobre la existencia del sistema a cualquier persona que pueda llamar al domicilio.

Luego, el usuario debe ingresar la contraseña para acceder. En el caso de ser incorrecta, se emite el mensaje “contraseña incorrecta” y se corta la comunicación, en el caso de ser correcta se reproduce el mensaje “ingrese el código”.

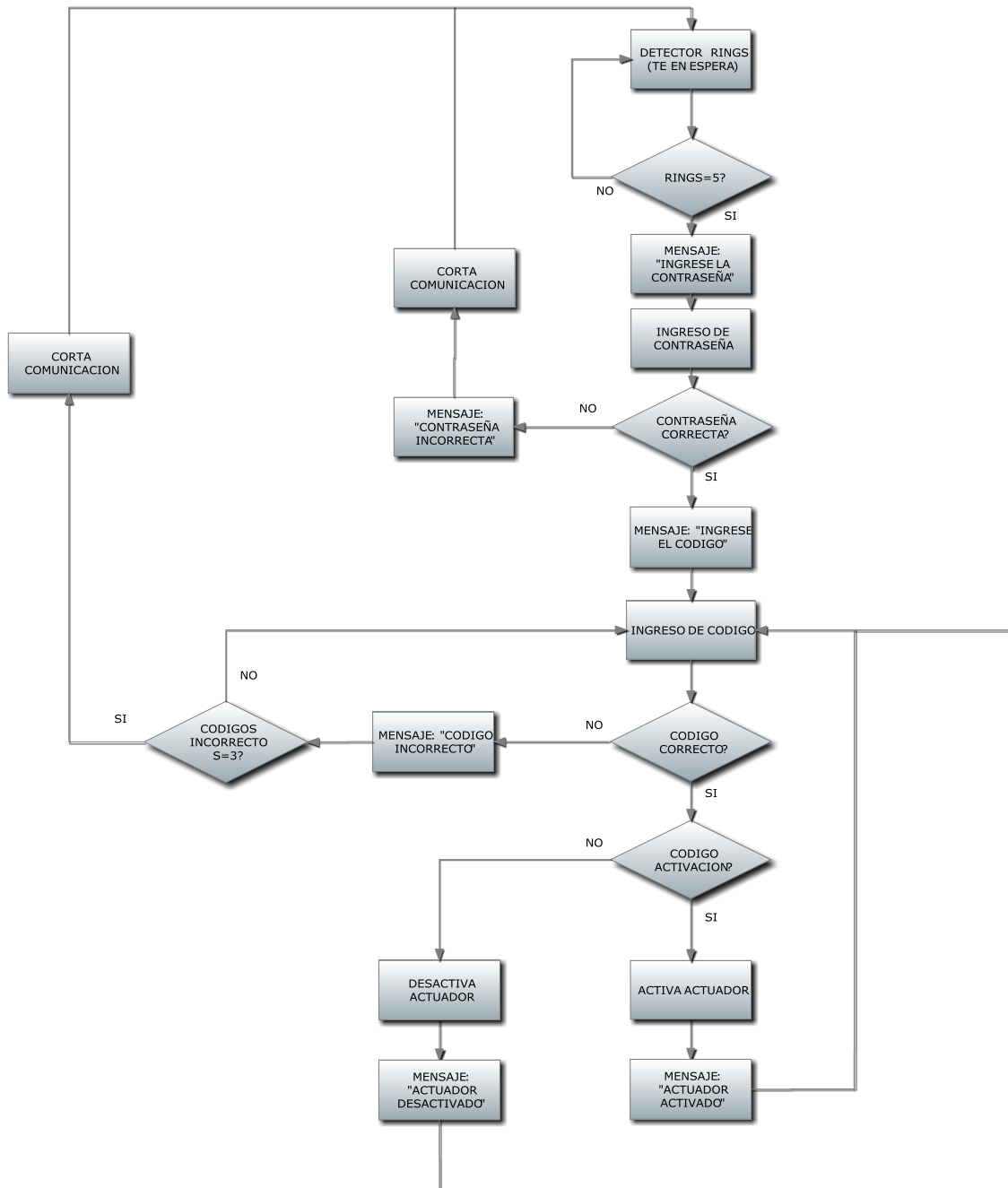


Figura 3.13. Esquema general del programa.

El usuario deberá ingresar un código ya sea de activación o desactivación de uno de los cuatro actuadores. Los códigos son fijos debido a que la seguridad del sistema reside en la contraseña.

Se ideó una codificación sencilla de recordar por parte del usuario, mostrada en la tabla 3.8. Corresponde al ingreso de dos dígitos consecutivos, el primer dígito concuerda con el número de actuador que se quiere activar o desactivar, el segundo corresponde a la acción, definiéndose el dígito 1 (uno) para activar o 0 (cero) para desactivar.

	Activado	Desactivado
Actuador 1	1 - 1	1 - 0
Actuador 2	2 - 1	2 - 0
Actuador 3	3 - 1	3 - 0
Actuador 4	4 - 1	4 - 0

Tabla 3.8. Codificación.

En el caso que el código ingresado por el usuario sea incorrecto, se reproducirá el mensaje “código incorrecto”, habiendo tres oportunidades de falla, después de éstas el sistema corta la comunicación.

Si el código ingresado es correcto se reproducirán los mensajes “actuador activado” o “actuador desactivado”, según corresponda, y nuevamente se dará la oportunidad de ingresar un nuevo código.

3.7.2. Detector de rings

El diagrama de flujo del detector de rings se muestra en la figura 3.14. Se utilizaron tres variables en este bloque de programa:

- C0: cuenta muestras tomadas del detector de corriente de llamada en cero.
- C1: cuenta muestras tomadas del detector de corriente de llamada en uno.
- RING: cuenta la cantidad de pulsos de llamada.

Sensado de línea corresponde al estado de la salida del detector de corriente de llamada que se conecta directamente a un puerto del microcontrolador, puerto E2 en este caso (ver tabla 3.7).

Línea, es la salida del microcontrolador que ejecuta la toma de línea por parte del circuito, colocando una resistencia de 560Ω sobre el par telefónico.

Se toman muestras cada 90 milisegundos, en el caso que la salida del detector de corriente de llamada sea uno, se incrementa la variable C1 y la variable C0 se resetea. Luego, si se toman tres muestras seguidas en uno, se incrementa la variable RING. En el caso que la cantidad de pulsos de llamada sean iguales a cinco ($RING = 5$, en este caso) entonces se toma la línea y se resetean las variables, en caso contrario se sigue sensando la línea en búsqueda de más pulsos de corriente de llamada.

Se tomó la precaución de corroborar la cantidad de muestras en uno tres veces para evitar que el ruido presente en el circuito sea tomado como un pulso falso.

En el caso que el sensado de corriente de llamada en la línea sea cero, se ejecuta el conteo de pulsos en cero tomados sobre la línea, de esta manera si el teléfono es atendido o si el emisor del llamado deja de llamar, luego de seis segundos, las variables se resetean para que el sistema esté listo para un nuevo llamado.

La constante 43 en hexadecimal, corresponde al número 67 en decimal, por lo tanto:

$$90mseg \cdot 67 = 6,03 seg$$

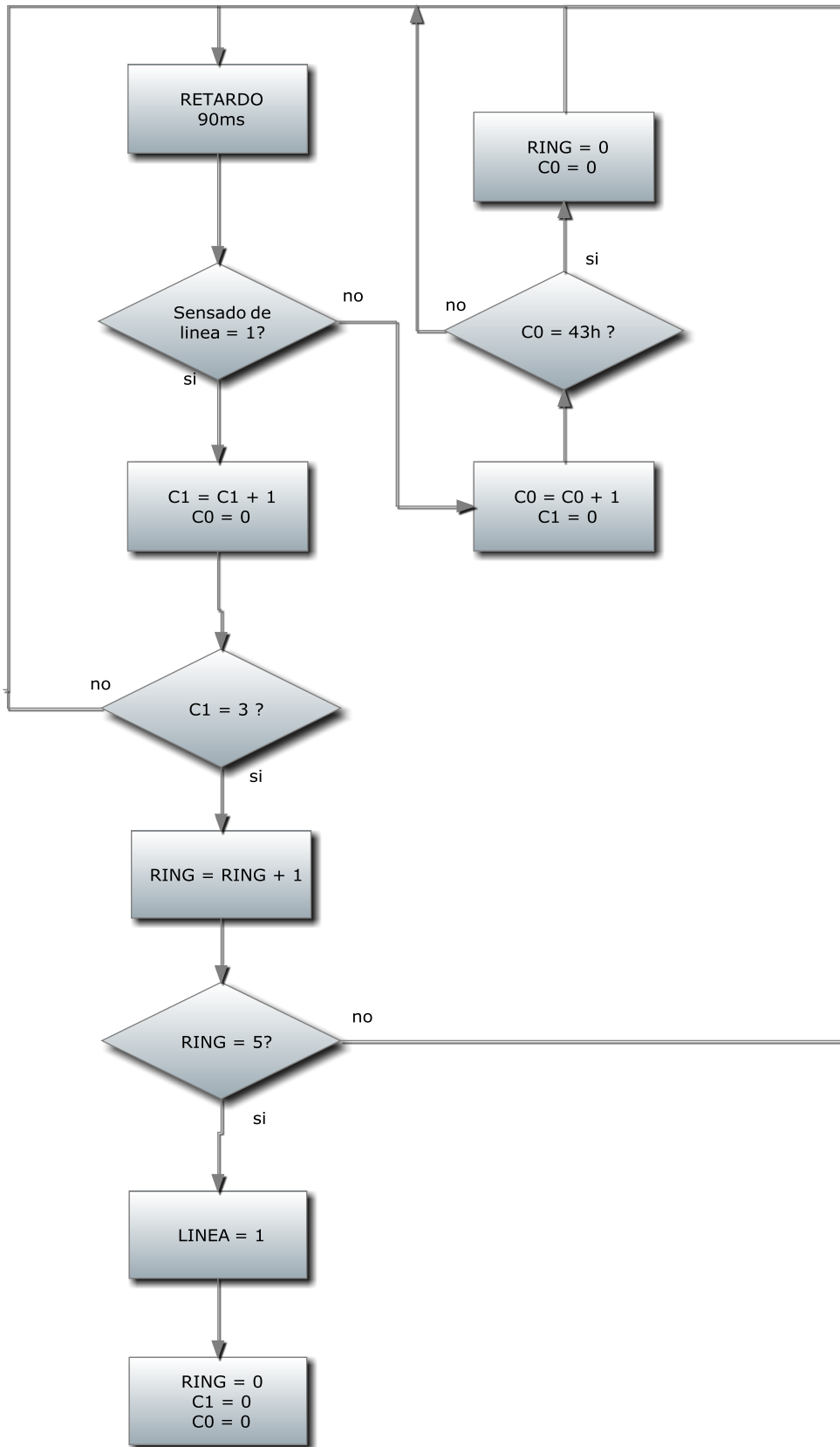


Figura 3.14. Diagrama de flujo del detector de rings.

3.7.3. Ingreso de dígitos

El ingreso de los dígitos tanto para la contraseña como para los códigos se realizó mediante un mismo protocolo ejecutado por una sólo función cuyo diagrama de flujo se muestra en la figura 3.15.

