

**DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO ELECTRÓNICO
PORTÁTIL PARA LA COMUNICACIÓN EN TIEMPO REAL,
ENTRE PERSONAS SORDAS A TRAVÉS DE LA RED CELULAR**

CARLOS ARTURO PALACIOS CÓRDOBA

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO**

2013

**DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO ELECTRÓNICO
PORTÁTIL PARA LA COMUNICACIÓN EN TIEMPO REAL,
ENTRE PERSONAS SORDAS A TRAVÉS DE LA RED CELULAR**

CARLOS ARTURO PALACIOS CÓRDOBA

**Trabajo de Grado presentado como requisito parcial para optar
al título de Ingeniero Electrónico**

**ASESOR
ROLANDO BARAHONA CABRERA
INGENIERO ELECTRÓNICO**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO**

2013

NOTA DE RESPONSABILIDAD

“La Universidad de Nariño no se hace responsable por las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”.

Acuerdo 1. Artículo 324. Octubre 11 de 1966. Emanado del honorable Consejo Directivo de la universidad de Nariño.

Nota de Aceptación:

Firma del Jurado

Firma del Jurado

Firma del Jurado

San Juan de Pasto, 15 de agosto de 2013

DEDICATORIA

A Dios por permitirme realizar uno de mis sueños y a mi familia, cuyo apoyo incondicional me da fuerzas para seguir adelante.

AGRADECIMIENTOS

El autor expresa sus agradecimientos a:

Yolanda Alfaro, Coordinadora del aula de apoyos tecnológicos para personas con discapacidad de la Universidad de Nariño, quien permitió la vinculación del proyecto para el beneficio de sus estudiantes.

CONTENIDO

	Pág.
INTRODUCCIÓN	16
1. MARCO REFERENCIAL.....	18
1.1 MARCO CONTEXTUAL	18
1.2 MARCO TEÓRICO	18
1.2.1 Principios de funcionamiento de la telefonía celular	19
1.2.2 Principios de funcionamiento de la tecnología Bluetooth.....	25
1.3 MARCO CONCEPTUAL	33
1.3.1 Tonos DTMF	33
1.3.2 Comandos AT	35
2. METODOLOGÍA	37
2.1 PROCESOS PARA EL CONTROL DEL TELÉFONO CELULAR	37
2.1.1 Configuración del módulo BlueSMiRF	39
2.1.2 Conexión entre el teléfono y el módulo BlueSMiRF	42
2.2 TRANSMISIÓN DE TONOS DTMF DURANTE UNA LLAMADA DE VOZ.....	43
2.2.1 Envío de tonos DTMF	44
2.2.2 Recepción de tonos DTMF	45
2.3 DISEÑO DE CIRCUITOS PARA EL DISPOSITIVO ELECTRÓNICO	48

2.3.1 Circuito para decodificación de tonos DTMF	49
2.3.2 Interfaz de usuario	50
2.3.3 Funciones del microcontrolador PIC16F876A.....	52
2.3.4 Funciones del microcontrolador PIC16F628A.....	53
2.3.5 Dispositivos BlueSMiRF y ZTE N281	53
2.4 SIMULACIÓN DE LOS CIRCUITOS DISEÑADOS MONTAJE DE PROTOTIPOS	54
2.5 IMPLEMENTACIÓN DE 2 DISPOSITIVOS ELECTRÓNICOS	55
2.6 CAPACITACIÓN A UNA PERSONA CON DISCAPACIDAD AUDITIVA PARA USAR EL DISPOSITIVO	56
3. RESULTADOS.....	58
3.1 CONTROL DEL TELÉFONO CELULAR ZTE N281	58
3.2 SIMULACIÓN HÍBRIDA Y PROTOTIPO DEL DISPOSITIVO ELECTRÓNICO	58
3.3 CARACTERÍSTICAS DE LOS DISPOSITIVOS ELECTRÓNICOS.....	60
3.4 RESULTADOS DE LA CAPACITACIÓN A USUARIOS SORDOS.....	62
4. CONCLUSIONES	63
RECOMENDACIONES.....	64
BIBLIOGRAFÍA.....	65
ANEXOS.....	67

LISTA DE TABLAS

	Pág.
Tabla 1. Frecuencias de GSM	21
Tabla 2. Circuitos de RS232 emulados Por RFCOMM	30
Tabla 3. Pares de frecuencias empleadas para la generación DTMF	33
Tabla 4. Principales pines de conexión del BlueSMiRF	39
Tabla 5. Principales pines de conexión de la tarjeta FT232RL	40
Tabla 6. Representación de caracteres mediante tonos DTMF	43
Tabla 7. Pines principales del circuito integrado MT8870DE	46
Tabla 8. Datos digitales en Q1-Q4 de acuerdo al tono DTMF presente.	47
Tabla 9. Pines Principales de pantalla LCD	52
Tabla 10. Costos de fabricación del dispositivo electrónico	62

LISTA DE FIGURAS

	Pág.
Figura 1. Principales elementos de la red móvil AMPS	20
Figura 2. Estructura principal de la red GSM	22
Figura 3. Estructura estación móvil (MS)	23
Figura 4. Estados de una conexión Bluetooth	27
Figura 5. Pila de protocolos Bluetooth	28
Figura 6. Perfiles comunes Bluetooth	31
Figura 7. Aspecto Físico ZTE N281	37
Figura 8. Aspecto Físico BlueSMiRF	38
Figura 9. Circuito Esquemático BlueSMiRF	39
Figura 10. Aspecto Físico de la Tarjeta FT232	40
Figura 11. Conexión para configuración del módulo BlueSMiRF	41
Figura 12. Configuración del puerto emulado por la tarjeta FT232	41
Figura 13. Secuencia de comandos configuración BlueSMiRF	42
Figura 14. Procesos de indagación y conexión entre el modulo y el teléfono	42
Figura 15. Aspecto físico del circuito integrado MT8870DE.	45
Figura 16. Diagrama de bloques del circuito integrado MT8870DE fabricado por Zarlink Semiconductor.	46

Figura 17. Diagrama de bloques principal del dispositivo electrónico.....	48
Figura 18. Circuito decodificador de tonos DTMF.....	49
Figura 19. Circuito de teclado matricial 6x6.....	50
Figura 20. Diagrama de flujo control de teclado matricial.....	51
Figura 21. Aspecto físico de la pantalla LCD alfanumérica 16x2.....	52
Figura 22. Circuito creado en Proteus para simulación.....	54
Figura 23. Circuito impreso principal para el dispositivo electrónico.....	55
Figura 24. Circuito impreso para el teclado matricial tipo QWERTY.....	56
Figura 25. Comunicación entre personas sordas mediante el dispositivo.	57
Figura 26. Conexión para simulación híbrida mediante Proteus.....	59
Figura 27. Aspecto físico prototipo de dispositivo electrónico.....	60
Figura 28. Aspecto físico de los dispositivos electrónicos.	61

LISTA DE ANEXOS

	Pág.
Anexo A. Especificaciones del microcontrolador PIC16F876A	68
Anexo B. Especificaciones del microcontrolador PIC16F628A	69
Anexo C. Código en lenguaje C para el PIC16F876A	70
Anexo D. Código en lenguaje C para el PIC16F628A	87
Anexo E. Circuito esquemático del dispositivo electrónico implementado	96
Anexo F. Formato de encuesta realizada a personas sordas	97
Anexo G. Manual de usuario para el dispositivo electrónico.....	99

GLOSARIO

AT: (Attention). Sigla utilizada para definir comandos específicos para control de dispositivos terminales. Su sintaxis se basa en petición y respuesta.

BLUETOOTH: especificación para redes inalámbricas de área personal que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda de los 2,4 GHz.

BPS: (Bits por segundo). Unidad utilizada para expresar la velocidad de transmisión de impulsos (0 y 1) en un segundo.

DTMF: (Dual Tone Multi-Frequency). Tonos compuestos por la suma algebraica de dos señales sinusoidales a frecuencias diferentes, utilizados para la transmisión de datos en el rango de frecuencias de la voz humana.

GSM: (Global System for Mobile Communications). Sistema de telefonía móvil digital de segunda generación más extendido del mundo.

MICROCONTROLADOR: circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto principalmente por la unidad central de procesamiento, memoria y periféricos de entrada/salida.

POSTLOCUTIVO: persona oyente que por diferentes causas, ha perdido la audición, total o parcialmente, después de adquirir el lenguaje.

SORDO: Persona incapaz o que tiene problemas para escuchar.

TTL: (Transistor-Transistor-Logic). Tecnología de construcción de circuitos electrónicos digitales basados en transistores, cuya tensión de alimentación está comprendida entre los 4.75V y los 5.25V.

WI-FI: conjunto de estándares para redes inalámbricas basado en las especificaciones IEEE 802.11.

RESUMEN

Este proyecto, desarrollado en modalidad de aplicación, tiene como finalidad dar solución al problema que presentan las personas con discapacidad auditiva quienes no pueden acceder a los servicios de llamadas de voz mediante la red celular. Para esto, se diseñó e implementó un dispositivo electrónico portátil, que usa la red celular para transmitir mensajes escritos mediante llamadas de voz, es decir, sin necesidad de adquirir planes de datos o servicios especiales que impliquen costos adicionales al usuario. De esta forma, se realizó un circuito para lograr la conexión entre el dispositivo electrónico y un teléfono celular mediante tecnología Bluetooth con el fin de tomar el control de su módulo GSM/GPRS y por medio de comandos AT realizar el envío de tonos DTMF. Así mismo se realizó el circuito correspondiente para la decodificación y visualización en pantalla de los tonos DTMF recibidos en el dispositivo. Como resultados del proyecto se implementaron dos dispositivos electrónicos portátiles que permiten el acceso a llamadas de voz para los usuarios con discapacidad auditiva acarreando beneficios como disponibilidad, portabilidad y economía. Cabe aclarar que el proyecto está ligado al aula de apoyos tecnológicos para personas con discapacidad de la Universidad de Nariño, donde su coordinadora y estudiantes se integraron al desarrollo del proyecto en las etapas de capacitación y pruebas de funcionamiento del dispositivo electrónico.

ABSTRACT

This project was developed in application mode and aims to solve the problem presented by people with hearing disabilities who cannot access the services of voice calls over the cellular network. For this, it was designed and implemented a portable electronic device, which uses the cellular network to transmit written messages through voice calls, this means no data plans or special services that involve additional costs to the user. Thus, a circuit was performed to get the connection between the electronic device and a cell phone using Bluetooth technology in order to take control of its GSM / GPRS module and through AT commands send DTMF tones. It was also made the circuit for decoding and display on screen of DTMF tones received on the device. As a result of the project were implemented two portable electronic devices that allow access to voice calling for hearing impaired users carrying benefits such as availability, portability and economy. The project is linked to classroom technology support for people with disabilities at the University of Nariño, where the coordinator and students joined the stages of training and performance testing of the electronic device

INTRODUCCIÓN

En los últimos años, el desarrollo de las telecomunicaciones se ha fortalecido con su constante crecimiento, sorprendiendo con la creación de nuevas tecnologías y dispositivos que permiten una comunicación con múltiples opciones y formas: voz, video o texto. Por consiguiente, las comunicaciones móviles en la actualidad han tomado gran atención y acogida por parte de los usuarios, siendo, el estándar GSM el más extendido del mundo al contar con más de 3000 millones de usuarios en más de 212 países. Lo anterior debido a las necesidades que demanda la actual sociedad de establecer una comunicación entre personas en cualquier lugar del mundo, sin importar si está en su hogar, vehículo, o en cualquier otra parte.

En la misma medida, las empresas de telefonía móvil se han concentrado en ofrecer a los usuarios en general, gran cantidad de teléfonos, dispositivos y servicios de última tecnología sin tener en cuenta que los servicios ofrecidos a los usuarios con discapacidad auditiva tienen un costo adicional que implica una dependencia de la capacidad económica de los usuarios para acceder a estas ayudas tecnológicas.

Frente a lo anterior, para que una persona con limitaciones auditivas pueda comunicarse a través de la red celular en tiempo real, debe hacer uso del “chat” o una video llamada, pagando un plan de datos para acceder a este servicio o adquiriendo un teléfono de gama alta con Wi-Fi lo que limita la portabilidad de la comunicación puesto que el usuario debe estar dentro del rango de la red.

Al respecto, los casos descritos anteriormente ponen en evidencia algunas de las falencias y limitaciones de los servicios prestados a una población discapacitada contextualizada a la ciudad de Pasto, en donde no se les presta mucha atención a este tipo de necesidades y no se ofrecen soluciones alternativas para que las personas sordas logren acceder y utilizar los medios de comunicación que se ofrecen actualmente de forma eficaz y sin que represente costos adicionales mensuales.

De esta manera, el presente proyecto tiene como objetivo general diseñar e implementar un dispositivo electrónico portátil, que permita una comunicación a distancia y en tiempo real a través de la red celular entre personas de la ciudad de Pasto que posean discapacidad auditiva, para que dichas personas puedan utilizar la tecnología de vanguardia sin incurrir en costos adicionales mensuales.

Así, para lograr dicho objetivo, el desarrollo del proyecto se divide en varias fases:

- Diseñar los circuitos electrónicos pertinentes tanto para el envío como para la recepción de tonos DTMF a través del teléfono celular de gama media.
- Diseñar los circuitos para la interfaz de usuario, con la que se envía y visualiza el texto digitado o recibido por el usuario discapacitado.
- Implementar dos dispositivos electrónicos para establecer una llamada de prueba a través de la red celular.
- Indagar sobre las implicaciones éticas para realizar las pruebas pertinentes sin incurrir en violaciones a los derechos fundamentales de la persona con discapacidad auditiva.
- Capacitar a una persona con discapacidad auditiva, para que logre manipular el dispositivo electrónico y pueda establecer una comunicación con otra persona, discapacitada o no, a través de la red celular.

Al respecto de la capacitación, por ser un proyecto en modalidad de aplicación, esta se realiza en el aula de apoyos tecnológicos para personas con discapacidad de la Universidad de Nariño donde se fomenta un gran interés por el desarrollo del presente proyecto, comprendiendo que existirá una nueva forma de comunicación entre sus integrantes sin limitar las relaciones interpersonales dentro del plantel educativo.

Cabe aclarar que el proyecto tiene utilidad siempre y cuando, los usuarios sordos sean capaces de interpretar el lenguaje oral (Sordos postlocutivos), debido a que se usan teclas alfanuméricas para la interfaz de usuario.

1. MARCO REFERENCIAL

1.1 MARCO CONTEXTUAL

El desarrollo del presente proyecto tiene cabida en Colombia, principalmente en el departamento de Nariño y en la ciudad de Pasto donde habitan alrededor de 130 personas con discapacidad auditiva¹ de las cuales aproximadamente 24 asisten al aula de apoyos tecnológicos para personas con discapacidad de la Universidad de Nariño, cuyo objetivo es desarrollar acciones que promuevan la inclusión educativa y social de las personas con sordo ceguera y discapacidad múltiple, a través de recursos tecnológicos, apoyo social y actividades recreativas y culturales que permitan potencializar habilidades, destrezas y capacidades para que se sientan útiles e importantes en la sociedad.

Hoy en día, aún se encuentran muchos municipios y corregimientos del departamento de Nariño que no tienen desplegadas redes celulares de tercera generación (3G). Esto implica que no hay acceso a servicios como internet móvil, chat o video llamadas lo que hace necesario el uso de servicios como llamadas de voz, dejando a las personas sordas sin opciones para comunicarse a distancia mediante la red celular. Siendo esta una razón importante que justifica el desarrollo del proyecto para esta población.