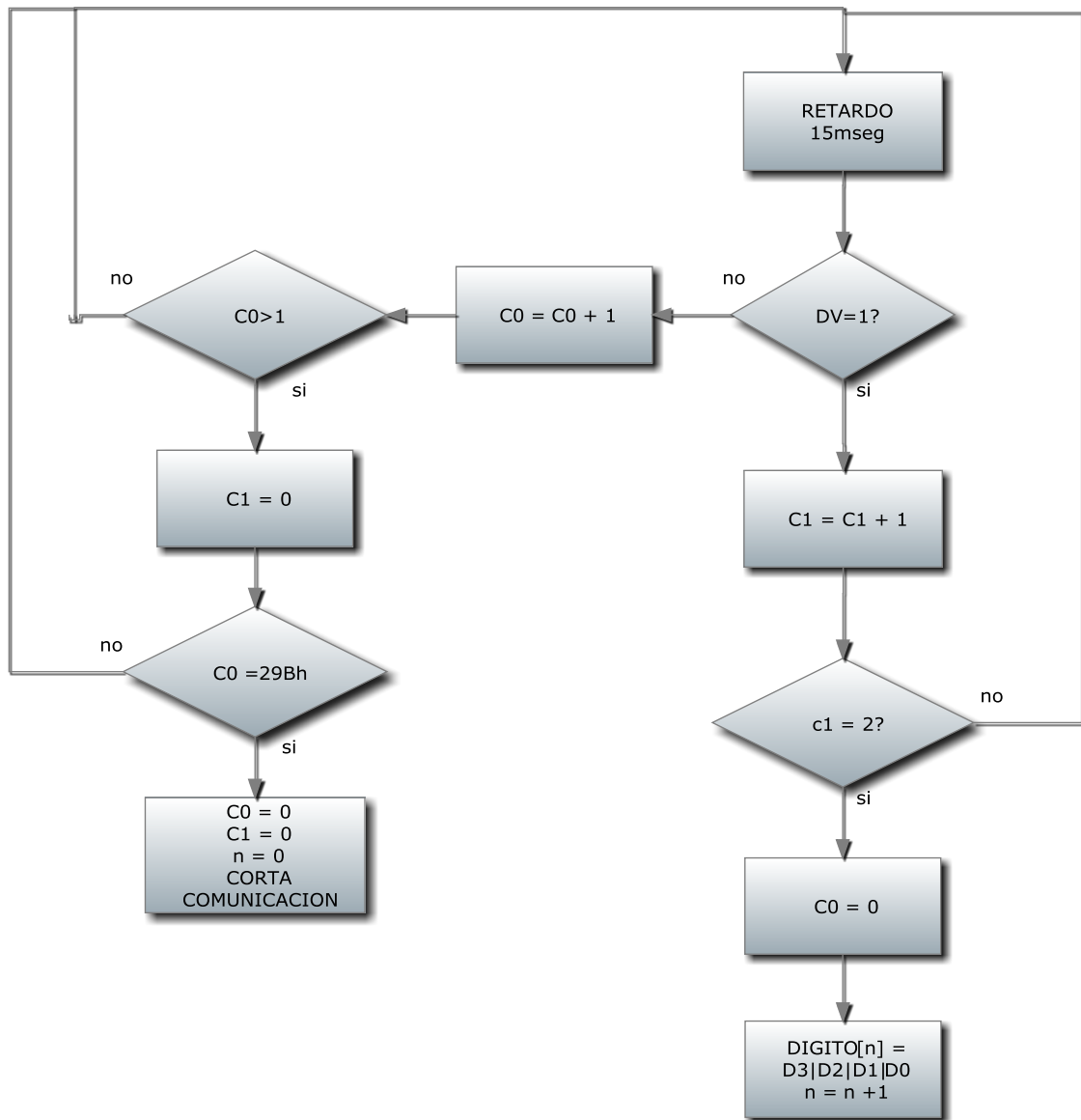


Figura 3.15. Protocolo para el ingreso de los dígitos.

Se realizó un esquema similar al de la sección previa donde se utilizaron las siguientes variables:

- C0: cuenta muestras tomadas de la salida DV (validación de datos) del detector de tonos multifrecuentes en cero.
- C1: cuenta muestras tomadas de la salida DV en uno.
- n: cuenta la cantidad de dígitos tomados.
- Array DIGITO: array de cuatro variables donde se almacenan los dígitos leídos.

DV corresponde al pin DV del detector de tonos multifrecuentes (ver sección: “Detector de tonos multifrecuentes”), el cual se conectó a una entrada del microcontrolador (ver tabla 3.7).

D0, D1, D2, D3 son datos de salida del detector de tonos multifrecuentes, conectados a las correspondientes entradas del microcontrolador (ver tabla 3.7).

Se toman muestras cada 15 milisegundos, en el caso que DV sea uno significa que existe un dato válido en el detector de tonos multifrecuentes, en este caso la variable C1 se incrementa y si se toman dos muestras en uno seguidas, entonces se lee y almacena el dato del dígito ingresado por el usuario en la memoria como DIGITO [n].

En el caso que DV sea cero, significa que no existe ningún dígito nuevo ingresado por el usuario, en este caso la variable C0 se incrementa. Si el usuario no ingresa un dígito durante diez segundos se resetean las variables y se corta la comunicación. Esto se deduce de:

$$C0 = 29B_n = 667_d \quad 667 \times 15\text{mseg} = 10 \text{ segundos.}$$

Para tomar como válido un cambio en DV se cuentan dos ceros seguidos, de esta manera se obtuvo inmunidad frente al ruido y los rebotes de las teclas.

3.7.4. Ingreso de la contraseña

El protocolo completo para el ingreso de la contraseña se muestra en la figura 3.16.

El programa se diseñó para contraseñas de cuatro dígitos únicamente, una vez ingresada, se compara con la contraseña almacenada en memoria. En el caso que la contraseña sea incorrecta se reproduce el mensaje “contraseña incorrecta” y se corta la comunicación, si es correcta se emite el mensaje “ingrese el código” y se continúa con el programa.

3.7.5. Ingreso de los códigos

El protocolo completo para el ingreso de los códigos se muestra en la figura 3.17.

Una vez ingresados los dos dígitos del código se ingresa a una estructura de decisión basada en la tabla 3.17. Si el código ingresado corresponde a un código de activación se reproduce el mensaje “actuador activado”, si corresponde a uno de desactivación se reproduce el mensaje “actuador desactivado”. Si el código es inválido, se reproduce el mensaje “código incorrecto”.

Se programó de manera que al ingresar dos códigos incorrectos se corte la comunicación.

3.7.6. Módulo de cambio de contraseña y cantidad de rings

El prototipo cuenta con un pulsador que permite la programación de la contraseña y la cantidad de rings que debe esperar el sistema para tomar la línea telefónica. Estas dos variables son almacenadas en el módulo de memoria EEPROM del microcontrolador lo que permite que queden registrados aún con cortes de alimentación.

El diagrama de flujo de este módulo se muestra en la figura 3.18.

Para utilizar un solo pulsador y por lo tanto un solo puerto de entrada al microcontrolador, se tuvo en cuenta la siguiente opción: si se presiona el pulsador por más de un segundo, se accede al módulo de cambio de cantidad de rings, y por menos de un segundo al de cambio de contraseña.

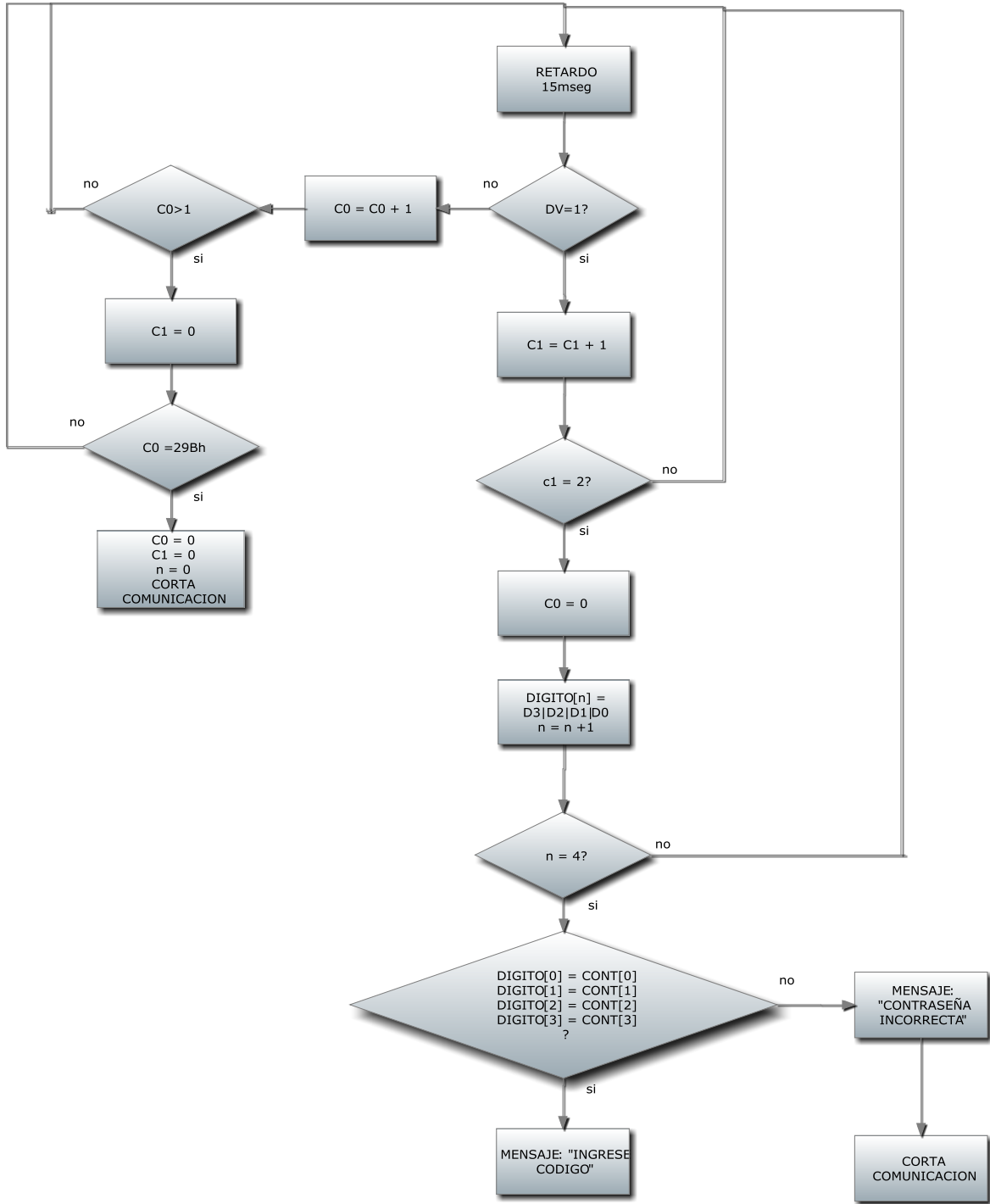


Figura 3.16. Protocolo para el ingreso de la contraseña.

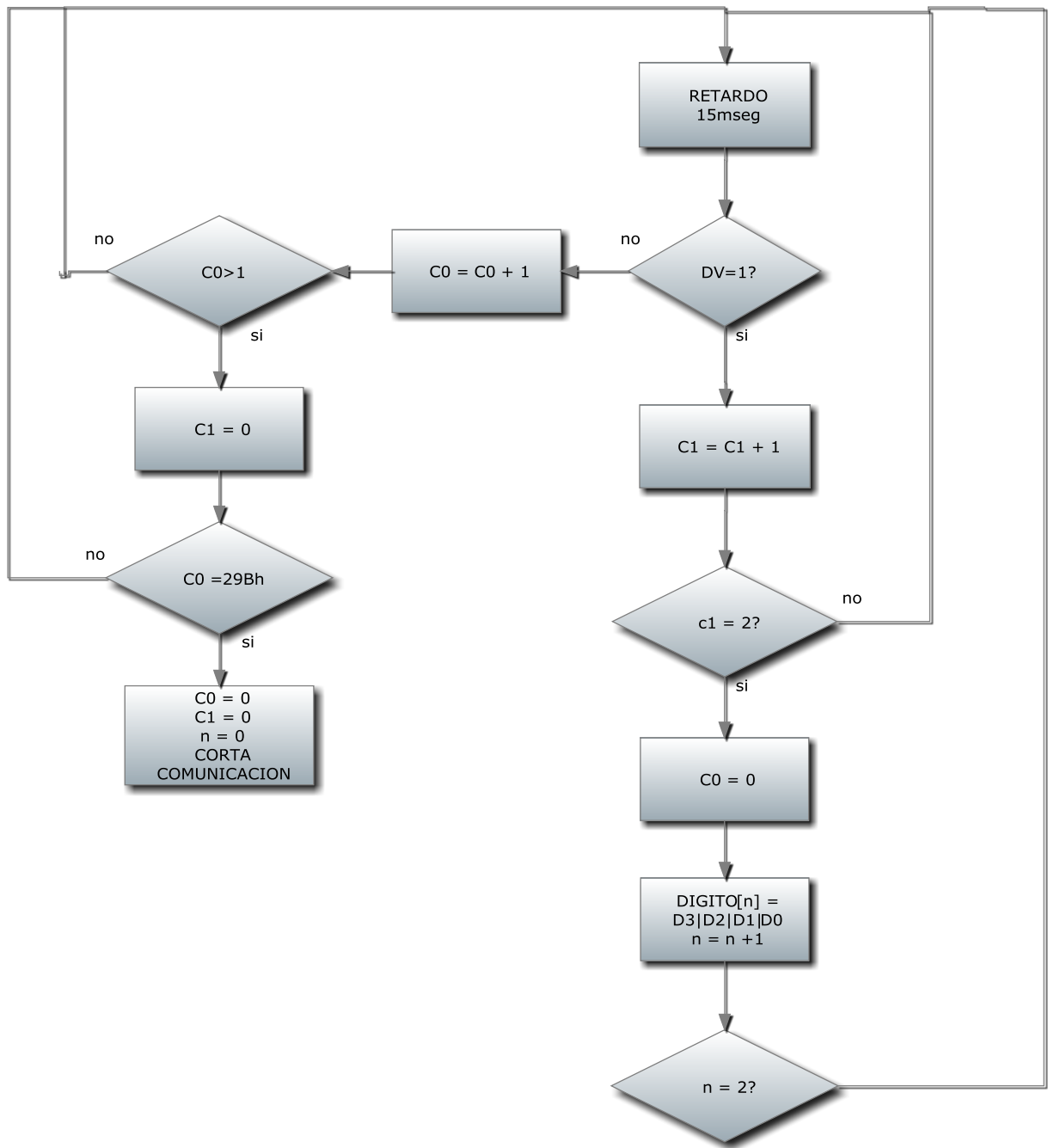


Figura 3.17. Protocolo para el ingreso de los códigos.

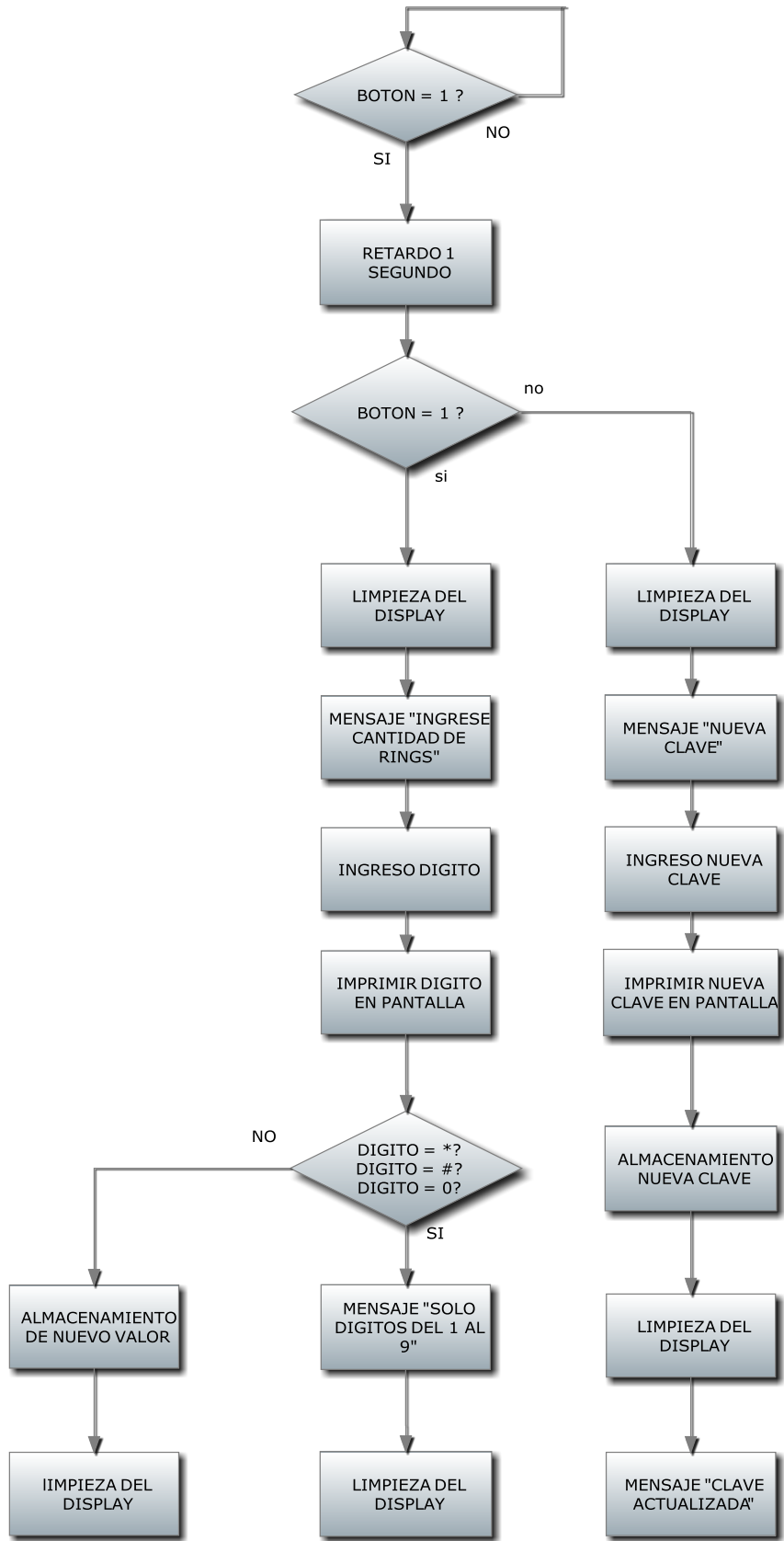


Figura 3.18. Protocolo para el cambio de contraseña y cantidad de rings

Para registrar los dígitos en la EEPROM del sistema se empleó el protocolo mostrado en el diagrama de flujo de la figura 3.19.

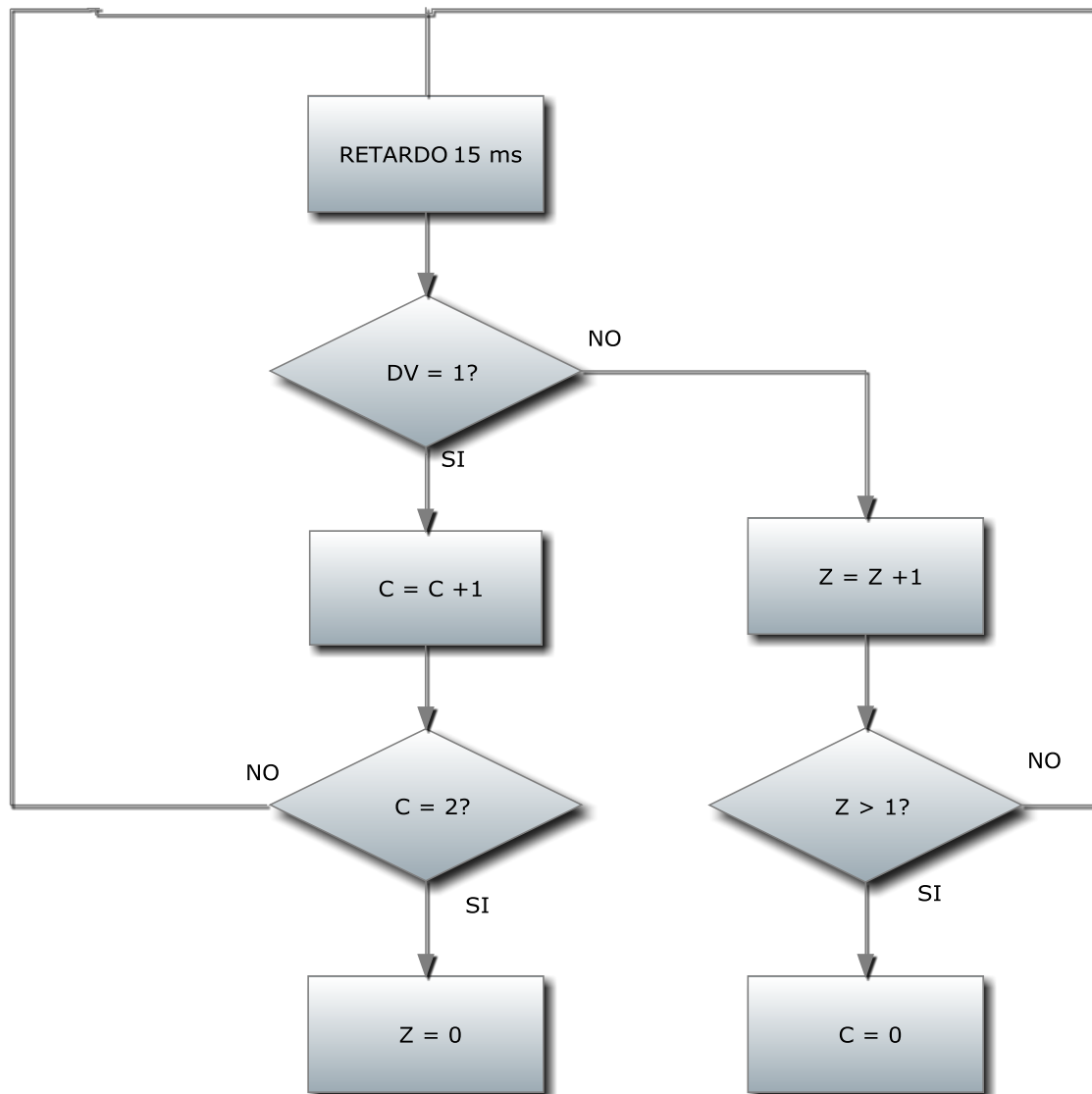


Figura 3.19. Protocolo de programación de dígitos.