Finalmente, se hace necesario mencionar que la mayoría de personas con discapacidad auditiva asistentes al aula de apoyos tecnológicos, están familiarizadas con el lenguaje oral y se encuentran en plena capacidad de leerlo y escribirlo. Esto significa un lugar óptimo para la aplicación del proyecto.

1.2 MARCO TEÓRICO

Para comenzar con el desarrollo del proyecto, se debe realizar un estudio detallado acerca de las principales tecnologías utilizadas como base para el funcionamiento del dispositivo electrónico.

¹ Diario del Sur. (2010). En Pasto falta apoyo a las personas sordas [Artículo en línea]. <http://www.mineduccion.gov.co/observatorio/1722/article-248605.html> (23 de junio de 2013)

Estas tecnologías, son principalmente, la telefonía celular puesto que es el medio que utiliza el dispositivo para la transmisión de las señales de voz y la tecnología Bluetooth, por medio de la cual se realiza la comunicación entre el dispositivo electrónico y el teléfono celular.

1.2.1 Principios de funcionamiento de la telefonía celular. Dada la necesidad que tienen las sociedades de comunicarse desde cualquier lugar, nace la telefonía celular, cuya característica principal es la portabilidad, y por su gran acogida, se convierte en uno de los medios de comunicación más aceptado y extendido del mundo. Es así como, a través de los años se han logrado grandes avances que permiten realizar cada vez un mayor número de tareas desde un mismo dispositivo; sin embargo aún se conservan sus principios de funcionamiento.

Así, para entender sus principios o bases de funcionamiento se hace necesario indagar acerca de cómo surge y evoluciona la telefonía celular puesto que cada avance aporta conceptos necesarios para entender su modo de operación.

Los inicios de la telefonía celular se dan a finales de los años Setenta y comienzos de los ochenta, donde se despliegan redes analógicas de primera generación (1G) usadas exclusivamente para transmisión de voz. Estas redes son la base de los avances actuales puesto que implementan una estructura de múltiples celdas que se enlazan con la torre de cobertura para mantener la comunicación mientras el usuario está en movimiento.

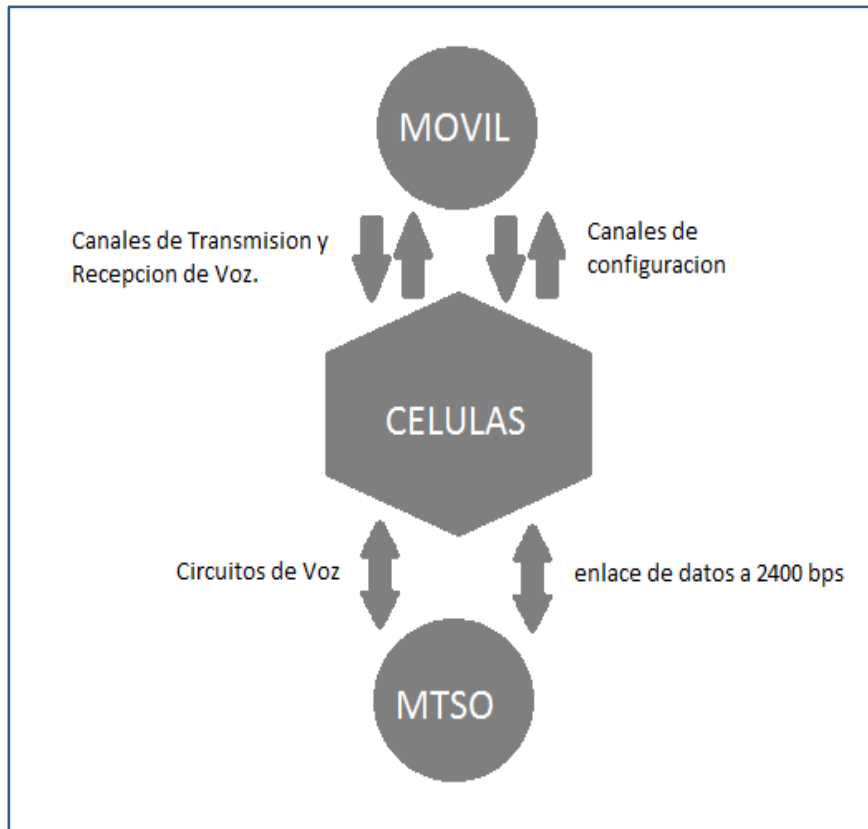
El estándar principal 1G se denomina sistema avanzado de telefonía móvil AMPS (Advanced Mobile Phone System) donde las células se interconectan con la Oficina de Conmutación de Telefonía móvil MTSO (Mobile Telecommunications Switching Office) y las señales de voz son llevadas mediante troncales de voz. Así mismo hay enlaces de datos a 2400 bps entre la MTSO y las células para propósitos de señalización.

Entre sus principales características² se tiene la modulación FM analógica con un ancho de banda de 30 kHz y con capacidad para 832 canales dúplex, de los cuales 21 son reservados para establecer la llamada, y el resto para la comunicación por voz.

De acuerdo a lo anterior, en la figura 1 se puede observar como es la estructura del sistema AMPS.

² Wireless Communication. (1995). Advanced Mobile Phone System (AMPS). <http://www.wirelesscommunication.nl/reference/chaptr01/telephon/amps.htm> (Octubre 26 de 2012)

Figura 1. Principales elementos de la red móvil AMPS



Fuente: RAMASAMI, Vijaya. Advanced mobile phone system an Overview. Pag 2 [en línea].

Posteriormente, alrededor de los años 90 se reemplazan los sistemas de transmisión análogos (1G) por sistemas de transmisión digitales de segunda generación (2G), buscando dar un mejor manejo de llamadas en casi los mismos espectros de radiofrecuencia ya asignados a la telefonía celular, soportando velocidades más altas para voz pero limitadas en comunicaciones de datos y permitiendo el acceso a nuevos servicios auxiliares como los mensajes cortos de texto SMS (Short Message Service) y mejor capacidad de envío de datos desde dispositivos de fax y módem.

Para los sistemas de segunda generación, se implementan dos estándares muy importantes para el desarrollo actual de la telefonía celular. Estos son: el sistema global para comunicaciones móviles GSM (Global System for Mobile Communications) y el sistema de acceso múltiple por división de código CDMA (Code Division Multiple Access).

- **GSM (Global System for Mobile Communications).** Nace en los años 80, en la CEPT (Conference Europeenne de postal et Télécommunications) donde se forma

un grupo especial de comunicaciones móviles GSM (del francés Groupe Speciale Mobile) el cual luego sería denominado sistema global para comunicaciones móviles GSM (Global System for Mobile Communications)³.

Sus aplicaciones comerciales se dan a inicios de los noventa y es tal su acogida que hoy en día es el estándar más extendido del mundo con más de 3000 millones de usuarios en más de 212 países.

Adicionalmente su principio de funcionamiento se basa en la técnica de acceso múltiple por división de tiempo TDMA (Time Division Multiple Access) comprimiendo la voz y dividiendo cada canal de voz en ocho partes utilizando técnicas de conmutación digital. Así se aumenta la capacidad del sistema ya que puede transmitir un mayor número de llamadas por cada célula.

En un principio se usa el ancho de banda para las frecuencias 850/900 MHz pero luego se extiende a 1800/1900 MHz como se muestra en la tabla 1.

Tabla 1. Frecuencias de GSM

Banda	Nombre	Canales	Uplink MHz	Downlink MHz	Notas
850	GSM 850	128-251	824-849	869-894	Usada en EEUU, Sudamérica y Asia
900	P-GSM 900	0-124	890-915	935-960	Banda con la que nace en Europa
	E-GSM 900	974-1023	880-890	925-935	Extensión del GSM 900
	R-GSM 900	N/A	876-880	921-925	GSM ferroviario
1800	GSM 1800	512-885	1710-1785	1805-1880	Usada en Europa y otros países extranjeros
1900	GSM 1900	512-810	1850-1910	1930-1990	Usada principalmente en EEUU (PCS) incompatible con GSM 1800 por solapamiento de bandas

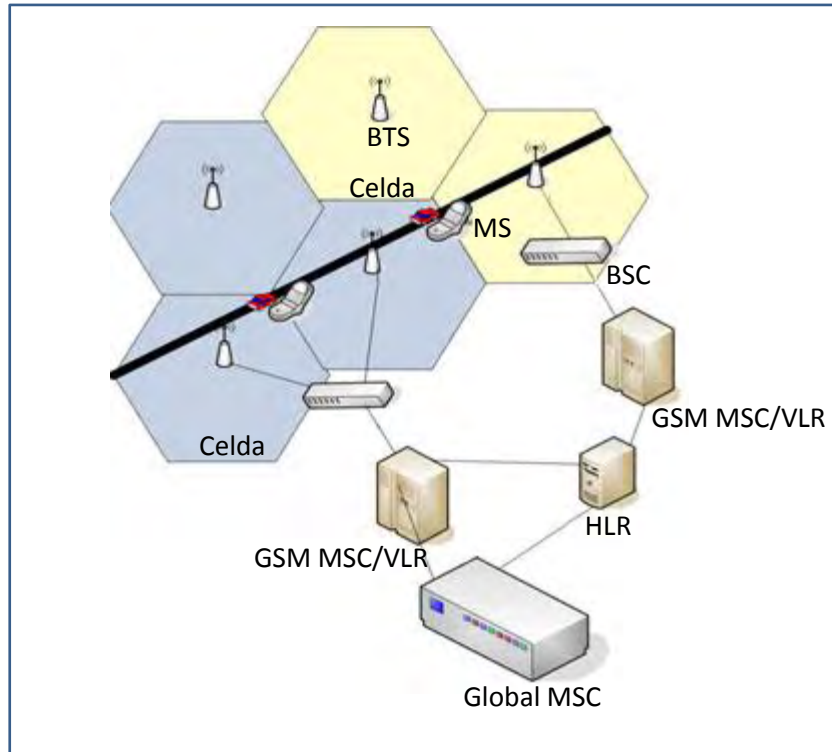
Fuente: NIÑO, Diego. (2011). Telefonía Móvil. [en línea].

De acuerdo con la tabla 1, se tiene que en Colombia, los operadores Claro y Movistar utilizan la banda de 850MHz, mientras que el operador Tigo utiliza la banda de los 1900MHz.

A continuación se presenta en la figura 2 la estructura principal de GSM.

³ GARCIA, Andrés, GARCIA, José. Telefonía Móvil. Escuela Politécnica Superior de Albacete. http://www.infoab.uclm.es/labelec/Solar/Comunicacion/Telefonia_movil/index_archivos/Page1637.htm (Octubre 26 de 2012)

Figura 2. Estructura principal de la red GSM



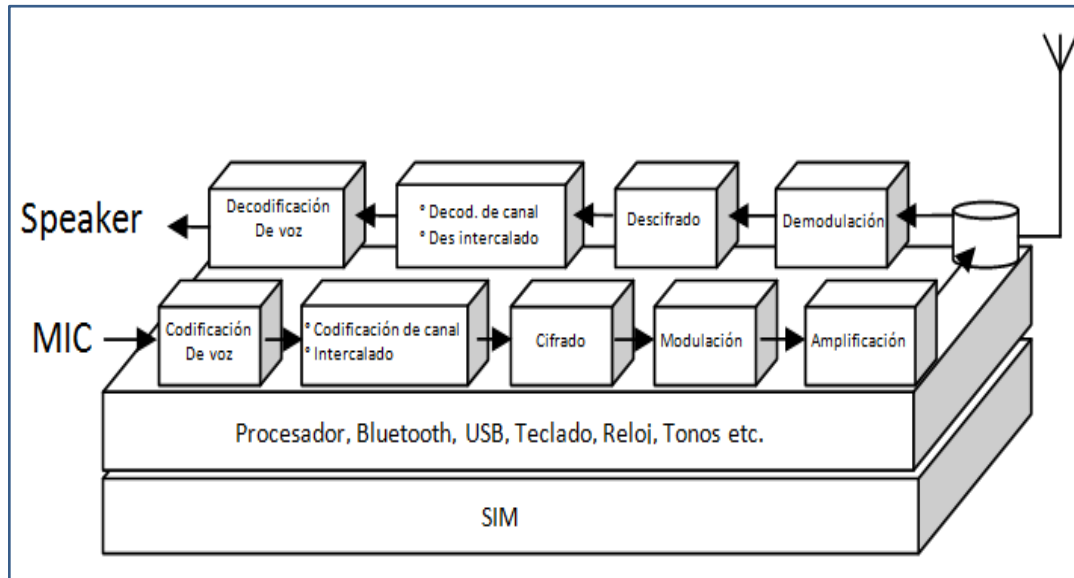
Fuente: EventHelix. GSM Network Example [en línea].

De acuerdo con la figura 2, el sistema GSM se divide en varias estaciones, comenzando por la Estación Móvil (MS, Mobile Station) compuesta por el módulo de identificación de abonado SIM (Subscriber Identity Module) que permite la identificación única del usuario y la terminal móvil por medio del identificador internacional de abonados móviles IMSI (International Mobile Subscriber Identity) y el equipo móvil (teléfono celular) que es identificado mediante el identificador internacional de equipos móviles IMEI (International Mobile Equipment Identity).

La estación móvil debe ser compatible con los servicios que ofrece el sistema GSM tales como llamadas de voz, mensajes de texto, envío y recepción de tonos DTMF y llamadas de emergencia aun cuando no se haya insertado la tarjeta SIM. La MS se comunica con la red celular a través de un vínculo de radio que utiliza TDMA y que es denominado interfaz de aire.

Su estructura puede observarse en la figura 3.

Figura 3. Estructura estación móvil (MS)



Fuente: HEINE, Gunnar. GSM Networks: Protocols, Terminology, and Implementation. Londres: Artech House, 1999. Pag 14.

Luego, la Estación Base del Transceptor (BTS, Base Transceiver Station) es utilizada como interfaz entre la Estación Móvil y la red y está conformada por equipos de radiofrecuencia ubicados en el centro de la célula, los cuales, según su potencia determinan la dimensión de la misma. A su vez, las BTS son controladas por el Controlador de Estación Base (BSC, Base Station Controller) que se encarga de monitorear y controlar los equipos de radiofrecuencia y adicionalmente se encarga de asignar canales y frecuencias de radio, medir la potencias de las estaciones móviles y tomar decisiones cuándo se presenta un Handover, es decir cuando el usuario pasa de una estación base a otra.

Finalmente se cuenta con el Centro de Conmutación Móvil (MSC, Mobile Switching Center) que se ocupa del enrutamiento de las llamadas y la configuración de las mismas. Este se comunica con múltiples BSC y también con otros MSC especiales como la compuerta del Centro de conmutación Móvil (GMSC, Gateway Mobile Switching Center) cuyas funciones son servir como compuerta entre la red GSM y otros tipos de redes como por ejemplo la telefonía fija.