Se utilizaron dos variables en este bloque de programa:

- Z: cuenta muestras tomadas de la salida DV (validación de datos) del detector de tonos multifrecuentes en cero.
- C: cuenta muestras tomadas de la salida DV en uno.

Se tomó la precaución de contar dos veces la cantidad de muestras tomadas en uno y en cero para evitar que ruidos presentes en la línea telefónica sean tomados como falsos pulsos.

Cuando se toman dos muestras en uno de la salida DV (C=2) se almacena en memoria el dígito ingresado y se resetea el contador Z.

La contraseña se almacena en la memoria EEPROM del dispositivo y se la traslada a la memoria RAM para su rápido acceso en la operación del sistema.

3.7.7. Módulo de reproducción de audio

Para reproducir el audio almacenado en la plataforma vocal se utilizó la función *reproducir* que posee tres variables de entrada correspondientes a la dirección de memoria del mensaje a reproducir, mostradas en la tabla 3.2.

La función coloca la dirección del audio en las salidas del microcontrolador ligadas a las entradas de dirección de la plataforma vocal (mostradas en la tabla 3.7), luego genera un pulso de 90 mili-segundos en la línea de entrada \overline{PLAYE} de la plataforma vocal, lo que desencadena la reproducción del mensaje.

3.7.8. Módulo para manejo de display

Para el manejo del display se utilizaron siete funciones, *lcdinit*, *line2*, *clear*, *home*, *command*, *display*, *display_string*, las cuales se detallan a continuación.

- *Command*: se utiliza para ingresar comandos en el display. Se comienza seteando la línea RS, luego se escribe el código del comando deseado en el puerto D del microcontrolador y se genera un pulso en la línea E (de habilitación) que permite la lectura por parte del display.
- *Line2*: permite escribir en la línea 2 del display.
- *Clear*: permite borrar el contenido del display y colocar el cursor en el inicio.
- *Home*: Coloca el cursor en el inicio.
- *Lcdinit*: permite ingresar los comandos para la inicialización del display, mostrados en la tabla 3.12. Se ingresan uno a uno los comandos a través de la función *command*.
- *Display*: permite ingresar datos para escribir en el display. Se comienza seteando la línea RS en uno, luego se escribe el código del carácter (ASCII) en el puerto D del microcontrolador y se genera un pulso en la línea E (de habilitación) que permite la lectura por parte del display.
- *Display_string*: se utilizó para escribir los caracteres del mensaje uno a uno en el display, para eso llama a la función *display* por cada carácter a escribir. El ciclo FOR se ejecuta una cantidad de veces igual a la cantidad de caracteres que posee el mensaje valor dado por la función *strlen* de la librería *string.h* del XC8. En el caso que contenga más de dieciséis caracteres se continúa escribiendo en la línea 2.

3.7.9. Módulo de presentación

Se utilizó la función *presentación* para mostrar en el display el estado de los actuadores. La pantalla muestra información de cada actuador (abreviado A, siendo i el número del actuador) seguido por la palabra ON o OFF, si éste se encuentra activado o desactivado respectivamente. En la primera línea se muestra el estado de los dos primeros actuadores y en la segunda el estado de los otros dos. La figura 3.20 es una foto tomada del display funcionando.

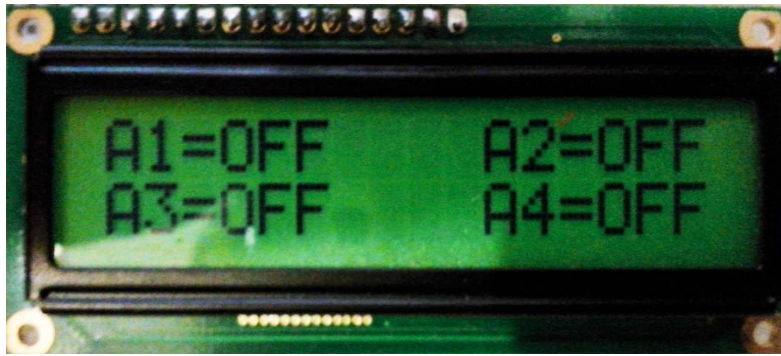


Figura 3.20. Vista del display.

Capítulo 4

Manual de mantenimiento

4.1. Especificaciones

- Tensión de alimentación: 5 V.
- Consumo de energía (funcionamiento al 100% de capacidad): 1,85 W.
- Consumo de energía (equipo en reposo): Menos de 0,05W.
- Lenguaje del software: Español.
- Fuente: 100-240 VAC/50-60 Hz a 5VDC 500mA.

4.2. Diagrama en Bloques

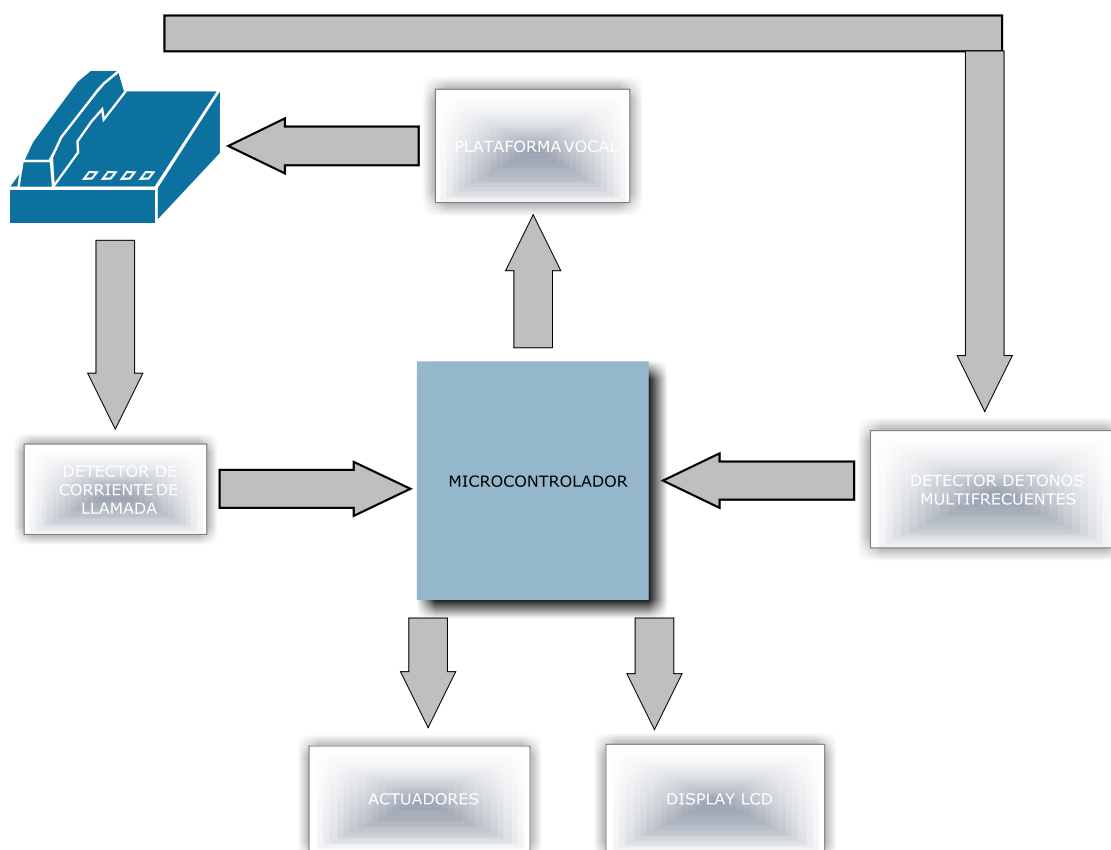


Figura 4.1. Diagrama en bloques

El equipo cuenta con los bloques mostrados en la figura 4.1, los cuales se detallan en el circuito general completo de la figura 4.2.

4.2.1. Detector de tonos multifrecuentes

El integrado HT9170 realiza la detección de los tonos multifrecuentes [4]. Este integrado utiliza técnicas de conteo digitales para detectar y decodificar los 16 pares de tonos multifrecuentes cuyo resultado es una salida de código de 4 bits, que es interpretada por el microcontrolador.

Emplea filtros capacitivos conmutados de alta precisión para separar las señales multifrecuentes en los dos tonos que la conforman. Incluye un rechazo de tono de discar integrado para eliminar la necesidad de un prefiltrado.

4.2.2. Plataforma vocal

La plataforma vocal se implementó mediante el integrado ISD1420 [5], el cual proporciona grabación y reproducción de mensajes de corta duración con alta calidad y en un solo chip. Se trata de un dispositivo CMOS que incluye un oscilador integrado, un preamplificador de micrófono, control automático de ganancia, filtro anti-aliasing y amplificador de altavoz. Las grabaciones son almacenadas en celdas de memoria no volátiles integradas.

4.2.3. Detector de corriente de llamada

El bloque de detección de corriente de llamada se utilizó para dotar al equipo telefónico de la capacidad de contestar automáticamente una llamada, adapta la señal de 80 V eficaces y 16/25Hz (corriente de llamada) presente en la línea a otra que pueda ser leída por el circuito digital (microcontrolador).

Este circuito se diseñó para estar permanentemente conectado a la línea, por lo tanto no debe consumir corriente continua de la misma, ya que esto equivaldría a ocuparla. Por este motivo se necesitó la inclusión de los condensadores C_1 y C_2 (figura 4.2) que bloquean la componente continua. También, la red RC serie limita la corriente alterna consumida.

Un puente rectificador asociado a un condensador de filtrado transforma cada tren de impulsos de llamada en una corriente continua capaz de activar el led de un optoacoplador 4N27 [8].

La salida del montaje se conectó directamente a la entrada del microcontrolador ya que el optoacoplador proporcionó un aislamiento de seguridad suficiente.

El puente de diodos se implementó mediante el integrado DB157 [7] que soporta una tensión RMS máxima de 700 V, la cual es una tensión máxima mucho mayor a la presente en la línea.

Este integrado posee dos pines correspondientes a la corriente alterna y dos pines correspondientes a la corriente rectificada de salida que se esquematizan en el encapsulado mediante los símbolos \sim , + y -.

Para tomar la línea telefónica se colocó una resistencia de $560\Omega/3W$ en paralelo a la línea, de esta manera la central interpreta que el teléfono ha sido descolgado. Esto se realizó mediante el relay LS 3. Cuando el relay conmuta, por orden del microcontrolador, la resistencia es conectada y se toma la línea.

4.2.4. Actuadores

Para manejar los relays se necesitaron buffers de corriente debido a que el devanado de los relay consume una corriente de 60mA (con 5V), la cual el microcontrolador no es capaz de entregar. Se optó por utilizar el integrado UNL2803 que corresponde a un arreglo de ocho transistores Darlington de alta corriente y alta tensión [9].

Los ocho transistores Darlington NPN de este integrado están especialmente diseñados para la interconexión entre circuitos digitales de bajo nivel lógico (como TTL o CMOS) y los requerimientos de tensión/corriente más altos de los relays.

El colector del par Darlington posee un diodo rápido conectado el cual está disponible para el usuario. Este diodo permite que los picos de corriente que se generan en el devanado al conmutar el estado de los relays se extingan a través del diodo y de esta manera se cuida el estado de los transistores que pueden quemarse por sobretensión.

4.2.5. Display

El bloque display LCD se implementó mediante un display Winstar modelo WH1602B, de 2 líneas y 16 caracteres [6]. Esta, es la interface que le permite al usuario programar la contraseña, la cantidad de tonos de llamada para que el sistema atienda la línea y mostrar el estado de los actuadores cuando el sistema se encuentra en reposo.

4.2.6 Microcontrolador

El bloque microcontrolador se implementó mediante un PIC 18F4550 [10] de la línea Microchip de 8 bits cuyas principales características son las siguientes: 32Kbytes de memoria flash, 2048 bytes de memoria RAM, 256 bytes de memoria EEPROM, 35 entradas/salidas, tensión de operación 5 volt, en encapsulado DIP de 40 patas.

4.3. Circuito completo

El circuito completo del equipo se muestra en la figura 4.2.

4.4. Diagrama de flujo

El diagrama de flujo correspondiente a la lógica de funcionamiento del equipo se muestra en la figura 4.3 y 4.4.

La figura 4.3 corresponde al funcionamiento general y la figura 4.4 a la programación de la contraseña y la cantidad de rings.

4.5. Procedimiento de localización de fallas

Siempre debe asegurarse que la tensión de alimentación sea 5 V y que llegue a los pines de alimentación de todos los circuitos integrados.

No puede efectuarse el ingreso de la contraseña, los códigos o la programación del dispositivo: se debe verificar el correcto funcionamiento del integrado HT9170 que detecta los tonos multifrecuentes, en este caso debe asegurarse que los dígitos estén siendo

correctamente transformados en el código binario correspondiente a través de la lectura de los terminales D0, D1, D2, D3 y DV.

El dispositivo toma la línea telefónica sin recibir respuesta de audio: verificar el correcto funcionamiento de la plataforma vocal (integrado ISD1420). Se debe cerciorar que los pines de dirección estén correctamente conectados a los terminales del microcontrolador correspondientes (ver figura 4.2).

Verificar la correcta vinculación de la plataforma vocal con la línea telefónica realizada por los pines 14 y 15 a través de capacitores de 1 μ f y resistencias de 100 Ω . En este caso chequear la conexión y el correcto estado de los elementos.

El dispositivo no responde a los pulsos de llamada: falla en el detector de corriente de llamada o en el relay encargado de tomar la línea.

En el primer caso medir sobre la salida del optoacoplador (pin 4 del mismo). Se deben presentar pulsos de 5V durante el tiempo que el teléfono "llama" y 0 volt en todo otro caso. De no cumplirse esta situación revisar los componentes del detector de corriente de llamada.

En el segundo caso asegurarse que el relay LS2 conmute correctamente, que la resistencia de 560 Ω se encuentre en buen estado y que las conexiones sean correctas.

El display LCD no enciende correctamente: Verificar conexiones de alimentación. Verificar resto de conexiones: pines de datos, habilitación y selección de registro que no permitan la correcta inicialización.

Si la falla se corresponde con el no encendido del backlight: revisar conexiones a la resistencia de 10 Ω desde los pines 15 y 16.

En caso de deficiencia en el contraste del display: revisar el pin 3 que se alimenta a través de un preset. Esta falla provoca que los dígitos no se muestren o se muestren completamente pintados. En este caso el ajuste del preset o su reemplazo puede solucionar el problema.

No operación de los actuadores (relays): Verificar el correcto funcionamiento del integrado ULN2803 que los alimenta.

No corre el programa: Verificar que el microcontrolador posea alimentación de +5V en los pines 11 y 32, y a masa en los pines 12 y 31.

Verificar que cristal de 8 MHz conectado entre los pines 13 y 14 este entregando la señal correcta.

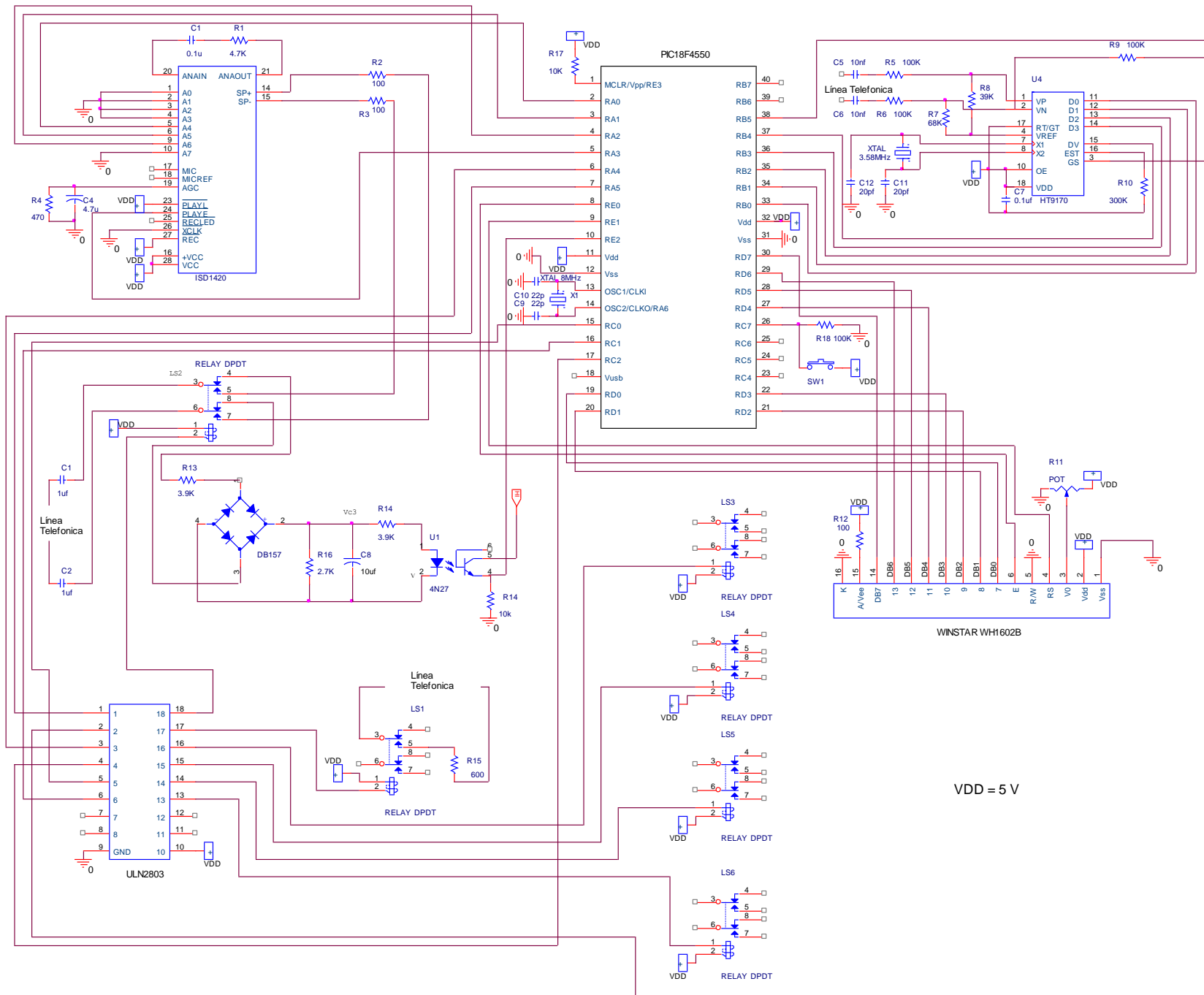


Figura 4.2. Circuito completo.

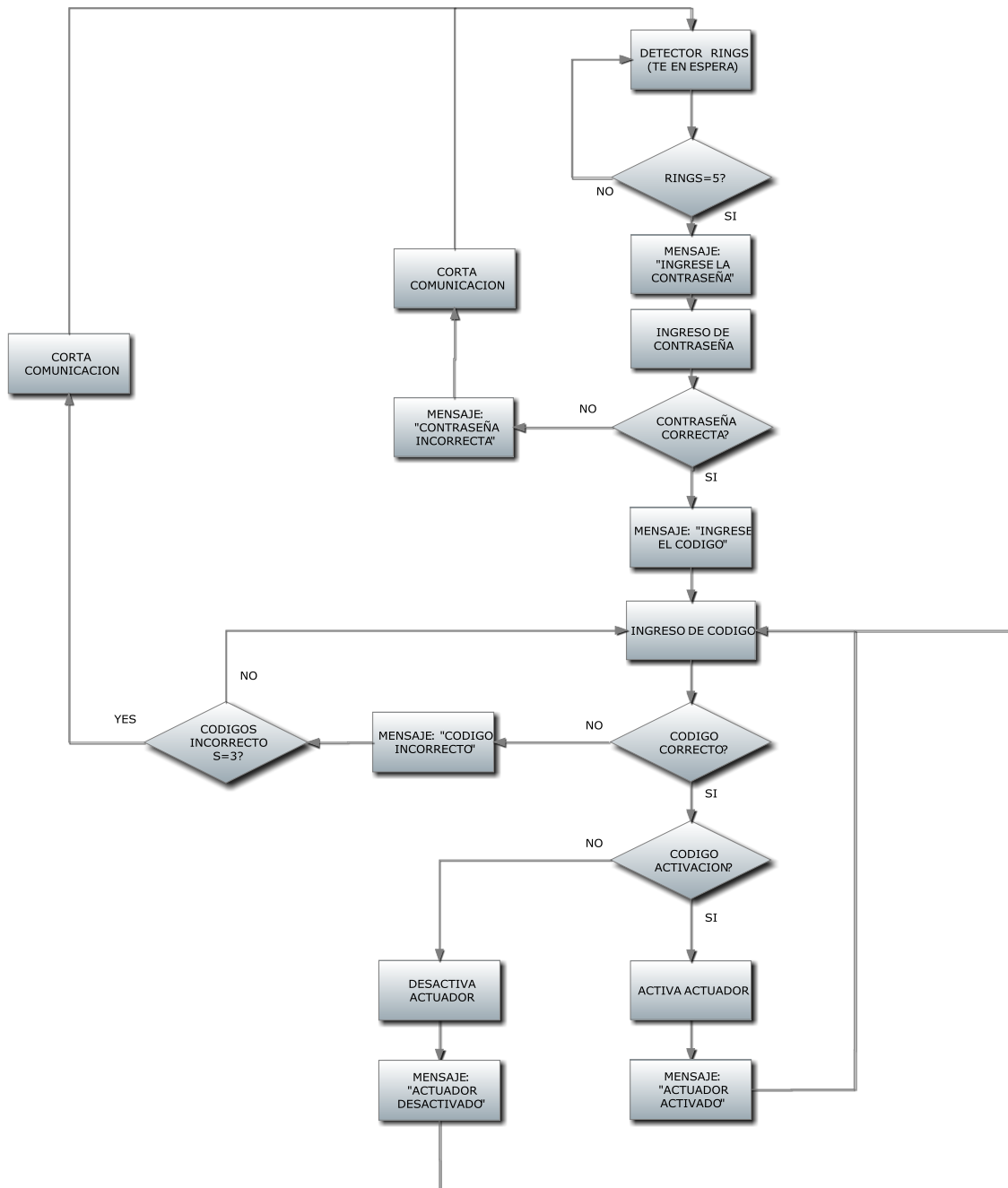


Figura 4.3. Esquema general del programa.