- **Sistema CDMA (Code Division Multiple Access).** Utiliza básicamente la misma estructura que GSM pero cambia drásticamente la comunicación entre la estación móvil y la y la estación base ya que se utiliza un método denominado acceso múltiple por división de código CDMA que aprovecha mejor el espectro y brinda mayor seguridad, puesto que cada usuario ocupa la totalidad del ancho de Banda de un mismo canal, mediante la utilización de un código "aleatorio" que es

asignado a cada usuario, permitiendo que varias señales viajen al mismo tiempo por dicho canal y luego el sistema identifique la señal de interés para transmitir las señales de voz correspondientes.

Posteriormente, se realizan mejoras a los sistemas 2G en cuanto a velocidad de transmisión de datos lo que permite a los usuarios el acceso a internet y a otros servicios como mensajería instantánea. Estas mejoras se conocen como tecnologías 2.5G pero no se adentra en su estudio puesto que no se utilizan para el desarrollo del proyecto.

Luego se implementan redes de tercera generación (3G) las cuales se definen mediante el estándar IMT-2000 dado por la unión internacional de comunicaciones ITU (International Telecommunication Union). Dicho estándar se desarrolla en base al sistema denominado Sistema Universal de Telefonía Móvil UMTS (Universal Mobile Telephone System) caracterizado porque su interfaz aérea se basa en tecnología W-CDMA (Wideband Code Division Multiple Access) garantizando mayores velocidades de transmisión de datos y voz, además de una mejor seguridad de la red ya que usa las bases de CDMA.

Así, en estas redes, además de ofrecer servicios de llamadas de voz, se ofrecen servicios de transmisión de contenidos digitales de texto, Voz, Video, internet y servicios multimedia con tasas de hasta 2Mbps ofreciendo un sistema consistente de servicios móviles para ordenadores y telefonía.

El sistema UMTS está compuesto principalmente por el equipo de usuario (UE, User Equipment) que al igual que en GSM es el equipo que lleva el usuario y en este caso debe ser capaz de comunicarse con la red celular mediante tecnología WCDMA. Dicha comunicación se establece mediante la red de Acceso de Radio UMTS (UTRAN) que consta de varios elementos entre los que se encuentran los RNC (Radio Network Controller) que se encargan de controlar el tráfico de los canales, la potencia de los enlaces, handover entre otras funciones; y los Nodos B (equivalentes a las Estaciones Base BTS en el estándar GSM).

El componente principal del sistema UMTS es la Red Central (Core Network) que está compuesta por dos elementos principales como el MSC (para conmutación de circuitos) que es el mismo MSC usado en la red GSM, por lo tanto cumple las mismas funciones además de ofrecer una interfaz de comunicación con el Nodo de Servicio de Soporte GPRS (SGSN, Serving GPRS Support Node) que es el componente fundamental en una red de conmutación de paquetes y se encarga entre otras cosas de manejar la información de la suscripción, el identificador internacional de abonados móviles IMSI, la información de la ubicación y la célula o área en la que el móvil está registrado.

Para las redes 3G también se realizan mejoras en cuanto a las velocidades de transmisión de datos originales dando origen a redes 3.5G.

Finalmente, en la actualidad, debido a la confluencia entre las redes inalámbricas y cableadas, es posible implementar redes IP (Internet Protocol) aptas para Smartphone, módems inalámbricos, entre otros permitiendo que absolutamente todos los datos incluyendo la voz de las llamadas, sean transmitidos por medio de paquetes conmutados con velocidades teóricas que están por encima de 1Giga Byte por segundo para usuarios estáticos y por encima de 100MBps para usuarios en movimiento. Esta generación se denomina 4G y es fruto del estándar de comunicaciones móviles denominado LTE (Long Term Evolution).

Cabe aclarar que el dispositivo desarrollado en el proyecto utiliza las redes de segunda o tercera generación accediendo a servicios de llamadas de voz para transmitir las señales necesarias para la comunicación.

Por otra parte, el dispositivo electrónico utiliza tecnología Bluetooth para gobernar el módulo GSM/GPRS o 3G de un teléfono celular y establecer llamadas, parámetros de configuración y enviar tonos DTMF a través de la red celular. Por lo tanto se realiza un estudio acerca de sus principios de funcionamiento.

1.2.2 Principios de funcionamiento de la tecnología Bluetooth. En los años 90, la compañía Ericsson Mobile Telecommunications se interesa por realizar estudios sobre redes de área local de corto alcance y bajo coste para dispositivos móviles y sus accesorios. A medida que se profundiza la investigación, se encuentra que las aplicaciones que involucran este tipo de redes son ilimitadas.

Es así como en 1998, otras compañías tales como Nokia, IBM, Toshiba e Intel se unen con Ericsson para formar un Grupo de Interés Especial Bluetooth SIG (Bluetooth Special Interest Group) con el fin de generar compatibilidad entre los dispositivos que utilicen este tipo de redes, generando la primera especificación de Bluetooth que involucra un conjunto de protocolos que dan flexibilidad para que se pueda usar en gran variedad de aplicaciones.

En este orden de ideas, la tecnología Bluetooth se define como el estándar IEEE 802.15.1 desarrollado para la comunicación inalámbrica de datos de corto alcance que opera en la banda de 2.4GHz caracterizándose por su robustez, baja complejidad, bajo consumo y bajo costo.

Los dispositivos Bluetooth se comunican mediante el esquema de conexión maestro esclavo, soportando hasta ocho dispositivos conectados simultáneamente en lo que se denomina una Piconet. Así mismo un dispositivo puede estar presente en varias piconet, aunque solo puede estar activo en una de ellas, formando una red dispersa o Scatternet.

Dentro de una piconet, existe un dispositivo que debe actuar como maestro, enviando señales de reloj para sincronización e información de saltos de

frecuencia para la transmisión de datos. Los demás integrantes de la piconet, actúan como esclavos.

Así, dentro de una red Bluetooth, sea en una piconet o en una Scatternet, dos o más dispositivos Bluetooth pueden pasar por diversos estados que se describen a continuación:

El estado “Standby” es aquel en el que se encuentran los dispositivos Bluetooth por defecto al momento de recibir alimentación eléctrica en sus circuitos. En este estado el dispositivo tiene un bajo consumo de energía y está listo para conectarse a una red como maestro o esclavo.

Luego, se tiene el estado Page/Inquiry que se origina cuando un dispositivo se intenta conectar con otro, enviándole alguno de los dos tipos de mensajes. El primero es del tipo búsqueda (page), cuando la dirección es conocida; el segundo del tipo petición (Inquiry), cuando la dirección no es conocida. Este último necesita una respuesta extra por parte del dispositivo receptor.

Una vez se realizan las operaciones de conexión, los dispositivos pasan a estado “Active” en el que se presenta la transmisión de datos.

Los dispositivos Bluetooth también pueden pasar por algunos estados temporales que son usados principalmente para fines de ahorro de energía. Estos se mencionan a continuación:

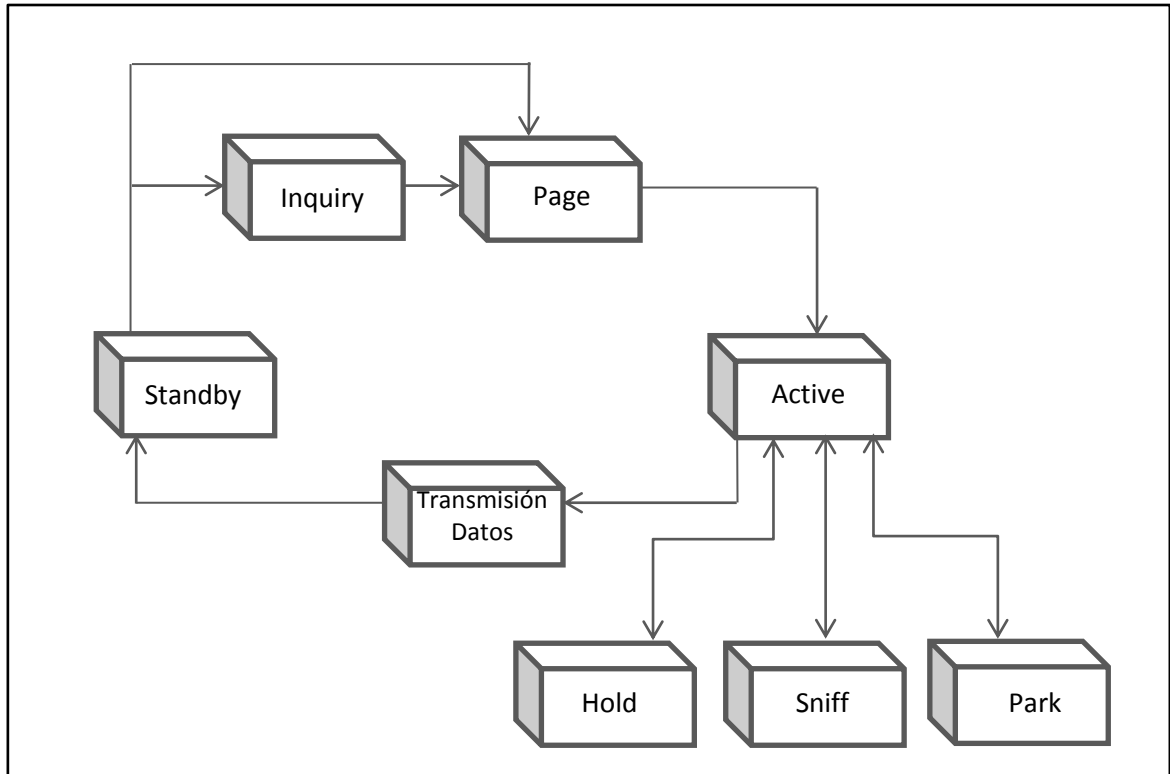
El estado “Hold” se aplica a un dispositivo cuando no hay necesidad de transmitir información durante un periodo relativamente largo, con la intención de conservar energía. Durante este estado el dispositivo puede ser descubierto por otros dispositivos que quieran transmitir información.

Por su parte, el estado “Sniff” se aplica para dispositivos esclavos. Se puede producir cuando el dispositivo pertenece a una piconet pero no hay transmisión en el momento, por lo que el dispositivo entra en modo de bajo consumo en el cual la radio “despierta” en ciertos intervalos para escuchar los mensajes que pueda haber para este.

Finalmente, en el estado “Park” el dispositivo Bluetooth solo escucha al “maestro” para mantener la sincronización, pero este no hace parte del tráfico de datos.

Una mejor forma de ver el paso de un estado a otro para un dispositivo Bluetooth es mediante el diagrama mostrado en la figura 4:

Figura 4. Estados de una conexión Bluetooth



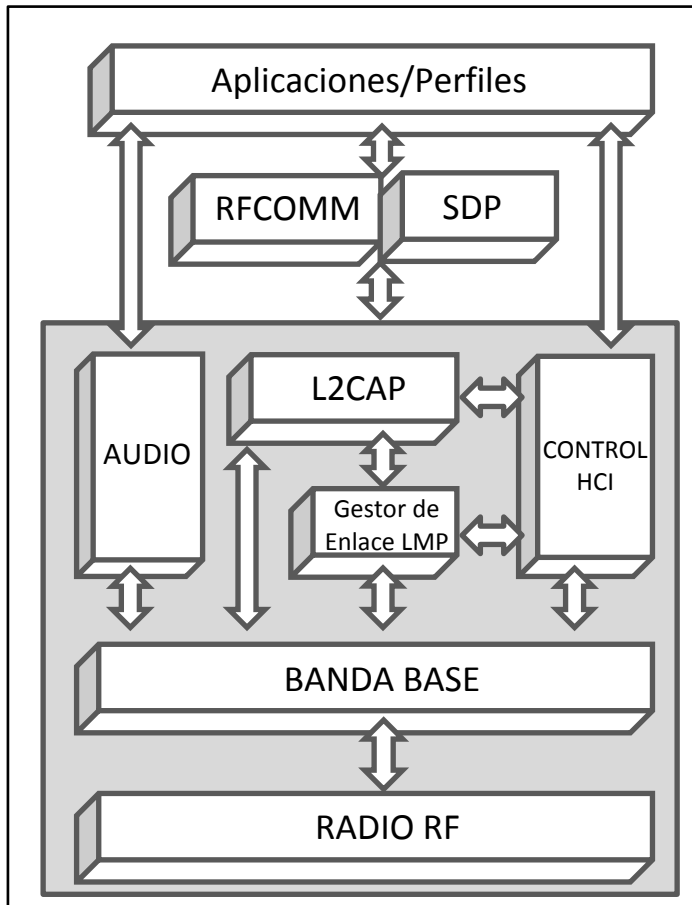
Fuente: GUTIÉRREZ, Daniel. Sistema pasarela Bluetooth para una red de sensores zigbee. En: Capítulo 4 Tecnología Bluetooth. [en línea].

La estructura de la tecnología Bluetooth se basa principalmente en una pila de protocolos, algunos de los cuales son propios de esta tecnología y otros que son adoptados de otras especificaciones con el fin de poder interactuar entre la tecnología Bluetooth y varios protocolos comerciales ya existentes. La pila de protocolos Bluetooth se puede dividir en cuatro capas lógicas:

- Núcleo de Bluetooth: Radio, Banda Base, LMP, L2CAP, SDP
- Sustitución de Cable: RFCOMM
- Protocolos Adoptados: PPP, UDP, TCP, IP, OBEX, WAP, IRMC, WAE
- Control de Telefonía: TCS-binary, AT-Comands

La figura 5 muestra en detalle cómo está constituida la pila de protocolos para la tecnología Bluetooth.

Figura 5. Pila de protocolos Bluetooth



Fuente: PATRÓN, David. Simulación de tramas de WCDMA. En: Capítulo 3 El estándar Bluetooth. Pag 5 [en línea].

- **Radio Bluetooth RF.** Bluetooth opera en una frecuencia de 2.402 MHz a 2.4835MHz la cual es libre para aplicaciones industriales, científicas y médicas. Lo anterior significa que debe coexistir en un ambiente bastante ruidoso por lo que se hace necesaria una interfaz robusta que utiliza un esquema de reconocimiento rápido y saltos de frecuencia.

De esta forma, la interfaz de radio Bluetooth emplea modulación GFSK (Gaussian Frequency Shift Keying) es decir que los datos lógicos, unos y ceros, se representan por una desviación de la frecuencia de la señal. Así, un uno lógico se representa por una desviación positiva y un cero por una desviación negativa.

- **Banda Base (Base Band).** En esta se definen los canales físicos donde cada uno contiene 79 frecuencias de radio diferentes que son accedidas de acuerdo a una secuencia de saltos aleatoria, generalmente 1600 saltos/s, que es

determinada por el dispositivo maestro. Así, se tiene que dos o más dispositivos usando la misma secuencia de saltos conforman una piconet.

Cada canal está dividido en espacios de tiempo denominados comúnmente como (timeslots) por medio de los cuales se realiza la transmisión de los paquetes de información. Cada timeslot corresponde a un salto de frecuencia y tiene una duración de 625us.

En la banda base también se definen dos tipos de enlaces para soportar servicios de transmisión de voz y datos. El primero se denomina enlace síncrono de conexión orientada (SCO) y el segundo enlace asíncrono de conexión menor (ACL).

El Enlace Síncrono de Conexión Orientada (SCO, Synchronous Connection Oriented) es un enlace síncrono, entre un maestro y un esclavo, en el que se soporta voz en tiempo real y servicios multimedia, utilizando un ancho de banda reservado comportándose como un enlace de conmutación de circuitos.