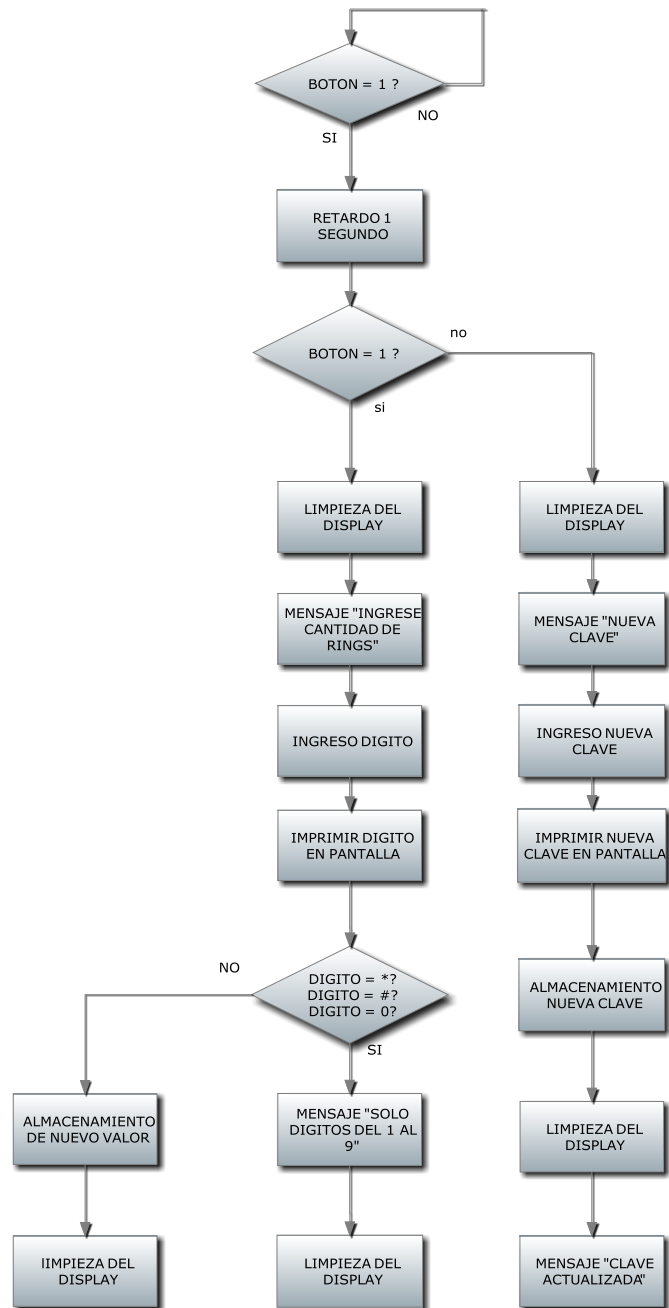


Figura 4.4. Protocolo para el cambio de contraseña y cantidad de rings.

Capítulo 5

Manual de operación

USO DEL MANUAL

Gracias por elegir este equipo. Este equipo le proporcionará manejo de actuadores por línea telefónica. Este manual ha sido diseñado específicamente para guiarlo a través de las características y funciones del dispositivo.

Leer primero

Antes de utilizar el sistema, lea el manual completo y todas las instrucciones para garantizar el uso correcto y seguro.

Las imágenes y capturas de pantalla que figuran en este manual podrían ser diferentes al producto real.

INSTALACIÓN

Desembalaje

Busque los siguientes elementos en la caja del producto:

- Dispositivo telefónico
- Fuente de alimentación
- Cable de telefónico
- Manual de operación
- Adaptador

Los elementos suministrados están diseñados exclusivamente para este dispositivo y podrían no ser compatibles con otros dispositivos.

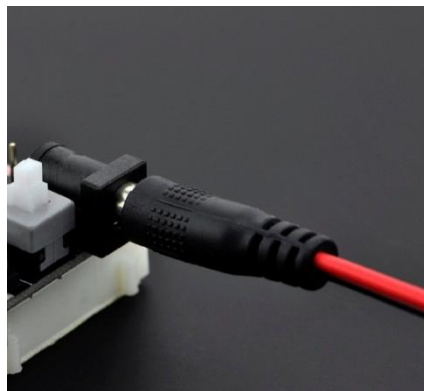
Otros accesorios podrían no ser compatibles con este dispositivo.

Inserción la fuente de alimentación y el cable telefónico

- 1** Conecte la ficha del cargador en una toma eléctrica.



- 2** Conecte la fuente de alimentación al dispositivo.



- 3** Conecte el adaptador provisto en la línea telefónica.



- 4** Conecte el teléfono en uno de los terminales y el dispositivo, mediante el cable provisto, en el otro.

Solo use fuentes y cables provistos por el fabricante. Los cargadores y cables no autorizados podrían dañar el dispositivo. La garantía no cubre daños causados por el uso inadecuado del producto.

CÓMO COMENZAR

Programar la cantidad de tonos de llamada.

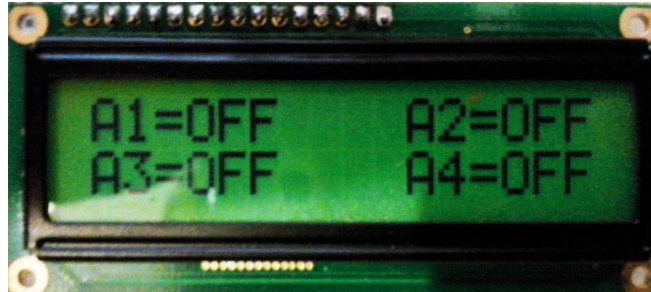
- 1** Mantenga pulsada la tecla de programación por más de un segundo. En la pantalla aparecerá el mensaje "Ingrese cantidad de rings".
- 2** Descuelgue su teléfono.
- 3** Ingrese la cantidad de tonos de llamado para que el sistema atienda la línea. Solo puede ingresar dígitos del 1 al 9.

Programar la clave

- 1** Pulse una vez la tecla de programación. En la pantalla aparecerá el mensaje "Nueva clave"
- 2** Descuelgue su teléfono
- 3** Ingrese la nueva clave de cuatro dígitos. Una vez ingresados, se mostrara el mensaje clave actualizada.

Presentación de la pantalla

Desde la pantalla puede ver el estado del dispositivo. Podrá visualizar el estado de los actuadores, se muestra el actuador (abreviado A) seguido de su numeración (1,2,3,4) y su estado (ON, OFF).



FUNCIONAMIENTO

Manipulación remota del sistema

Aprenda a utilizar las funciones del sistema para activar y desactivar los actuadores.

- 1** Disque el número de teléfono donde está conectado su dispositivo. Luego de cumplirse la cantidad de llamados programados el dispositivo responderá a la llamada.
- 2** El dispositivo cuenta con una plataforma vocal que lo guiará en el manejo del mismo. Primero se le pedirá que ingrese la contraseña de acceso que deberá marcar mediante el teclado telefónico.

Si la contraseña es correcta se le pedirá que ingrese los códigos de dos dígitos correspondientes a la activación o desactivación de los actuadores que se muestran en la tabla. El ingreso de cada dígito esta temporizado, deberá ingresar los dígitos con un retardo menor a seis segundos sino el sistema cortará la comunicación.
- 3**

	Activado	Desactivado
Actuador 1	1 - 1	1 - 0
Actuador 2	2 - 1	2 - 0
Actuador 3	3 - 1	3 - 0
Actuador 4	4 - 1	4 - 0

- 4** Ingresado el código se le confirmará la operación. Luego podrá seguir ingresando códigos. En caso de ingresar un código incorrecto se le informará mediante un mensaje. El sistema admite el ingreso de hasta dos códigos incorrectos, luego de esto se cortará la comunicación.

INFORMACIÓN SOBRE SEGURIDAD

Antes de usar el dispositivo lea la siguiente información a fin de evitar situaciones que pudieran causar lesiones a usted o a otras personas y daños al dispositivo.

- No utilice cables de alimentación o conectores dañados ni enchufes sueltos.
- No toque el cable de alimentación con las manos mojadas ni tire del cable para desconectar la fuente de alimentación.
- No doble ni dañe el cable de alimentación.
- No golpee ni deje caer la fuente de alimentación ni el equipo.
- No alimente el equipo con fuentes que no estén aprobadas por el fabricante.
- Evite dañar el equipo, los cables o la fuente de alimentación.
- Evite exponer el equipo a temperaturas muy altas o muy bajas. Las temperaturas extremas pueden dañar el equipo.
- Mantenga el equipo seco. La humedad y los líquidos pueden dañar las partes o los circuitos electrónicos del equipo. No encienda el equipo si está mojado.
- No utilice ni guarde el equipo en lugares con polvo o sucios. El polvo puede provocar el funcionamiento incorrecto del equipo.
- Guarde el equipo sólo sobre superficies planas. Si se cae, el equipo puede dañarse

Conclusiones y recomendaciones finales

Se diseñó un equipo para el manejo de actuadores por línea telefónica cumpliéndose las especificaciones propuestas al inicio del proyecto. El desarrollo se realizó priorizando permanentemente la facilidad de interacción con el usuario. Durante el transcurso del diseño se fueron optimizando las distintas etapas del sistema, realizando las modificaciones necesarias.

Dada la versatilidad del equipo permitiría agregar a futuro ampliaciones que permitan manejar telefónicamente equipos con mayores requerimientos, por ejemplo programables, haciendo las adecuaciones en las interfaces correspondientes.

Se está contemplando la posibilidad de salida al mercado con la venta directa de la placa a usuarios especializados (instaladores de alarmas, electricistas y otros), o como equipo en su correspondiente gabinete para usuarios en general. Comercialización que se llevaría a cabo a través de la web.

En el aspecto personal se considera haber logrado una buena experiencia en el manejo práctico de microcontroladores y circuitos en general. Ganando confianza en cuanto a que los conocimientos adquiridos durante la carrera permiten una rápida interpretación de temas afines no desarrollados en la misma.