El enlace Asíncrono de Conexión Menor (ACL, Asynchronous Connection-less) es un enlace entre un maestro y uno o varios esclavos activos, donde no se reserva el ancho de banda. Es un enlace de tipo conmutación de paquetes y se usa para transmitir información sin requerimientos temporales.

Así mismo, en la banda base se realiza el establecimiento de conexión, donde se utilizan dos tipos de procedimientos de acceso, dependiendo si se conoce o desconoce la dirección del dispositivo al que se desea conectar. Estos procesos son los de búsqueda (Page) y de petición (Inquiry) explicados anteriormente.

- **Gestor de Enlace (LMP, Link Manager Protocol).** Como su nombre lo indica, este protocolo se encarga de gestionar el enlace entre dispositivos Bluetooth una vez se terminan los procesos de page o inquiry. El dispositivo emisor envía un mensaje de petición de conexión (LMP_host_connection_req) al receptor, el cual puede aceptar (LMP_accepted) o rechazar (LMP_not_accepted) esta petición. En caso de ser aceptada, ambos dispositivos se envían un mensaje de configuración completa (LMP_setup_complete) y se procede a la transmisión de paquetes.

Además, el LMP se encarga de la seguridad, el control de paquetes, la potencia, la calidad del servicio y el control de la piconet.

- **Interface de Control (HCI, Host Controller Interface).** Consiste en una serie de comandos para que el controlador de banda base y el gestor de enlace accedan a los parámetros de configuración.

- **Logical Link Control and Adaption Protocol L2CAP.** Este protocolo se encarga de realizar la adaptación de otros protocolos al protocolo de banda base. Entre sus principales funciones se encuentran la multiplexación de protocolos de alto nivel, la segmentación y re ensamblado de paquetes largos, el descubrimiento de dispositivos y control de calidad de servicio.

- **Service Discovery Protocol SDP.** Permite a las aplicaciones descubrir los servicios disponibles en su entorno así como determinar sus propiedades específicas. Esta búsqueda de servicios debe actualizarse continuamente debido al dinamismo en cuanto al ingreso o salida de dispositivos en el entorno.

La comunicación SDP es del tipo servidor-cliente, donde el cliente busca servicios determinados en el servidor.

- **Protocolo RFCOMM (Radio Frequency Communication).** Es el protocolo mediante el cual, el dispositivo electrónico desarrollado en el presente proyecto entabla una comunicación con el teléfono celular. Este permite emular puertos serie de la norma RS232 sobre el protocolo L2CAP permitiendo conexiones simultáneas entre dispositivos Bluetooth. Este protocolo se usa principalmente, para actualizar sistemas comunicados por cable a sistemas de comunicaciones inalámbricas.

El protocolo RFCOMM es capaz de emular los nueve circuitos que conforman la interfaz RS232 como se muestra en la Tabla 2.

Tabla 2. Circuitos de RS232 emulados Por RFCOMM

Pin	Nombre
1	Data carrier Detect (CD)
2	Received Data (RD)
3	Transmitted Data (TD)
4	Data terminal Ready (DTR)
5	Signal Ground (SG)
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Ring Indicator (RI)

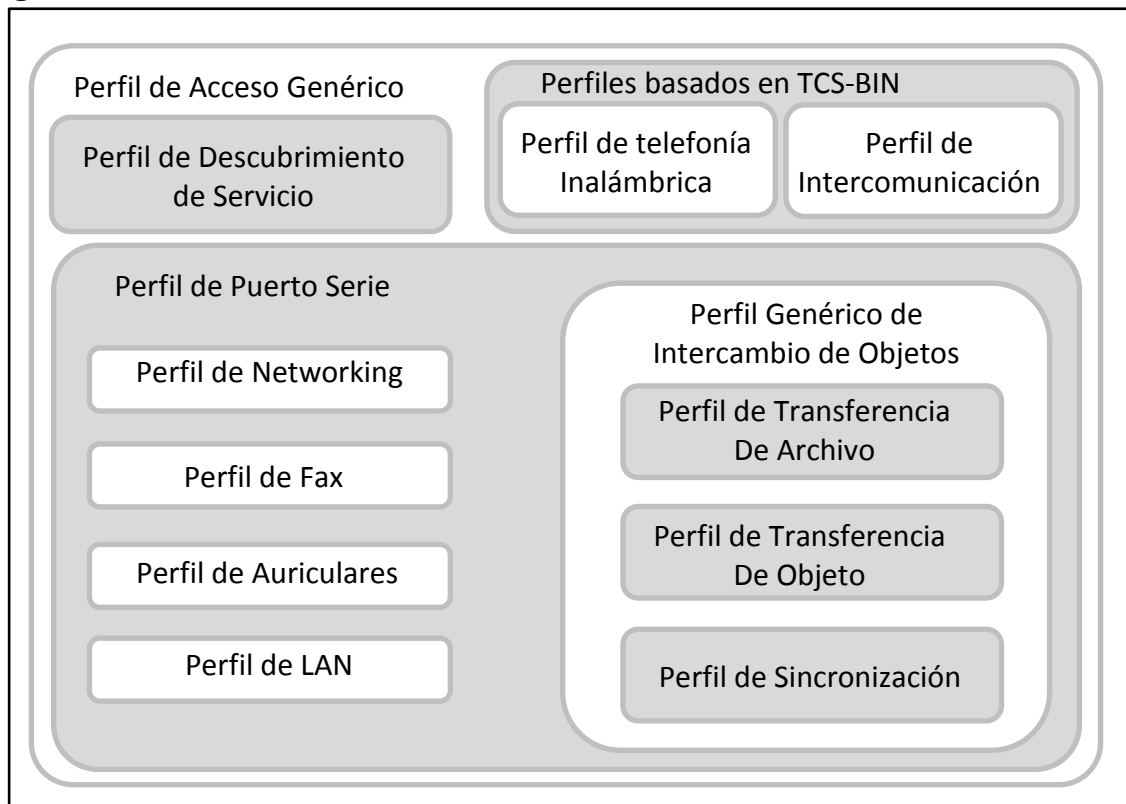
Fuente: PALACIOS, Enrique, DOMÍNGUEZ, Fernando, LÓPEZ, Lucas. Microcontrolador PIC16F84 Desarrollo de proyectos. Madrid: RA-MA Editorial, 2006. Pag 306.

Además, dos dispositivos que usen RFCOMM para su comunicación, pueden abrir simultáneamente hasta 60 puertos series emulados. Esto es gracias al Identificador de conexión de enlace de datos DLCI (Data Link Connection Identifier) que identifica las conexiones entre aplicaciones cliente servidor. El DLCI es representado por 6 bits pero solamente es utilizable en el rango de 2-61 es decir 60 canales.

Así, para hacer uso de la pila de protocolos Bluetooth, se dispone de los perfiles (figura 6) que son los procedimientos que se usan para situaciones o tareas específicas en un dispositivo. Cada dispositivo debe tener por lo menos un perfil, es decir una tarea o aplicación para la cual ha sido diseñado. Así, para que 2 dispositivos puedan comunicarse deben compartir un mismo perfil.

Así mismo, los perfiles ayudan a optimizar el desarrollo de dispositivos Bluetooth, puesto que no es necesario implementar toda la pila de protocolos Bluetooth en un dispositivo, sino que por el contrario, solo se implementa la parte que se va a utilizar. Esto permite aprovechar mejor los recursos con los que cuenta el dispositivo ya que no se satura la memoria con perfiles que no se usaran en un dispositivo específico.

Figura 6. Perfiles comunes Bluetooth



Fuente: GUTIÉRREZ, Daniel. Sistema pasarela Bluetooth para una red de sensores zigbee. En: Capítulo 4 Tecnología Bluetooth. [en línea].

A continuación se describen los perfiles utilizados para el desarrollo del proyecto.

- **Perfil de Acceso Genérico (GAP).** En este se encuentran los procedimientos generales para el descubrimiento y el establecimiento de conexión entre dos dispositivos Bluetooth.
- **Perfil de Aplicación de Descubrimiento de Servicios SDAP (Service Discovery Application Profile).** Se encarga de los procedimientos necesarios que usa un dispositivo para descubrir los servicios y características que se encuentran en otros dispositivos.
- **Perfil de Puerto Serie SPP (Serial Port Profile).** Define los procesos específicos para emular puertos serie entre dispositivos Bluetooth. Dichos procesos se aplican en el protocolo RFCOMM. En otras palabras, reemplaza una conexión serie cableada por una conexión inalámbrica en las aplicaciones de comunicación y señales de control basadas en conexiones de serie RS-232.

Por otra parte, desde sus inicios, Bluetooth ha pasado por diversas versiones que cada vez mejoran sus características y ofrecen más servicios en los dispositivos y es importante realizar una breve descripción de cada una de estas, puesto que el dispositivo electrónico debe ser compatible con teléfonos celulares que cuenten con una versión diferente de Bluetooth.

- **Bluetooth v.1.1.** La tecnología se ratifica como estándar IEEE 802.15.1-2002, donde se corrigen algunos errores del estándar original y se añade soporte para canales no cifrados y un indicador de señal recibida (RSSI) que mide la potencia de las señales recibidas en redes inalámbricas.
- **Bluetooth v.1.2.** Se implementa una solución para que Bluetooth pueda coexistir sin interferencias con Wi-Fi en el espectro de los 2.4GHz. Se usa la técnica Adaptive Frequency Hopping (AFH) que ofrece transmisiones más eficientes y un cifrado más seguro. Asimismo se mejora la calidad de la transmisión de voz.
- **Bluetooth v.2.0 + EDR.** Se centra en mejorar la velocidad de transmisión de datos incorporando la técnica Enhanced Data Rate (EDR) que permite alcanzar velocidades de hasta 3Mbps.
- **Bluetooth v.2.1 + EDR.** El principal avance de esta versión es la técnica Secure Simple Pairing (SSP) que simplifica los pasos para realizar una conexión entre los dispositivos y también se reduce el consumo de energía en modo bajo consumo.

- **Bluetooth v.3.0 + HS.** Se aumenta la velocidad de transferencia de datos a valores teóricos de hasta 24Mbps, lo anterior debido a que se utiliza AMP (Alternate MAC/PHY) una técnica asociada a Wi-Fi para el transporte de datos y la radio Bluetooth solamente se utiliza para detección, conexión inicial y configuración del perfil.

- **Bluetooth v.4.0.** La última versión dada por SIG en el 2010 integra todas las versiones de Bluetooth incluyendo transmisiones de alta velocidad y una nueva generación denominada Bluetooth de baja energía BLE (Bluetooth Low Energy) que incluye dicha tecnología en dispositivos mucho más pequeños aumentando sus aplicaciones.

1.3 MARCO CONCEPTUAL

En esta sección se describen los conceptos más importantes en el desarrollo del proyecto tales como los tonos DTMF y los comandos AT.

1.3.1 Tonos DTMF. Los tonos duales de multi frecuencia DTMF (Dual Tone Multi Frequency), fueron diseñados por laboratorios Bell en los años 60, como una alternativa para el marcado por pulsos de la telefonía fija de esta época. Generando un estándar que a través de los años se traduce en varias especificaciones dadas por diferentes organismos de telefonía y que difieren muy poco del concepto original.

El estándar DTMF comprende la transmisión de 16 tonos que están conformados por los números (1-9) además de los símbolos (* #) y las letras (A-D). Así, un tono DTMF está compuesto por la suma algebraica en tiempo real de dos ondas sinusoidales de diferentes frecuencias una alta y otra baja, escogidas muy cuidadosamente para evitar armónicos y otros problemas en la transmisión. Dichas frecuencias se muestran en la tabla 3.

Tabla 3. Pares de frecuencias empleadas para la generación DTMF

Frecuencias Altas Frecuencias Bajas	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C

Tabla 3. (Continuación) Pares de frecuencias empleadas para la generación DTMF

Frecuencias Altas Frecuencias Bajas	1209 Hz	1336 Hz	1477 Hz	1633 Hz
941 Hz	*	0	#	D

Fuente: Reocities. Tonos duales de multi frecuencia (DTMF) [en línea].

La expresión matemática que representa los tonos DTMF se obtiene del siguiente análisis:

- | | | |
|---|---|-----|
| 1. Cada señal de un tono DTMF tiene la forma: | $\cos(A) + \cos(B)$ | (1) |
| 2. Partiendo de la identidad Trigonométrica | $\cos(u + v) + \cos(u - v) = 2 \cos(u) \cos(v)$ | (2) |
| 3. Se tiene que | $\cos(u + v) + \cos(u - v) = \cos(A) + \cos(B)$ | (3) |
| 4. Donde | $A = u + v$ | (4) |
| | $B = u - v$ | (5) |
| 5. Sumando (4) y (5) y despejando | $u = \frac{1}{2}(A + B)$ | (6) |
| 6. Restando (4) y (5) y despejando | $v = \frac{1}{2}(A - B)$ | (7) |
| 7. De acuerdo a la ecuación (2) reemplazamos en términos de A y B | $2 \cos(u) \cos(v) = 2 \cos\left(\frac{1}{2}(A + B)\right) \cos\left(\frac{1}{2}(A - B)\right)$ | (8) |
| 8. Sabemos que A y B están en radianes así que reemplazamos | $2 \cos(u) \cos(v) = 2 \cos(\pi(A + B)) \cos(\pi(A - B))$ | (9) |

Donde A y B son las frecuencias bajas y altas en Hertz mencionadas en la tabla 2.

Así, las señales de los tonos DTMF son generadas por un circuito codificador que genera una señal análoga en el rango de la voz humana a partir de un dato digital. Este codificador se implementó principalmente para telefonía fija, pero hoy en día podemos encontrarlo inmerso en los dispositivos de comunicaciones móviles.

Por otra parte, el proceso de decodificación consiste en una serie de filtros para separar la señal en grupos de alta y baja frecuencia para luego medir el cruce por cero promediando los periodos y así producir la decodificación de un dígito obteniendo una señal digital partiendo de una analógica.

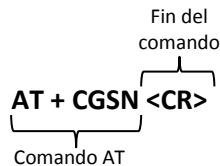
De esta forma, para que un tono DTMF sea correctamente decodificado, se debe tener en cuenta una tolerancia a la frecuencia menor al 1.5%; una duración de la señal mínimo de 40 ms y una relación señal-ruido (SNR) de 15dB y en el peor de los casos de 26dB.

De igual manera, se puede interrumpir la señal por 10 ms máximo para que no sea detectada como otro tono distinto y cada tono debe ser separado de otro con una pausa de mínimo 40ms. Por último teniendo en cuenta que los tonos DTMF se transmiten por el mismo medio que la voz humana, esta última no debe ser detectada como un tono válido.

1.3.2 Comandos AT. Los comandos AT (Attention) son un grupo de instrucciones desarrolladas por Dennis Hayes en 1977 principalmente como interfaz de comunicación para configuración e instrucciones de módems. Luego estos comandos fueron adoptados y extendidos por las compañías Microcomm y US Robotics hasta universalizarlos.

Con el desarrollo de la telefonía móvil, el estándar GSM adopta los comandos AT para la comunicación con sus terminales, por lo que la mayoría de los terminales GSM poseen un juego de comandos AT para configuración, establecimiento de llamadas, SMS y otras funciones tales como el envío de tonos DTMF. En cuanto a su sintaxis consta de una petición por parte del usuario y una respuesta, de aceptación o de error, por parte del terminal como se puede observar en el siguiente ejemplo:

Se introduce el comando AT para solicitarle al terminal su número de serie y se finaliza con el carácter especial retorno de carro <CR>:



Una vez recibido el comando, el terminal da una respuesta que está en medio de una secuencia de inicio y una secuencia de fin. Estas secuencias están compuestas por los caracteres especiales retorno de carro <CR> y cambio de línea <LF>:

Secuencia de Inicio Respuesta Secuencia de Fin

<CR> <LF> 357692035272262 <CR> <LF>
<CR> <LF> OK <CR> <LF>

En caso de ingresar un comando AT no valido, el terminal responderá con un mensaje de error:

Respuesta

<CR> <LF> ERROR <CR> <LF>

Hoy en día, estos comandos están incluidos en gran variedad de dispositivos de radiofrecuencia, permitiendo al usuario realizar tareas de configuración y control.

2. METODOLOGÍA

2.1 PROCESOS PARA EL CONTROL DEL TELÉFONO CELULAR

Para el desarrollo del proyecto se utilizó un Teléfono celular de gama media desarrollado por la compañía ZTE Corporation en 2010 cuya referencia es ZTE N298. Este teléfono es compatible con redes GSM 900/ DSC 1800 /GSM 850/ PCS 1900 y 2.5G GPRS y EDGE. Entre sus características más importantes se encuentra el modulo Bluetooth v2.0 con perfiles de puerto serie (SPP), carga de objetos, transferencia de archivos, auriculares entre otros. Además de una pantalla táctil de 2.8", cámara de 2MP, java v2.1, memoria externa MicroSD con capacidad máxima de 4GB y conexión a internet mediante WAP 2.0.

A continuación en la figura 7 se muestra el aspecto físico del teléfono móvil ZTE N281.

Figura 7. Aspecto Físico ZTE N281



Fuente: Este trabajo.

Además, se utilizó el dispositivo Bluetooth BlueSMiRF SILVER diseñado por Sparkfun Electronics que a su vez cuenta con el modulo Bluetooth RN-42 que es diseñado por Roving Networks. El modulo es compatible con los estándares Bluetooth v2.1, v2.0+EDR, v1.2, v1.1 y está construido en base a los protocolos GAP, SDP, RFCOMM y L2DCAP con perfil SPP. Lo anterior permite al dispositivo emular una conexión de puerto serie inalámbrica por medio de tecnología Bluetooth.

El BlueSMiRF Silver cuenta con las siguientes Características⁴:

- Voltaje de alimentación 3.3V – 6V
- Rango Máximo 18 mts
- Bajo consumo de potencia 25mA
- Opera bajo ambientes WiFi, 802.11g, Zigbee
- Comunicación serial 2400bps – 115200bps
- Configuración mediante comandos AT

En la figura 8 se muestra el aspecto físico del módulo BlueSMiRF Silver mientras que en la figura 9 se muestra el circuito esquemático del mismo.

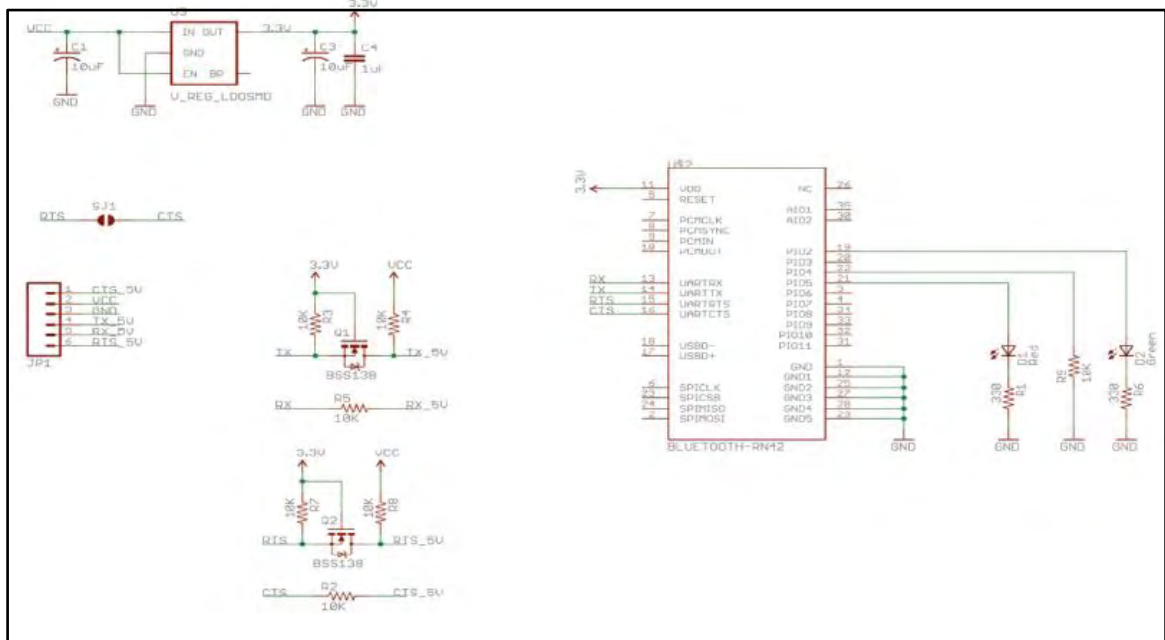
Figura 8. Aspecto Físico BlueSMiRF



Fuente: Este trabajo.

⁴Roving Networks. RN-42/RN-42-N Datasheet.
<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/Bluetooth/Bluetooth-RN-42-DS.pdf>
(Noviembre 16 de 2012)

Figura 9. Circuito Esquemático BlueSMiRF



Fuente: Sparkfun Electronics. Bluetooth Modem - BlueSMiRF Silver [en línea].

Así mismo, los pines principales del BlueSMiRF se muestran en la tabla 4:

Tabla 4. Principales pines de conexión del BlueSMiRF

Nombre de Pin	Descripción
Vcc	Alimentación de +3.3v a +6v
GND	Conexión a tierra
Tx-O	Transmisión de datos UART
Rx-I	Recepción de datos UART

Fuente: Este trabajo.

Teniendo en cuenta las características del teléfono móvil ZTE N281 y del módulo BlueSMiRF, se determinó que son óptimos para el desarrollo del proyecto y en esta sección se describe los procesos realizados para tomar el control del teléfono a través de tecnología Bluetooth usando el módulo BlueSMiRF.

2.1.1 Configuración del módulo BlueSMiRF. Antes de realizar la conexión entre el teléfono y el módulo, este último se configuró para establecer los parámetros con los que se trabajó en el proyecto, en este caso se decidió trabajar a una velocidad de transmisión de datos de 9600bps en el puerto serie por lo que se debía configurar el modulo a este parámetro.

Para la configuración, se hizo necesario el uso de tarjeta FT232, diseñada por Sigma Electrónica y que contiene el circuito Integrado FT232RL un convertor de

interfaz serial USB-UART, el cual tiene la capacidad de crear un puerto serie en un PC por medio de conexión USB. Lo anterior significa que se puede contar con un puerto COM en PCs que no poseen puertos seriales con la norma RS232. Su aspecto físico se muestra en la figura 10, mientras que sus principales pines de conexión se muestran en la tabla 5. Cabe aclarar que esta tarjeta se alimenta directamente del puerto USB por lo que usa niveles de voltaje TTL en sus pines.

Figura 10. Aspecto Físico de la Tarjeta FT232



Fuente: Sigma Electronica Ltda. Tarjeta FT232. [en línea].

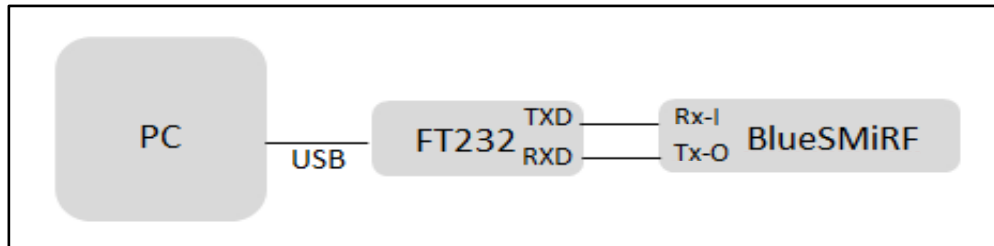
Tabla 5. Principales pines de conexión de la tarjeta FT232RL

Nombre de Pin	Descripción
GND	Conexión a tierra
TXD	Transmisión de datos UART
RXD	Recepción de datos UART

Fuente: Este Trabajo.

Así, para la configuración del módulo BlueSMiRF se alimentó el módulo con 5v y sus Pines Tx y Rx se conectaron a la tarjeta FT232 que a su vez se conectó mediante cable USB a un computador, como se muestra en la figura 11.

Figura 11. Conexión para configuración del módulo BlueSMiRF

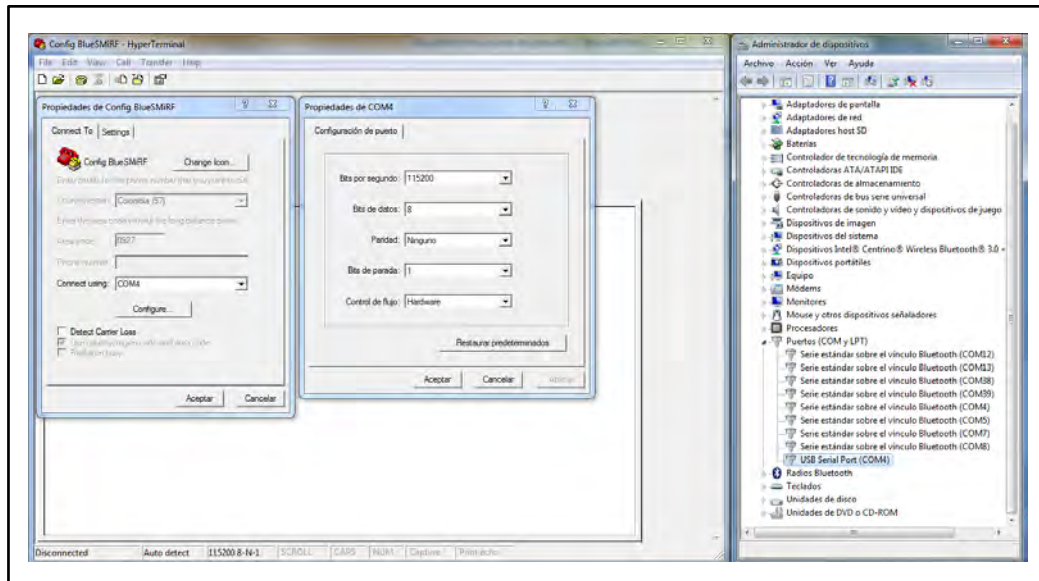


Fuente: Este trabajo.

El módulo BlueSMiRF funciona básicamente en dos modos, el modo datos y el modo comandos. En el modo datos, el módulo está listo para comportarse como un puerto serie, es decir emulando una conexión física entre los dispositivos conectados. Por su parte, el modo comandos se utiliza para la configuración del mismo.

Así, una vez realizada la conexión, se inició un programa para comunicación por puerto serie como es Hyperterminal y en primer lugar, se configuró el puerto emulado por la tarjeta FT232 (COM4) a 115200bps que es la velocidad por defecto para el BlueSMiRF como se muestra en la figura 12.

Figura 12. Configuración del puerto emulado por la tarjeta FT232

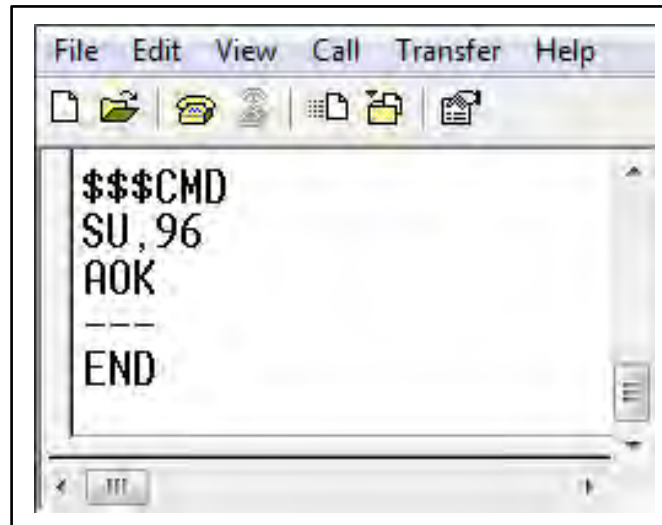


Fuente: Este trabajo.

Luego de aplicar la configuración del puerto, se ingresó al modo de comandos para el BlueSMiRF digitando “\$\$\$” e inmediatamente el modulo respondió con “CMD<CR>” lo que indica que está listo para recibir comandos de configuración.

Para cambiar la velocidad de transmisión a 9600bps (figura 13), se digitó el comando “SU,96<CR>” y el modulo respondió con “AOK<CR>” lo que significa que la configuración había cambiado. Por último se digitó “---<CR>” para salir de modo comandos al modo datos. Cabe aclarar que los cambios se aplicaron al reiniciar el dispositivo.

Figura 13. Secuencia de comandos configuración BlueSMiRF



Fuente: Este trabajo.

2.1.2 Conexión entre el teléfono y el módulo BlueSMiRF. Con el módulo conectado a Hyperterminal, se enviaron los comandos para establecer una conexión Bluetooth por primera vez con el teléfono celular. Para esto fue necesario que el módulo BlueSMiRF realice procesos de Indagación (Inquiry) para luego proceder a conectarse a la dirección del teléfono mediante procesos de búsqueda (Page). Lo anterior se realizó con los comandos AT mostrados en la figura 14. Cabe aclarar que el teléfono tenía su dispositivo Bluetooth activado.

Figura 14. Procesos de indagación y conexión entre el modulo y el teléfono

```

$$$CMD // entra en modo de configuracion
I //comando AT para relizar operaciones de tipo Inquiry
Inquiry, COD=0 //el modulo comienza con la busqueda de dispositivos Bluetooth
Found 1 //muestra el numero de dispositivos encontrados
D0154A462B22, telefono, 5A0204 // muestra la direccion y el nombre del dispositivo encontrado
Inquiry Done //indica que ha finalizado la busqueda
C, D0154A462B22 //Comando AT para establecer conexion con la direccion digitada
TRYING // el modulo intenta conectarse al dispositivo con esa direccion, si la conexion es exitosa
// pasa automaticamente a modo datos.

```

Fuente: Este trabajo.

Una vez conectados los dispositivos por primera vez, el módulo BlueSMiRF guarda en su memoria la dirección del teléfono. Así, para una próxima conexión solo se necesita entrar en modo comandos y luego digitar el comando "C<CR>" para que el dispositivo intente conectarse al último teléfono conectado.

2.2 TRANSMISIÓN DE TONOS DTMF DURANTE UNA LLAMADA DE VOZ

Para los procesos de envío y recepción de tonos DTMF se usaron dos métodos diferentes. El envío se realizó por medio de comandos AT mientras que la recepción se realizó por medio de señales de audio tomadas del auricular del teléfono celular.

Como el número de tonos DTMF está limitado a 16, se hizo necesario implementar un método que permita transmitir caracteres de alfabeto, números y símbolos especiales con los 16 tonos disponibles. Este método consiste en enviar uno o dos tonos DTMF para representar un carácter, es decir que se pueden transmitir caracteres representados por un solo tono DTMF y otros caracteres representados por una secuencia de 2 tonos DTMF.