Apéndice A

Se presenta a continuación el programa ejecutado por el microcontrolador escrito en ANSI C.

```
#define PIC18F4550
#include <p18f4550.h>
#include <xc.h>
#include <string.h>

// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

// CONFIG1H
#pragma config FOSC = HS      // Oscillator Selection bits (HS oscillator (HS))
#pragma config FCMEN = OFF    // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor
disabled)
#pragma config IESO = OFF     // Internal/External Oscillator Switchover bit (Oscillator
Switchover mode disabled)

// CONFIG2L
#pragma config PWRT = ON      // Power-up Timer Enable bit (PWRT enabled)
#pragma config BOR = OFF     // Brown-out Reset Enable bits (Brown-out Reset disabled in
hardware and software)
#pragma config BORV = 3      // Brown-out Reset Voltage bits (Minimum setting)
#pragma config VREGEN = OFF   // USB Voltage Regulator Enable bit (USB voltage regulator
disabled)

// CONFIG2H
#pragma config WDT = OFF     // Watchdog Timer Enable bit (WDT disabled (control is placed
on the SWDTEN bit))
#pragma config WDTPS = 32768 // Watchdog Timer Postscale Select bits (1:32768)

// CONFIG3H
#pragma config CCP2MX = ON   // CCP2 MUX bit (CCP2 input/output is multiplexed with
RC1)
#pragma config PBADEN = ON   // PORTB A/D Enable bit (PORTB<4:0> pins are configured
as analog input channels on Reset)
#pragma config LPT1OSC = OFF // Low-Power Timer 1 Oscillator Enable bit (Timer1
configured for higher power operation)
#pragma config MCLRE = ON    // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin
disabled)

// CONFIG4L
#pragma config STVREN = ON   // Stack Full/Underflow Reset Enable bit (Stack full/underflow
will cause Reset)
#pragma config LVP = OFF     // Single-Supply ICSP Enable bit (Single-Supply ICSP disabled)
#pragma config ICPRT = OFF   // Dedicated In-Circuit Debug/Programming Port (ICPORT)
Enable bit (ICPORT disabled)
```

```

#pragma config XINST = OFF    // Extended Instruction Set Enable bit (Instruction set
                             extension and Indexed Addressing mode disabled (Legacy mode))

// CONFIG5L
#pragma config CP0 = OFF    // Code Protection bit (Block 0 (000800-001FFFh) is not code-
                             protected)
#pragma config CP1 = OFF    // Code Protection bit (Block 1 (002000-003FFFh) is not code-
                             protected)
#pragma config CP2 = OFF    // Code Protection bit (Block 2 (004000-005FFFh) is not code-
                             protected)
#pragma config CP3 = OFF    // Code Protection bit (Block 3 (006000-007FFFh) is not code-
                             protected)

// CONFIG5H
#pragma config CPB = OFF    // Boot Block Code Protection bit (Boot block (000000-
                             0007FFFh) is not code-protected)
#pragma config CPD = OFF    // Data EEPROM Code Protection bit (Data EEPROM is not
                             code-protected)

// CONFIG6L
#pragma config WRT0 = OFF    // Write Protection bit (Block 0 (000800-001FFFh) is not write-
                             protected)
#pragma config WRT1 = OFF    // Write Protection bit (Block 1 (002000-003FFFh) is not write-
                             protected)
#pragma config WRT2 = OFF    // Write Protection bit (Block 2 (004000-005FFFh) is not write-
                             protected)
#pragma config WRT3 = OFF    // Write Protection bit (Block 3 (006000-007FFFh) is not write-
                             protected)

// CONFIG6H
#pragma config WRTC = OFF    // Configuration Register Write Protection bit (Configuration
                             registers (300000-3000FFFh) are not write-protected)
#pragma config WRTB = OFF    // Boot Block Write Protection bit (Boot block (000000-
                             0007FFFh) is not write-protected)
#pragma config WRTD = OFF    // Data EEPROM Write Protection bit (Data EEPROM is not
                             write-protected)

// CONFIG7L
#pragma config EBTR0 = OFF    // Table Read Protection bit (Block 0 (000800-001FFFh) is
                             not protected from table reads executed in other blocks)
#pragma config EBTR1 = OFF    // Table Read Protection bit (Block 1 (002000-003FFFh) is
                             not protected from table reads executed in other blocks)
#pragma config EBTR2 = OFF    // Table Read Protection bit (Block 2 (004000-005FFFh) is
                             not protected from table reads executed in other blocks)
#pragma config EBTR3 = OFF    // Table Read Protection bit (Block 3 (006000-007FFFh) is
                             not protected from table reads executed in other blocks)

// CONFIG7H

```

```
#pragma config EBTRB = OFF // Boot Block Table Read Protection bit (Boot block (000000-0007FFh) is not protected from table reads executed in other blocks)
```

```
#define _XTAL_FREQ 8000000
```

```
#define LINEA RB5
```

```
#define SensadoLinea RE2
```

```
#define HabilitaAudio RC2
```

```
#define DV RB4
```

```
#define D0 RB0
```

```
#define D1 RB1 //Bits de salida del DTMF
```

```
#define D2 RB2
```

```
#define D3 RB3
```

```
#define PV0 RA0
```

```
#define PV1 RA1 //Bits de la plataforma vocal
```

```
#define PV2 RA2
```

```
#define PLAYE RA3
```

```
#define relay1 RA4
```

```
#define relay2 RA5 //Puertos de salida de los relays
```

```
#define relay3 RC0
```

```
#define relay4 RC1
```

```
#define ENA RE0 //señales de habilitación y selección de registro para el display
```

```
#define RS RE1
```

```
#define BOTON RC6 //BPulsador para la programación de la contraseña y los rings
```

```
// Variables //
```

```
unsigned int C0; //Contador de ceros
```

```
unsigned int C1; //Contador de unos
```

```
unsigned char n; //Contador de dígitos
```

```
unsigned char Ring; //Contador de rings
```

```
unsigned char LineaTomada; //Variable que indica si la linea fue tomada o no
```

```

unsigned char DIGITO[4];           // Cuatro digitos (del 0 al 3)

unsigned char contadorcodigoincorrecto; //Cuenta la cantidad de veces que el
codigo se ingreso mal

unsigned int i;                   //Para el ciclo for del delay de 3 segundos

int x;                            //Para el ciclo for de display_string

unsigned char m;                 //Para el ciclo for del ingreso de los digitos del cambio
de contraseña

char caracter[1];               //caracter que representa el numero ingresado por el
usuario

unsigned char CONT[4];          //registro en RAM donde se almacena la contraseña,
copiada de la EEPROM;

unsigned int C;                 //Variables para contar ceros y unos en el cambio de
contraseña

unsigned int Z;

unsigned char CringsDTMF;       //Variables donde se almacenan la cantidad de
rings programados por el usuario

unsigned char Nrings;

//Funciones

void ingresodigito (unsigned char *punteron, unsigned char *punteroDIGITO);

void esperadigito (unsigned int *punteroC1, unsigned int *punteroC0, unsigned char
*punteron);

void delay3seg (void);

void reproducir (unsigned char vPV0, unsigned char vPV1, unsigned char vPV2);

```

```

void delay1seg (void);

//Funciones para manejo de display

void lcdinit(void);           //Inicialización del display
void line2(void);           //Mover a la linea 2
void home(void);           //Mueve el cursor a la posicion inicial
void clear(void);           //Limpea el Display
void command(unsigned char lcd_comm); //Ingresar comando al display
void display(unsigned char lcd_data); //Ingresar caracter al display
void display_string(char string[32]); // Ingresar cadena de caracteres al
display

//Funciones para programacion de la contraseña y cantidad de rings

void cambioASCII (unsigned char *m, unsigned char *CONT);
void ingresodisplay (unsigned int *C, unsigned int *Z);
void presentacion (void);
void IngresoRings (unsigned char *CringsDTMF, unsigned char *Nrings);

//Programa pincipal
void main(void) {

ADCON1=0x0F;           //Todos los pines son entradas y salidas digitales
CMCON=0x07;           // Comparadores apagados, puerto A como entrada/salida

TRISB5 = 0;           // PORTB5 = Salida = LINEA

TRISE2 = 1;           // PORTE2 = Entrada = SensadoLinea

TRISC2 = 0;           //PORTC2 = HabilitaAudio = Salida

```

```

TRISB0 = 1;
TRISB1 = 1;
TRISB2 = 1;      //Entradas que vienen del DTMF
TRISB3 = 1;
TRISB4 = 1;

TRISA0 = 0;
TRISA1 = 0;      //Salidas hacia la plataforma vocal
TRISA2 = 0;
TRISA3 = 0;

TRISA4 = 0;
TRISA5 = 0;      //Salidas de los relay
TRISC0 = 0;
TRISC1 = 0;

TRISE0 = 0;      //Salida ENA (habilitación) hacia el display
TRISE1 = 0;      //Salida RS (selector de registro) hacia el display
TRISD = 0x00;    // EL puerto D es la salida de los datos del display

TRISC6 = 1;      //Pulsador = entrada

C=0;
Z = 0;
m=0;

Ring = 0;
C0 = 0;
C1 = 0;
n = 0;

LineaTomada = 0;

DIGITO[0] = 0;
DIGITO[1] = 0;
DIGITO[2] = 0;
DIGITO[3] = 0;

```



```
PV0 = 0;
PV1 = 0;
PV2 = 0;
PLAYE = 1;
```

```
LINEA=0;
```

```
relay1=0;
relay2=0;           //Todos los relay cerrados
relay3=0;
relay4=0;
```

```
i = 0;
```

```
HabilitaAudio = 0;
```

```
contadordodigoincorrecto = 0;
```

```
lcdinit();           // Inicializo el display
```

```
CONT[0] = eeprom_read (0x00);
CONT[1] = eeprom_read (0x01); //Lee la contraseña de la EEPROM
CONT[2] = eeprom_read (0x02);
CONT[3] = eeprom_read (0x03);
```

```
Nrings = eeprom_read (0x04); //Lee la cantidad de rings de la EEPROM
```

```
while(1){
clear();
presentacion()
```

```

//Se sensa el pulsador
while(LineaTomada == 0){
    if(BOTON = 1){
        delay1seg();
        if(BOTON=1){
            clear();
            display_string("Ingrese cantidad");
            line2();
            display_string("de rings:");
            ingresodisplay(&C,&Z);
            IngresoRings (&CringsDTMF, &Nrings);
            eeprom_write(0x04, Nrings);
            delay3seg();
            clear();
            presentacion();
        }else{
            clear();
            display_string("Nueva clave");
            line2();
            for(m=0;m<4;m++)
            {
                ingresodisplay(&C,&Z);
                cambioASCII(&m,&CONT[m]);
            }

            eeprom_write(0x00,CONT[0]);
            eeprom_write(0x01,CONT[1]);
        }
    }
}

```

```

eeprom_write(0x02,CONT[2]);
eeprom_write(0x03,CONT[3]);

delay3seg();

clear();
display_string("Clave ");
line2();
display_string("actualizada!");
delay3seg();
clear();

presentacion();
}
}

```

//Programa que sensa la linea, esperando una llamada entrante

```

__delay_ms(90);
if(SensadoLinea==1){
    C1=C1+1;
    C0=0;
    if(C1==0x03){
        Ring=Ring+1;
        if (Ring==Nrings){
            LINEA=1;
            delay1seg();
            HabilitaAudio = 1;
            LineaTomada = 1;
            Ring=0;
            C1=0;

```

```

        C0=0;
        reproducir (0,0,0);      ///"Ingrese la contraseña"
    }
}
}else{
    C0=C0+1;
    C1=0;
    if(C0==0x43){
        Ring=0;
        C0=0;
    }
}
}
}

```

//Fin del programa de sensado de línea

```

esperadigito(&C1,&C0, &n);

if(LineaTomada == 1){

    ingresodigito(&n, &DIGITO[n]);

    if(n==Nmax){

        if(DIGITO[0] == CONT[0] && DIGITO[1] == CONT[1] && DIGITO[2] == CONT[2]
&& DIGITO[3] == CONT[3]){

            reproducir (0,1,0); ///"Ingrese el código"

            //no reseteo C0 y C1 porque vuelvo a esperar dígito de esta manera no
confunde el ultimo dígito de la contraseña con el primer dígito del código

            n=0;

            while(contadorcodigoincorrecto<2 && LineaTomada == 1){

                esperadigito(&C1,&C0,&n);

                if(LineaTomada==1){

                    ingresodigito(&n, &DIGITO[n]);

                    if (n==2){