Así es como la Tabla 6 representa el método utilizado para el envío y recepción de tonos DTMF.

Tabla 6. Representación de caracteres mediante tonos DTMF

CARÁCTER	CARÁCTER CON MAYUS	CARÁCTER CON ALT	TONO DTMF	No. DE TONOS
a	A	*	1	1
b	B	!	2	1
c	C	9	3	1
d	D	5	4	1
e	E	2	5	1
f	F	6	6	1
s	S	4	7	1
espacio		&	8	1
i	I	-	9	1
del		=	0	1
u	U	_	*	1
l	L	"	#	1
m	M	.	A	1
n	N	,	B	1
o	O	+	C	1

Tabla 6. (Continuación) Representación de caracteres mediante tonos DTMF

CARÁCTER	CARÁCTER CON MAYUS	CARÁCTER CON ALT	TONO DTMF	No. DE TONOS
p	P	@	1	2
q	Q	#	2	2
r	R	3	3	2
g	G	\$	4	2
t	T	(5	2
k	K	%	6	2
v	V	?	7	2
w	W	1	8	2
x	X	8	9	2
y	Y)	0	2
z	Z	7	*	2
MAYUS		0	#	2
alt			A	2
j	J	;	B	2
h	H	:	C	2

Fuente: Este trabajo.

En la columna “CARÁCTER” se encuentran aquellos que se representan por defecto, las columnas “MAYUS” y “ALT” son banderas que se activan o desactivan, permitiendo representar caracteres distintos con los mismos tonos DTMF para un total de 85 caracteres.

Así mismo, en la columna “No DE TONOS”, se muestra la cantidad de tonos necesarios para representar determinados caracteres. Cuando este valor es 2, se transmite en secuencia el tono DTMF “D” y luego el tono mostrado en la columna “TONO DTMF”.

2.2.1 Envío de tonos DTMF. Este procedimiento se realizó mediante comandos AT específicos, que permiten al teléfono enviar a través de la red celular dichos tonos para que sean escuchados por el teléfono remoto. Estos son:

AT+CMUT // Comando para inhabilitar el micrófono del teléfono para evitar ruidos //del ambiente.

AT+VTS // Comando para el envío de un tono DTMF a través de la red celular.

Así, por ejemplo, si se desea enviar el tono DTMF “A”, se enviarían las órdenes al teléfono celular por medio de comandos AT, de la siguiente manera:

AT+CMUT=1<CR> // se silencia el micrófono del teléfono.

AT+VTS=A<CR> // Envío del tono DTMF “A” a través de la red celular.

Para que estos comandos sean válidos, se hace necesario que se encuentre establecida una llamada de voz. De lo contrario el teléfono no aceptará los comandos y responderá con error.

2.2.2 Recepción de tonos DTMF. Para realizar el proceso de recepción de tonos DTMF, fue necesario hacer uso de la señal de audio proveniente del auricular del teléfono celular y acoplarla al decodificador de tonos MT8870DE para obtener un dato digital que permitió trabajar y visualizar en pantalla el carácter recibido.

En cuanto al decodificador de tonos DTMF MT8870DE, se trata de un circuito integrado capaz de recibir y decodificar tonos DTMF, que cuenta con una etapa de amplificación para adecuar la señal de entrada y acoplar impedancias, luego la señal pasa por 8 filtros pasa bandas de sexto orden que usan técnicas de capacitores conmutados y que son diseñados para separar las frecuencias duales del sistema DTMF.

Además, cuenta con una etapa para convertir la señal analógica en digital, teniendo la capacidad de interpretar los 16 tonos DTMF en un código binario de 4 bits para que se integre de forma sencilla a sistemas basados en microcontroladores.

El aspecto físico del Decodificador DTMF MT8870DE puede observarse en la figura 15.

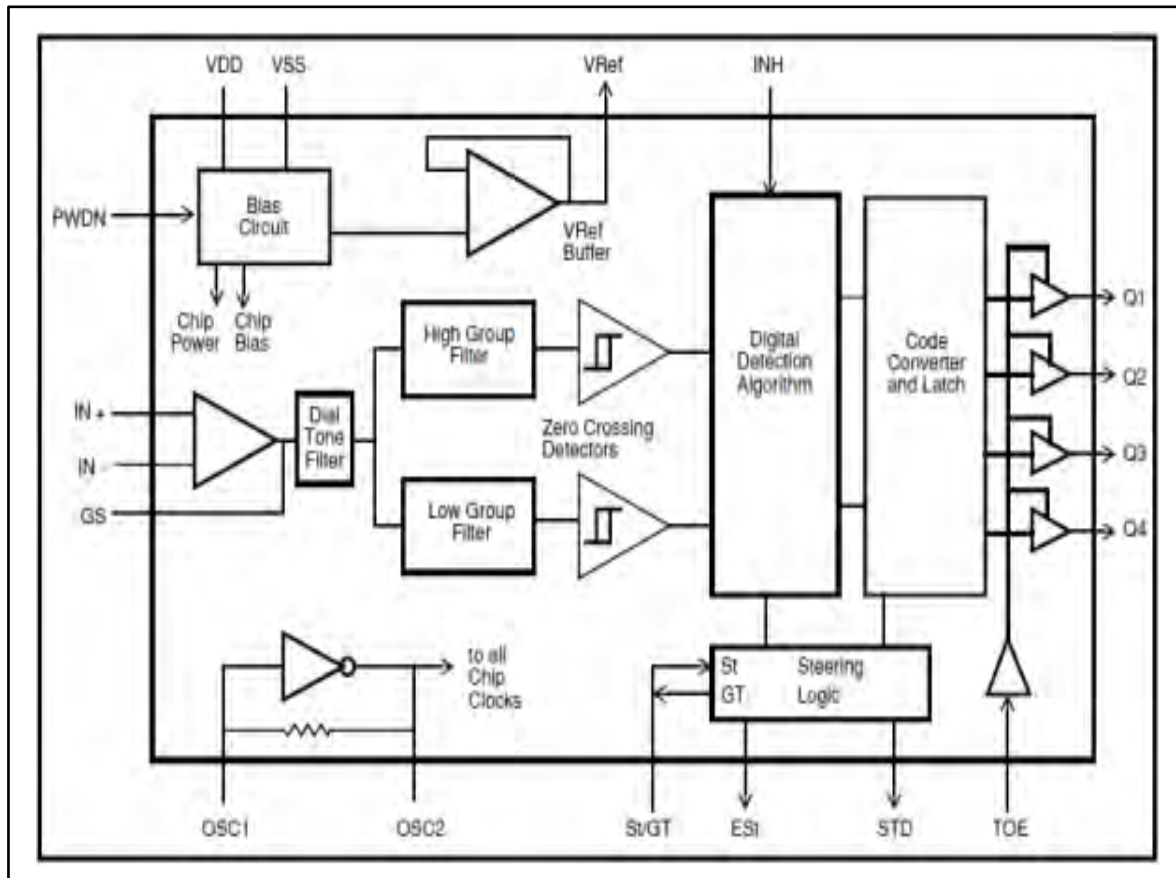
Figura 15. Aspecto físico del circuito integrado MT8870DE



Fuente: Este trabajo.

Para tener una apreciación en detalle que permita observar cada una de las etapas o bloques que conforman el circuito integrado MT8870DE, se dispone de la figura 16.

Figura 16. Diagrama de bloques del circuito integrado MT8870DE fabricado por Zarlink Semiconductor



Fuente: Zarlink Semiconductor. MT8870D/MT8870D-1 Integrated DTMF Receiver [en línea].

Así mismo, por medio de la tabla 7 se pueden observar las funciones que cumplen los pines más relevantes del circuito integrado MT8870DE, así como su configuración como entradas o salidas.

Tabla 7. Pines principales del circuito integrado MT8870DE

Terminales	Entrada/salida	Función
IN +	Entrada	Entrada no inversora a la etapa de amplificación.
IN -	Entrada	Entrada inversora a la etapa de amplificación.

Tabla 7. (Continuación) Pines principales del circuito integrado MT8870DE

Terminales	Entrada/salida	Función
GS	Salida	Selector de ganancia para la etapa de amplificación.
PWDN	Entrada	Cuando está en alto pone el circuito Integrado en modo de bajo consumo de energía.
OSC1, OSC2	Entrada/Salida	Oscilador de 3.57Mz usado para la base de tiempo del circuito.
Vdd y Vss	Entrada	Alimentación.
TOE	Entrada	Habilita las salidas Q1-Q4.
Q1-Q4	Salida	Código binario de 4 bits correspondiente al DTMF recibido.
StD	Salida	Se activa cuando se ha recibido y decodificado un DTMF valido.

Fuente: Zarlink Semiconductor. MT8870D/MT8870D-1 Integrated DTMF Receiver [en línea].

En cuanto a las salidas Q1-Q4, se ponen en estado alto o bajo dependiendo de los tonos DTMF identificados por el MT8870DE. Dicha correspondencia se muestra en la Tabla 8.

Tabla 8. Datos digitales en Q1-Q4 de acuerdo al tono DTMF presente

Dígito	Q4	Q3	Q2	Q1
NA	Z	Z	Z	Z
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
0	1	0	1	0
*	1	0	1	1
#	1	1	0	0

Tabla 8. (Continuación) Datos digitales en Q1-Q4 de acuerdo al tono DTMF presente

Dígito	Q4	Q3	Q2	Q1
A	1	1	0	1
B	1	1	1	0
C	1	1	1	1
D	0	0	0	0

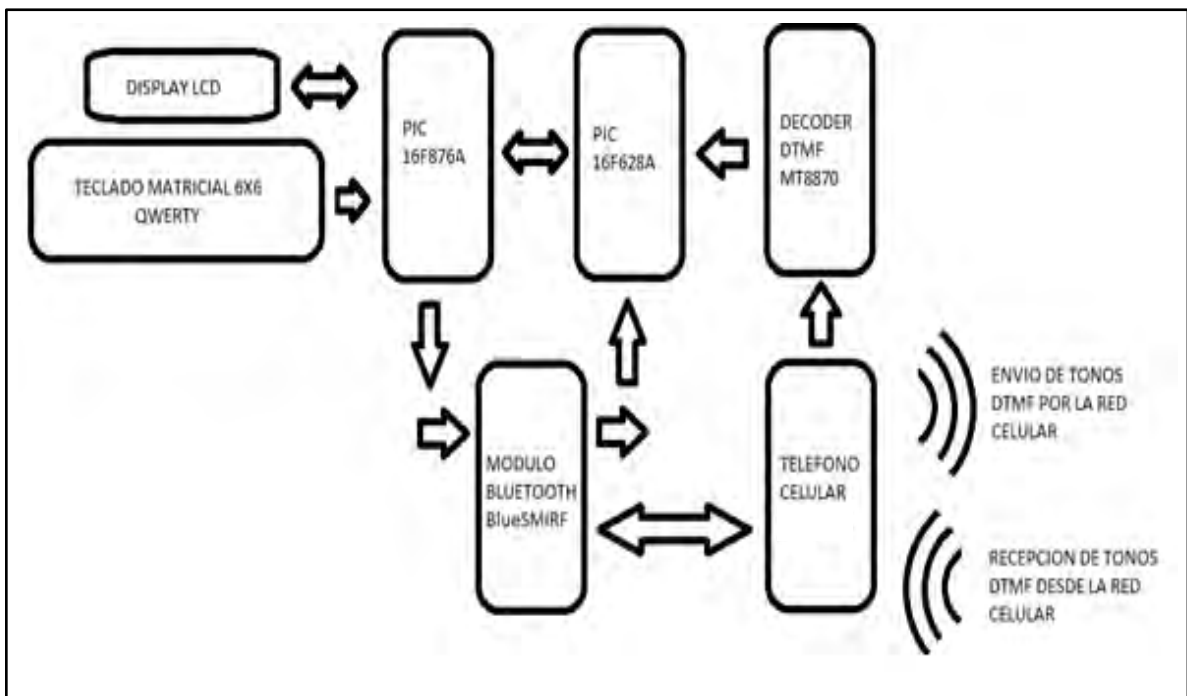
Fuente: Zarlink Semiconductor. MT8870D/MT8870D-1 Integrated DTMF Receiver [en línea].

Así por ejemplo, cuando el circuito MT8870DE recibe un tono DTMF cuyo par de frecuencias corresponden al dígito “9”, en sus salidas Q4, Q3, Q2, Q1 estarán presentes los datos 1, 0, 0, 1 respectivamente.

2.3 DISEÑO DE CIRCUITOS PARA EL DISPOSITIVO ELECTRÓNICO

El dispositivo electrónico se implementó mediante 3 etapas principales como son el control del teléfono ZTE N281, el envío y la recepción de tonos DTMF y la interfaz de usuario. Estas etapas se encuentran implícitas en el diagrama de bloques mostrado en la figura 17.

Figura 17. Diagrama de bloques principal del dispositivo electrónico



Fuente: Este Trabajo.

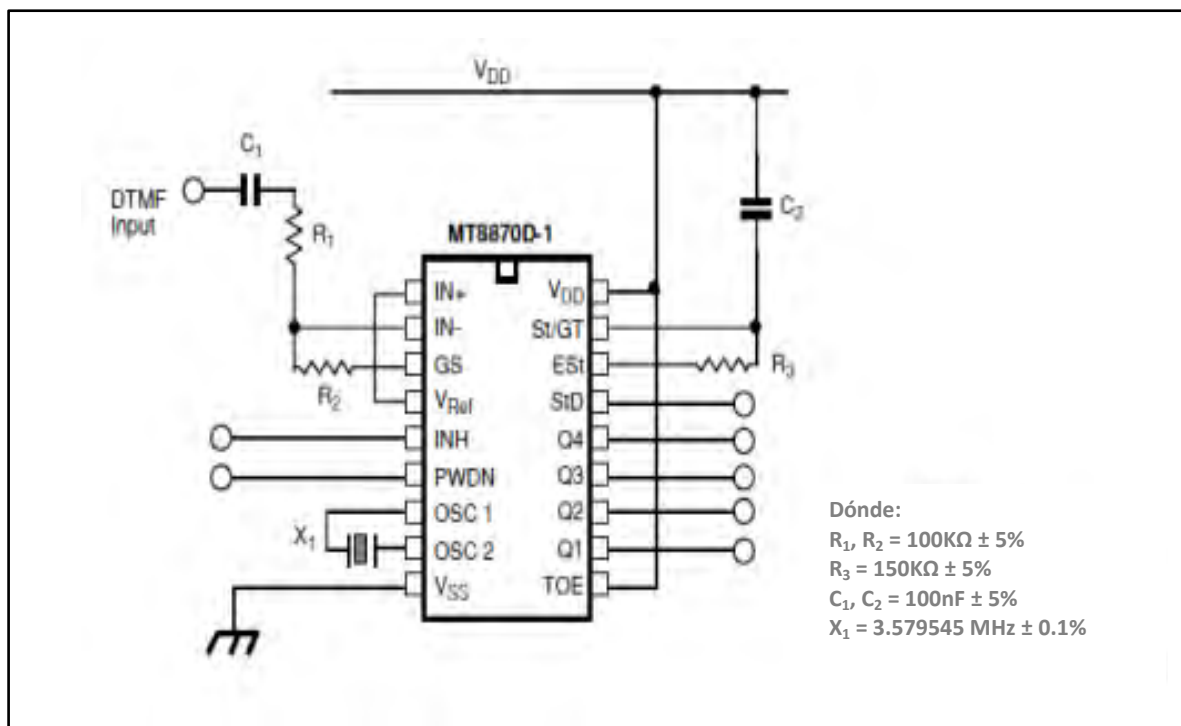
El decodificador DTMF MT8870DE, es el encargado de recibir las señales de audio provenientes del teléfono celular y las decodifica obteniendo un dato digital que es óptimo para trabajar con los microcontroladores. La interfaz de usuario, está conformada por un teclado matricial que envía la información acerca de las teclas presionadas por el usuario, mientras que la pantalla LCD permite visualizar la información transmitida.