```

//no reseteo C0 y C1 porque vuelvo a esperar digito de esta manera no confunde el último digito del ultimo código con el primer digito del nuevo código

```
n=0;
switch(DIGITO[1]){ //DIGITO[1] es el segundo digito
case 0x01:
    switch(DIGITO[0]){ //DIGITO[0] es el primer digito
        case 0x01:
            reproducir(0,0,1); //"actuador activado"
            relay1=1;
            break;
        case 0x02:
            reproducir(0,0,1); //"actuador activado"
            relay2=1;
            break;
        case 0x03:
            reproducir(0,0,1); //"actuador activado"
            relay3=1;
            break;
        case 0x04:
            reproducir(0,0,1); //"actuador activado"
            relay4=1;
            break;
        default:
            reproducir(1,1,0); //"Código Incorrecto"
            contadorcodigoincorrecto = contadorcodigoincorrecto + 1;
            break;
    }
}
break;
case 0x0A:
    switch(DIGITO[0]){
        case 0x01:
```

```

        reproducir(1,0,1);    //"Actuador desactivado"//

        relay1=0;

    break;

    case 0x02:

        reproducir(1,0,1);    //"Actuador desactivado"//

        relay2=0;

    break;

    case 0x03:

        reproducir(1,0,1);    //"Actuador desactivado"//

        relay3=0;

    break;

    case 0x04:

        reproducir(1,0,1);    //"Actuador desactivado"//

        relay4=0;

    break;

    default:

        reproducir(1,1,0);    //"Código Incorrecto"//

        contadorcodigoincorrecto=contadorcodigoincorrecto+1;

    break;

}

break;

default:

    reproducir(1,1,0);    //"Codigo Incorrecto"//

    contadorcodigoincorrecto=contadorcodigoincorrecto+1;

    break;

} //del primer switch

} // del if n = 2

} //del if(LineaTomada == 1)

clear();

presentacion();

```

```

        } //del while(contadorcodigoincorrecto=2)

//Delay para q se escuche el mensaje antes de cortar//
        delay3seg();
        HabilitaAudio = 0;
        LINEA=0;
        LineaTomada = 0;

//no reseteo C0 y C1 porque vuelvo a esperar digito. de esta manera no confunde el ultimo
digito del ultimo codigo con el primer digito del codigo

        n=0;
        contadorcodigoincorrecto = 0;

    }else{
        reproducir(1,0,0);    //"Contraseña incorrecta"//
        //delay de 3 segundo para que termine de reproducir el mensaje antes que
corte//

        delay3seg();
        HabilitaAudio=0;
        LINEA=0;
        LineaTomada = 0;

//no reseteo C0 y C1 porque vuelvo a esperar digito de esta manera no confunde el ultimo
digito de la contraseña con el primer digito de la contraseña

        n=0;
    }
}
} //del if(LineaTomada == 1)
}
}

void ingresodigito (unsigned char *punteron, unsigned char *punteroDIGITO)
{
    *punteroDIGITO = D3;

```

```

    *punteroDIGITO = (*punteroDIGITO<<1) | D2;
    *punteroDIGITO = (*punteroDIGITO<<1) | D1;
    *punteroDIGITO = (*punteroDIGITO<<1) | D0;
    *punteron = *punteron + 1;
}

void esperadigito (unsigned int *punteroC1, unsigned int *punteroC0, unsigned char *punteron)
{
    while(1){
        __delay_ms(15);
        if (DV==1) {
            *punteroC1 = *punteroC1 + 1;
            if (*punteroC1==2) {
                *punteroC0=0;
                return;
            }
        }else{
            *punteroC0 = *punteroC0+1;
            if (*punteroC0>1){
                *punteroC1=0;
                if (*punteroC0==0x29B){ //10 segundos
                    *punteroC0 = 0;
                    *punteroC1 = 0;
                    *punteron = 0;
                    HabilitaAudio = 0;
                    LINEA = 0; //Si no ingresa digitos por 10 segundos entonces corta
                    LineaTomada = 0;
                    return;
                }
            }
        }
    }
}

```



```

    }
}

void delay3seg(void)
{
    for(i=0;i<37;i++){
        __delay_ms(90);
    }
}

void reproducir (unsigned char vPV0,unsigned char vPV1, unsigned char vPV2)
{
    PV0=vPV0;
    PV1=vPV1;           //”actuador desactivado”
    PV2=vPV2;
    PLAYE=0;
    __delay_ms(40);
    PLAYE=1;;
}

void delay1seg(void)
{
    for(i=0;i<11;i++){
        __delay_ms(90);
    }
}

// Inicializa el display en el modo de 8 bits
void lcdinit(void) {
    __delay_ms(80);           // 80ms de delay luego de prender el dispositivo
    command(0x30);

    __delay_ms(7);           // Delay de más de 4.1ms
    command(0x30);

    __delay_ms(1);           // Delay de mas de 1ms
    command(0x30);
}

```

```

    __delay_ms(20);           // Delay de más de 10 ms
    command(0x3C);           // Interfaz de 8 bits. Fuente 5x8

    __delay_ms(1);           // Delay de más de 39us
    command(0x0C);           // Display prendido, sin cursor, sin parpadeo

    __delay_ms(1);           // Delay de más de 39us
    command(0x01);           // Display clear

    __delay_ms(20);          // Delay de más de 15ms
    command(0x06);           // Incremento del cursor a la derecha

    __delay_ms(2);           // Delay de más de 1.53ms
}

```

//Escribir en la segunda linea

```

void line2(void)
{
    command(0xC0);
}

```

//Clear display

```

void clear(void)
{
    command(0x01);
}

```

// Mueve el cursor a la posicion inicial

```

void home(void)
{

```

```

        command(0x02);
    }

    // Ingresar comando al display

    void    command(unsigned char lcd_comm)
    {
        RS = 0;                //Modo ingreso de comando
        PORTD = lcd_comm;      //Comando en el PORTD
        ENA = 1;
        __delay_us(1);        //pulso en la línea de habilitación
        ENA = 0;
        __delay_ms(2);
        RS = 1;                //Modo ingreso de caracter
    }

```

//Enviar un solo caracter al display

```

    void    display(unsigned char lcd_data)
    {
        RS = 1;                //Modo ingreso de caracter
        PORTD = lcd_data;      //Caracter en el PORTD
        ENA = 1;
        __delay_us(1);        //pulso en la línea de habilitación
        ENA = 0;
        __delay_ms(1);
    }

```

//Enviar texto al display

```
void display_string(char string[32])
{
    for(x=0;x<strlen(string);x++)
    {
        display(string[x]);           // Muestra los caracteres uno a uno
        if(x==15){line2();}          // Si el mensaje tiene mas de 16 caracteres pasa
a la linea 2
    }
}
```

//Ingreso de nueva contraseña y mostrarla en el display

```
void cambioASCII (unsigned char *m, unsigned char *CONT)
{
    *CONT = D3;
    *CONT = (*CONT<<1) | D2;
    *CONT = (*CONT<<1) | D1;
    *CONT = (*CONT<<1) | D0;

    switch(*CONT){
        case 0b00001010:           // En el caso que ingrese 0
            display_string("0");
            break;
        case 0b00001011:           // En el caso que ingrese *
            display_string("*");
            break;
        case 0b00001100:
```

```
        display_string("#");           //En el caso que ingrese #
break;
case 0b00000001:
    display_string("1");
break;
case 0b00000010:
    display_string("2");
break;
case 0b00000011:
    display_string("3");
break;
case 0b00000100:
    display_string("4");
break;
case 0b00000101:
    display_string("5");
break;
case 0b00000110:
    display_string("6");
break;
case 0b00000111:
    display_string("7");
break;
case 0b00001000:
    display_string("8");
break;
case 0b00001001:
    display_string("9");
break;
}
```

```
}
```

```
//Espera que el usuario ingrese digito de la nueva contraseña
```

```
void ingresodisplay (unsigned int *C, unsigned int *Z)
```

```
{
```

```
    while(1){
```

```
        __delay_ms(15);
```

```
        if(DV==1){
```

```
            *C = *C + 1;
```

```
            if (*C==2){
```

```
                *Z=0;
```

```
                return;
```

```
            }
```

```
        }else{
```

```
            *Z=*Z+1;
```

```
            if(*Z>1){
```

```
                *C=0;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
//Presenta el estado de los relays en la pantalla del display
```

```
void presentacion (void)
```

```
{
```

```
    display_string("A1=");
```

```
    if (relay1==1 ){
```

```
        display_string("ON  ");
```

```
    }else{
```

```

        display_string("OFF ");
    }

    display_string("A2=");
    if (relay2 == 1 ){
        display_string("ON");
    }else{
        display_string("OFF");
    }

    line2();

    display_string("A3=");
    if (relay3 == 1 ){
        display_string("ON ");
    }else{
        display_string("OFF ");
    }

    display_string("A4=");
    if (relay4 == 1 ){
        display_string("ON");
    }else{
        display_string("OFF");
    }
}

```

//Ingresa y muestra el digito ingresado por el usuario en la programacion de la cantidad de rings

```
void IngresoRings (unsigned char *CringsDTMF, unsigned char *Nrings)
```

{

```
*CringsDTMF = D3;  
*CringsDTMF = (*CringsDTMF<<1) | D2;  
*CringsDTMF = (*CringsDTMF<<1) | D1;  
*CringsDTMF = (*CringsDTMF<<1) | D0;
```

```
switch(*CringsDTMF){  
    case 0b00000001:  
        display_string("1");  
        *Nrings = *CringsDTMF;  
        break;  
    case 0b00000010:  
        display_string("2");  
        *Nrings = *CringsDTMF;  
        break;  
    case 0b00000011:  
        display_string("3");  
        *Nrings = *CringsDTMF;  
        break;  
    case 0b00000100:  
        display_string("4");  
        *Nrings = *CringsDTMF;  
        break;  
    case 0b00000101:  
        display_string("5");  
        *Nrings = *CringsDTMF;  
        break;  
    case 0b00000110:  
        display_string("6");  
        *Nrings = *CringsDTMF;
```



```
break;

case 0b00000111:
    display_string("7");
    *Nrings = *CringsDTMF;
break;

case 0b00001000:
    display_string("8");
    *Nrings = *CringsDTMF;
break;

case 0b00001001:
    display_string("9");
    *Nrings = *CringsDTMF;
break;

default:
    clear();
    display_string("Solo digitos");
    line2();
    display_string("del 1 al 9");
}

}
```

Referencias

- [1] J. D. Cabezas Pozo, "Sistemas de Telefonía", International Thomson Ediciones Spain Paraninfo, S.A, 2007.
- [2] Inzirillo Roberto, Quero José, "El aparato telefónico", Cátedra Sistemas de Telefonía, Universidad de Mendoza.
- [3] PepeTronic Inc. Server, Telefonía y sistemas electrónicos de telemando, <http://www.repuestoelectronico.com/archivos/Telefonia-y-telemando.pdf>, pp. 3-5.
- [4] Holtek Semiconductor Inc, Hoja de datos: HT9170 DTMF Receiver, 1999.
- [5] Winbond Electronics Corp, Hoja de datos: ISD1400 Series, 2004.
- [6] Winstar Display Co, nota de aplicación: WH1602B 2x16 display.
- [7] Rectron Semiconductor, Hoja de datos: DB157 Single phase bridge rectifiers.
- [8] Motorola Semiconductor, Hoja de datos: 4N27 6-pin DIP optoisolators transistor output, 1995.
- [9] Motorola Semiconductor, Hoja de datos: ULN2803 octal high voltage, high current Darlington transistor arrays, 1996.
- [10] Microchip Technology Inc, hoja de datos: PIC18F4550 40 pin, high-performance, enhanced flash, USB Microcontrollers with nanoWatt technology, 2009.
- [11] Microchip Technology Inc, Guia de usuario: MPLAB X IDE, 2011-2014.
- [12] Microchip Technology Inc, Guia de usuario: MPLAB XC8 compiler, 2012.
- [13] G. Galeano. Programación de sistemas embebidos en C. Alfaomega, 1º Edición, 2009.
- [14] Microchip Technology Inc, Guia de usuario: PICkit 3 programmer/debugger, 2010.