Es importante resaltar que los elementos que gobiernan el funcionamiento del dispositivo electrónico son dos microcontroladores PIC que albergan en sus memorias el firmware que transmite las señales de control hacia los periféricos y obtiene los datos recibidos desde los periféricos para realizar acciones determinadas.

Por su parte, la pareja del módulo BlueSMiRF y el teléfono celular conforman el puente que permite acceder a la red celular para el envío y recepción de la información.

2.3.1 Circuito para decodificación de tonos DTMF. Para el diseño del circuito que gobierna el decodificador MT8870DE, se tomó como referencia el circuito recomendado por el fabricante el cual se muestra en la figura 18.

Figura 18. Circuito decodificador de tonos DTMF



Fuente: Zarlink Semiconductor. MT8870D/MT8870D-1 Integrated DTMF Receiver [en línea].

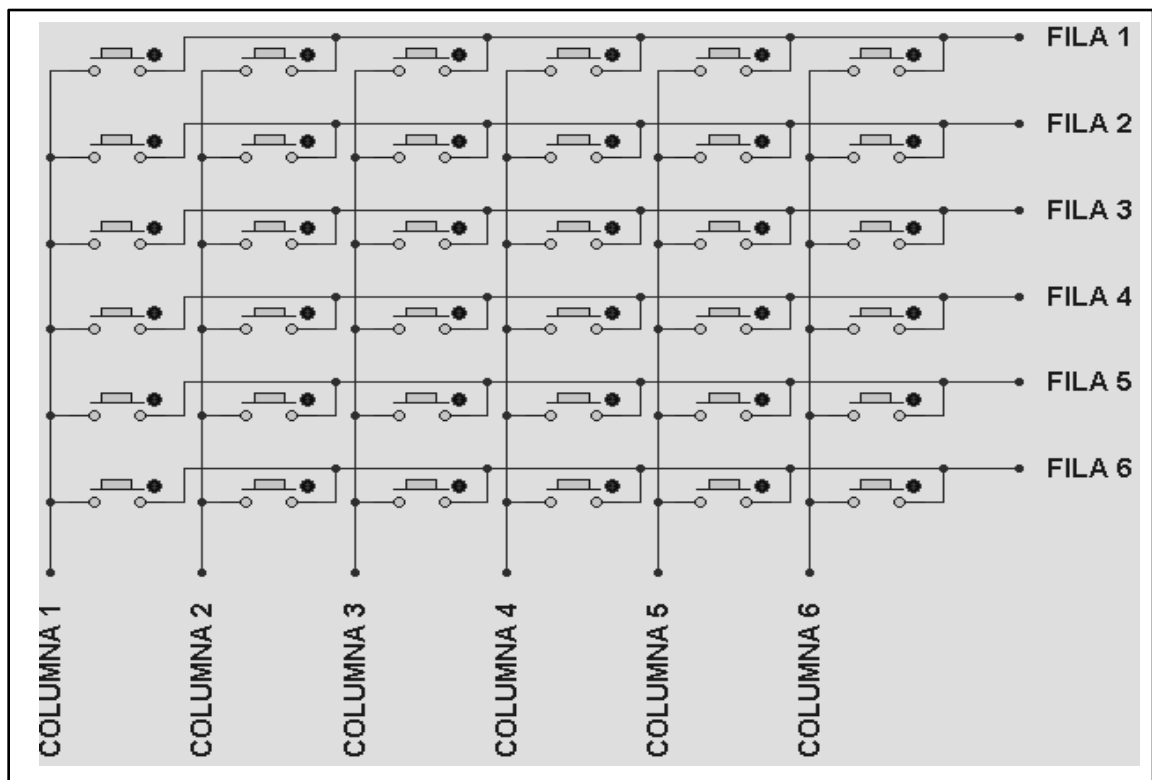
La entrada DTMF Input se conecta al auricular del teléfono celular, las salidas Q1-Q4 entregan el dato digital que representa el tono DTMF recibido y la salida StD entrega un 1 lógico cuando se finaliza la decodificación de un tono DTMF válido.

Así mismo, las resistencias R1 y R2 establecen la ganancia para la entrada de audio, mientras que la resistencia R3 y el condensador C2 forman un circuito RC que establece el tiempo de duración mínimo de cada tono DTMF para que éste último pueda ser decodificado por el integrado.

2.3.2 Interfaz de usuario. Para la interfaz de usuario se hizo uso de dos componentes esenciales que permiten la interacción del usuario con el dispositivo.

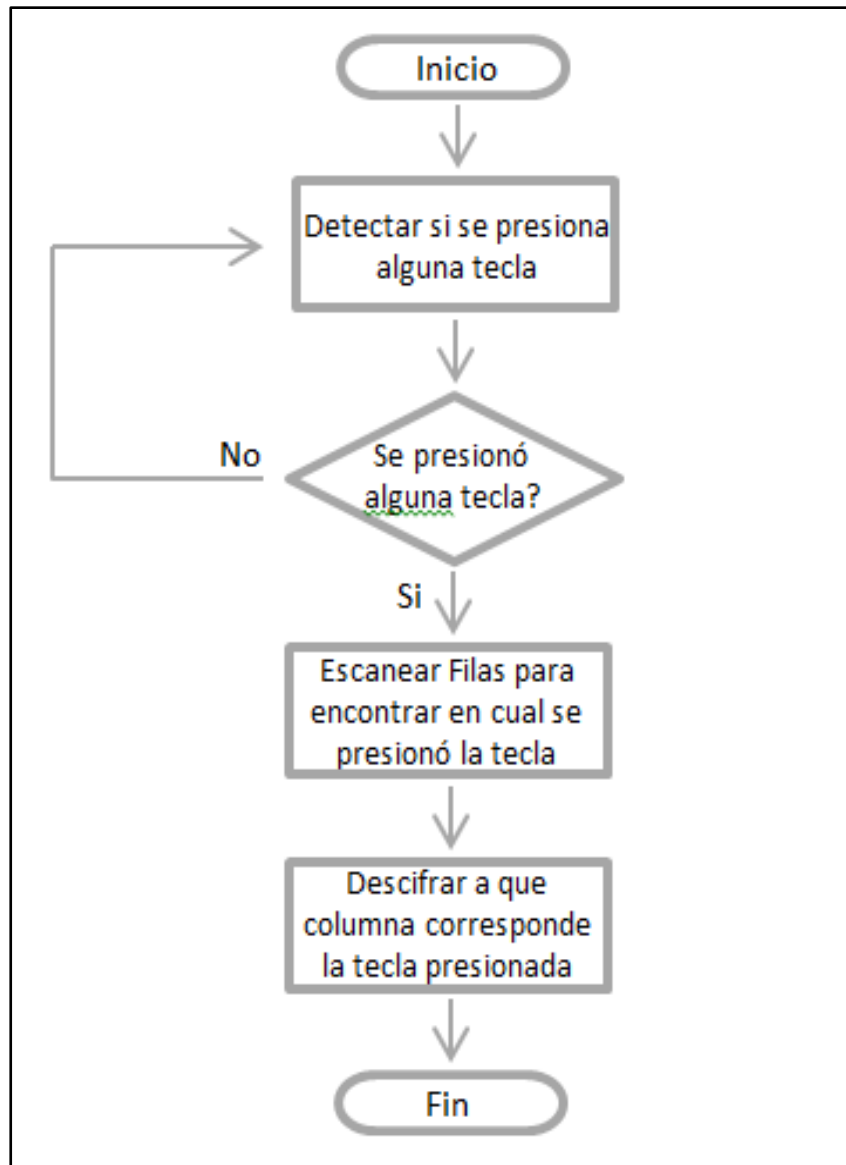
El primero es un teclado matricial de 6x6 que gracias a su configuración permite usar 36 teclas con solo 12 líneas de conexión. Su estructura es como se muestra en la figura 19 mientras que su funcionamiento es descrito en el diagrama de flujo mostrado en la figura 20.

Figura 19. Circuito de teclado matricial 6x6



Fuente: Este Trabajo.

Figura 20. Diagrama de flujo control de teclado matricial



Fuente: Este Trabajo.

El segundo componente de la interfaz de usuario que se utilizó, es una pantalla LCD de tipo alfanumérica en configuración 16X2, es decir que posee dos líneas de 16 caracteres cada una. La línea superior se usa para visualizar el texto enviado y la línea inferior se usa para visualizar el texto recibido. Su aspecto físico se muestra en la figura 21 y sus pines de control son descritos en la tabla 9.

Figura 21. Aspecto físico de la pantalla LCD alfanumérica 16x2



Fuente: Este Trabajo.

Tabla 9. Pines Principales de pantalla LCD

Nombre del Pin	Descripción
Vss	Alimentación Negativa
Vdd	Alimentación Positiva
Vee	Ajuste de Contraste
RS	Modo comando RS = 0; Modo Dato RS = 1
R/W	Escribir en el registro R/W = 0; Leer desde el Registro R/W = 1
D0 – D7	Pines de Datos 8 bits

Fuente: este Trabajo.

2.3.3 Funciones del microcontrolador PIC16F876A. Este microcontrolador fue utilizado para interactuar con la interfaz de usuario, además de controlar el módulo BlueSMiRF y el teléfono móvil. Sus tareas principales se muestran a continuación:

- Detectar los caracteres digitados por el usuario mediante el teclado Matricial.
- Leer y escribir los caracteres enviados y recibidos en la pantalla LCD.
- Enviar los comandos de configuración y control al módulo BlueSMiRF para establecer una conexión con el teléfono móvil.
- Enviar comandos AT al teléfono móvil para establecer y terminar llamadas, enviar tonos DTMF y configurar algunos parámetros.
- Compartir información con el PIC16F628A correspondiente a los tonos DTMF recibidos por el decodificador MT8870DE y a las respuestas dadas por el módulo BlueSMiRF y el teléfono celular.

Dichas funciones se implementaron mediante el software para computador CCS C Compiler, que permite programar en lenguaje C para luego compilar en lenguaje ensamblador y grabar el firmware en la memoria del PIC.

El código en lenguaje C correspondiente a la programación de este microcontrolador, se describe en el ANEXO C.

2.3.4 Funciones del microcontrolador PIC16F628A. La función principal de este microcontrolador, es la interpretación del dato digital de 4 bits que proporciona el circuito integrado MT8870DE una vez haya recibido un tono DTMF en su entrada.

Además, se encarga de intercambiar información con el PIC16F876A enviándole por puerto serie, el carácter recibido desde el decodificador MT8870DE y recibiendo parámetros que le permitan conocer el estado de una llamada en proceso. Sus principales funciones se describen a continuación:

- Recibir los datos provenientes del MT8870DE, interpretarlos y enviarlos para que se puedan mostrar en pantalla.
- Recibir del teléfono, los datos correspondientes cuando se presenta una llamada entrante.
- Recibir mensajes de respuesta del teléfono y del módulo BlueSMiRF y transmitirlos al PIC16F876A.

El código en lenguaje C correspondiente a la programación de este microcontrolador, se describe en el ANEXO D.

2.3.5 Dispositivos BlueSMiRF y ZTE N281. Esta pareja de dispositivos fueron usados para realizar la tarea de comunicarse con la red celular. Sus funciones se describen a continuación:

- Recibir comandos AT de configuración y control por parte del PIC16F876A.
- Enviar las respuestas correspondientes a través del PIC16F628A.
- Enviar al MT8870DE la señal de audio recibida a través de la red celular.
- Ingresar a la red celular para transmitir información.

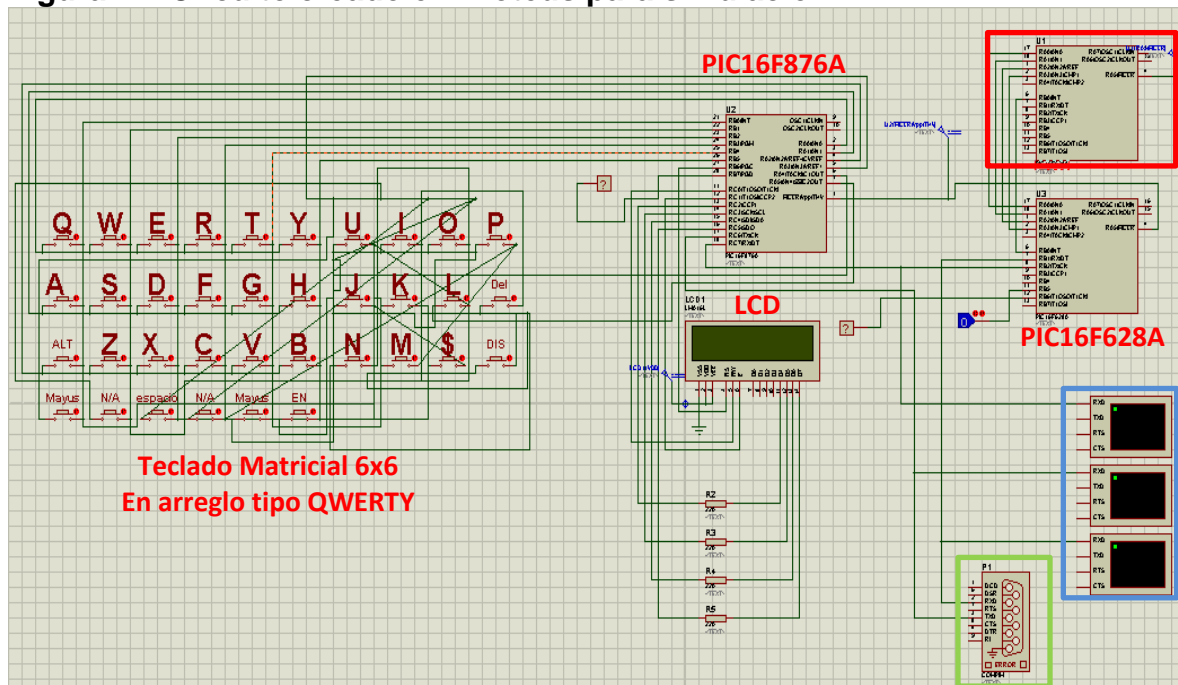
2.4 SIMULACIÓN DE LOS CIRCUITOS DISEÑADOS Y MONTAJE DE PROTOTIPOS

La simulación de los circuitos diseñados se hizo necesaria debido a que permite un acercamiento al funcionamiento real para el dispositivo electrónico diseñado.

De esta forma, en la etapa de simulación de los circuitos diseñados para el dispositivo electrónico, se utilizó el software de computador Proteus v7.9 desarrollado por la compañía Labcenter Electronics, que gracias a su sistema de modelamiento virtual (VSM), componentes animados y modelos de microcontroladores, permitió tener un acercamiento bastante preciso de cuál es el funcionamiento del dispositivo electrónico.

Es así como se creó el circuito virtual mostrado en la figura 22 para realizar las pruebas pertinentes y observar el funcionamiento del dispositivo.

Figura 22. Circuito creado en Proteus para simulación



Fuente: Este Trabajo.

En la figura anterior, el componente encerrado en el recuadro rojo, simula el comportamiento del decodificador MT8870DE debido a que este no se encontraba disponible en el software de simulación; los componentes encerrados en el recuadro azul, permiten visualizar los datos transmitidos por puerto serie en cada una de las líneas de este tipo y el componente encerrado en el recuadro verde corresponde al COMPIMM que permite interactuar con un puerto serie real para

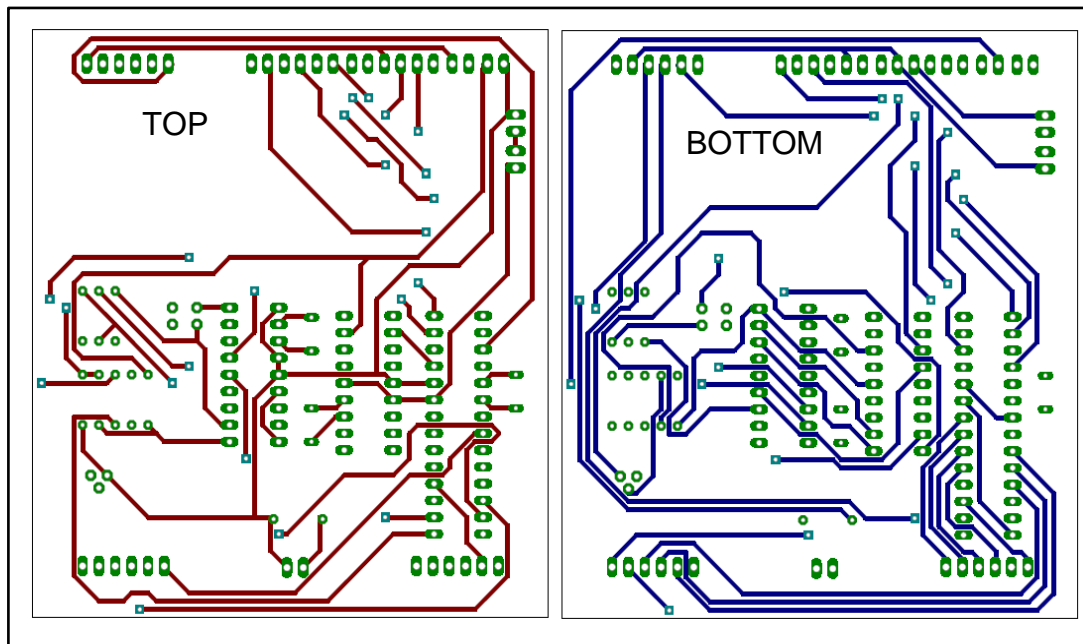
realizar una simulación híbrida entre los componentes virtuales y los componentes reales para el caso del BlueSMiRF y el teléfono celular.

Una vez finalizada la etapa de simulación, se implementan los prototipos de los dispositivos basándose en el circuito descrito en el ANEXO E, para verificar el comportamiento real y realizar las pruebas y correcciones pertinentes.

2.5 IMPLEMENTACIÓN DE DOS DISPOSITIVOS ELECTRÓNICOS

Para la implementación de los dispositivos electrónicos, se fabricaron dos circuitos impresos doble capa, mediante el método de transmisión térmica. El primero contiene los circuitos integrados principales y los conectores para los periféricos. Su aspecto se muestra en las Figura 23.

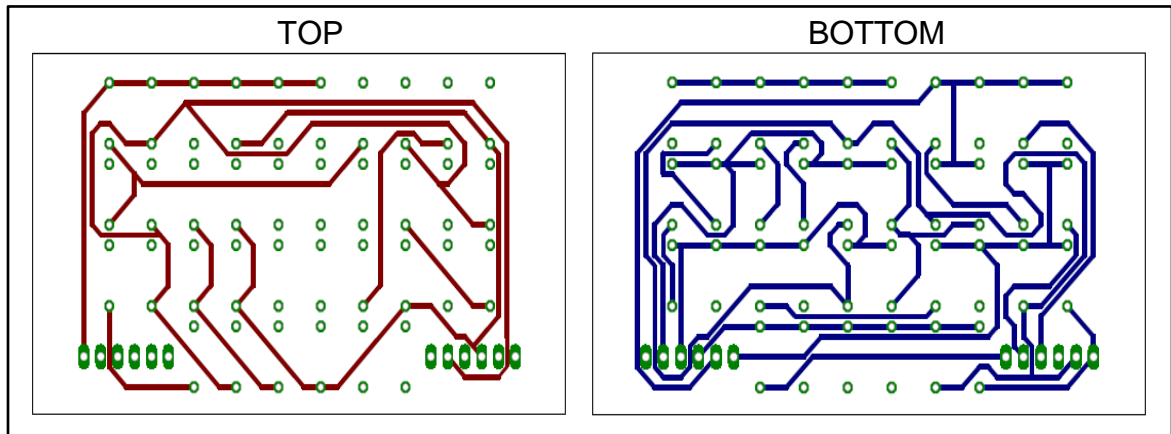
Figura 23. Circuito impreso principal para el dispositivo electrónico



Fuente: Este trabajo.

El segundo circuito impreso implementado, corresponde al teclado matricial de 6x6 en arreglo tipo QWERTY, el cual se muestra en la figura 24.

Figura 24. Circuito impreso para el teclado matricial tipo QWERTY



Fuente: Este trabajo.

Luego se realizó el montaje de los componentes en los circuitos impresos y se fabricó las cubiertas en fibra de vidrio para darle un mejor aspecto físico a los dispositivos.

2.6 CAPACITACIÓN A UNA PERSONA CON DISCAPACIDAD AUDITIVA PARA USAR EL DISPOSITIVO

Antes de realizar las pruebas de funcionamiento con los usuarios sordos, el autor del proyecto realizó un estudio acerca de las implicaciones éticas que pueda tener el uso del dispositivo en las personas con discapacidad auditiva. Es decir, que debe asegurarse que el uso del dispositivo no genere problemas éticos en la persona discapacitada.

Dicho estudio se realizó mediante la formulación de una encuesta cuyo formato se muestra en el Anexo F y que se aplicó a 3 personas con discapacidad auditiva asistentes al aula de apoyos tecnológicos para personas con discapacidad de la Universidad de Nariño. Estas personas fueron Fabricio Tovar, Andrés López y Yuly Patricia Portillo.

Dicho estudio se llevó a cabo en la Universidad de Nariño sede centro con la ayuda de una persona intérprete quien se comunicó mediante lenguaje de señas permitiendo al autor obtener las opiniones de las personas encuestadas y conocer si se ataca en cierto modo a esta sociedad. La capacitación se dividió fundamentalmente en cuatro etapas:

En primera etapa se explicó detalladamente a los usuarios sobre las características y el funcionamiento del dispositivo.

En la segunda etapa se realizó una encuesta donde se mencionan los aspectos éticos y al final se cuestiona si las personas acceden a realizar pruebas de funcionamiento con el dispositivo.

En la tercera etapa, se procedió a establecer una comunicación portátil y a distancia a través de la tecnología celular entre dos personas con discapacidad auditiva usando únicamente saldo en minutos. Como lo muestra las figura 25.

Por último, la cuarta etapa se llevó a cabo mediante la realización de una encuesta que permitió al autor del proyecto conocer si se cumple con el objetivo principal.

Cabe aclarar que a la capacitación asistieron 8 personas, pero algunas de ellas no estuvieron dispuestas a participar de las encuestas más si estuvieron presentes en la explicación y pruebas de funcionamiento del dispositivo.

Figura 25. Comunicación entre personas sordas mediante el dispositivo



Fuente: Este trabajo

3. RESULTADOS

A continuación se describen los resultados obtenidos en cada una de las etapas de acuerdo con la metodología aplicada a lo largo del presente proyecto.

3.1 CONTROL DEL TELÉFONO CELULAR ZTE N281

Una vez establecida la conexión por Bluetooth del tipo puerto serie a través del Protocolo RFCOMM y el perfil SPP entre el teléfono y el módulo BlueSMiRF. Se logró enviar, mediante el software Hyperterminal, una serie de comandos AT específicos al teléfono para iniciar y terminar llamadas de voz, así como para el envío de tonos DTMF y otro tipo de configuraciones. Los comandos utilizados fueron:

- ATD<numero>;<CR> // comando para iniciar una llamada de voz a un número //determinado.
- ATA<CR> // Comando para contestar una llamada entrante.
- ATH<CR> // Comando para terminar una llamada en curso.
- ATE<0/1><CR> // Comando para configuración de eco desactivado o activado.
- AT+CPAS<CR> // comando que solicita información sobre el estado de la // llamada.
- AT+CMUT=<0/1><CR> // Comando para inhabilitar el micrófono del teléfono //para evitar ruidos del ambiente.
- AT+VTS=<DTMF><CR> // Comando para el envío de un tono DTMF a través //de la red celular.

3.2 SIMULACIÓN HÍBRIDA Y PROTOTIPO DEL DISPOSITIVO ELECTRÓNICO

La Figura 26 muestra la conexión realizada para la simulación Híbrida.

Figura 26. Conexión para simulación híbrida mediante Proteus

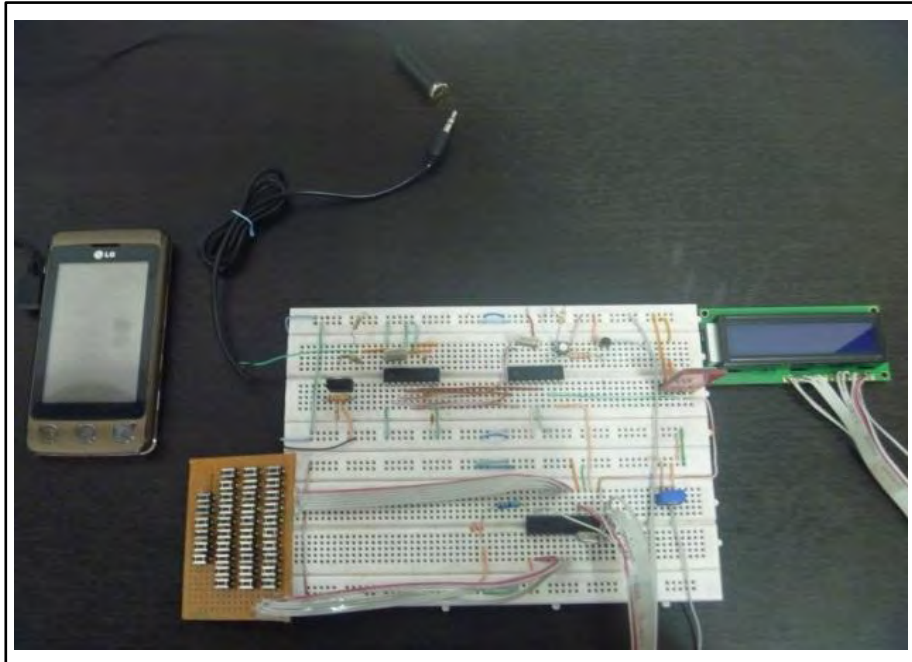


Fuente: Este Trabajo.

Por medio de la simulación se logró corregir algunos errores en la programación del firmware de los microcontroladores y se comprobó el correcto funcionamiento de gran parte del dispositivo electrónico, excepto la recepción de tonos DTMF mediante el auricular del teléfono celular debido a que el software Proteus no contiene en sus librerías el CI MT88870DE.

Luego, se implementaron dos prototipos del dispositivo electrónico cuyo aspecto físico es como se muestra en la figura 27.

Figura 27. Aspecto físico prototipo de dispositivo electrónico



Fuente: Este Trabajo.

Finalmente, una vez implementados los 2 prototipos necesarios para la comunicación, se logró comprobar el correcto funcionamiento de los circuitos haciendo énfasis en la etapa de recepción de tonos DTMF a través del MT8870DE puesto que no pudo ser simulada. Además se realizaron pruebas de comunicación que resultaron exitosas.

3.3 CARACTERÍSTICAS DE LOS DISPOSITIVOS ELECTRÓNICOS

Las dimensiones de cada dispositivo electrónico son de 9.5cm de largo, 9.5cm de ancho y 2.7cm de espesor.

Además, cada dispositivo electrónico cuenta con un LED y un pulsador en la parte superior para propósitos de señalización y sincronización, un interruptor para el encendido del dispositivo, un conector para el cargador de batería y un conector tipo Jack de 3.5mm para el ingreso de las señales de audio al dispositivo.

Su aspecto se muestra en la figura 28.

Figura 28. Aspecto físico de los dispositivos electrónicos



Fuente: Este Trabajo.

Cabe aclarar que el dispositivo se alimenta mediante una batería de Li-on de 3.7v con capacidad de 1200 mAh lo que permite el uso portátil del mismo y teniendo en cuenta que el consumo de corriente promedio es de 32mA, es decir un consumo de potencia de solo 118 mWatts, se puede determinar el tiempo de descarga de la batería de acuerdo con la siguiente expresión:

$$\text{tiempo de descarga} = \frac{\text{capacidad de corriente (mAh)}}{\text{consumo del dispositivo (mA)}} \quad (1)$$

$$\text{tiempo de descarga} = \frac{1200\text{mAh}}{32\text{mA}} = 37.5 \text{ h} \quad (2)$$

Así, el dispositivo cuenta con una autonomía de 37 horas aproximadamente.

Finalmente, cabe mencionar que el costo total para la fabricación de cada dispositivo electrónico fue de \$193.600 pesos discriminados como se muestra en la tabla 10.

Tabla 10. Costos de fabricación del dispositivo electrónico

Producto	Costo unitario en pesos	Cantidad	Costo en pesos
Microcontrolador PIC16F876A	8.700	1	8.700
Microcontrolador PIC16F628A	4.100	1	4.100
Módulo BlueSMiRF	96.000	1	96.000
Display LCD	11.000	1	11.000
Pulsadores de teclado	200	34	6.800
Decodificador MT8870	7.500	1	7.500
XTAL	1.000	3	3.000
Componentes pasivos	--	--	3500
Conectores y switch	--	--	3.000
Cubierta	50.000	1	50.000
TOTAL			193.600

Fuente: Este trabajo.

3.4 RESULTADOS DE LA CAPACITACIÓN A USUARIOS SORDOS

Como resultados de la capacitación se obtuvo que las 3 personas entrevistadas manifestaron claramente que el dispositivo no les afecta negativamente sino que por el contrario les ayuda a mejorar su calidad de vida. Dichas respuestas se graban en video para corroborar su veracidad.

Luego de capacitación y la prueba de funcionamiento, las personas participan en la encuesta final manifestando que efectivamente el dispositivo desarrollado permite comunicación a través de la red celular en tiempo real de forma económica.

En el Anexo G se dispone de un manual de usuario para la correcta utilización del dispositivo.

4. CONCLUSIONES

Las personas con discapacidad auditiva mejoran su calidad de vida puesto que poseen un nuevo método para realizar comunicaciones portátiles a distancia que les permite comunicarse a través de minutos, logrando acceder a promociones, y tarifas preferenciales al igual que los usuarios sin discapacidades físicas.

La comunicación por llamadas de voz tiene ventajas como la unilateralidad de saldo, es decir que solamente un usuario debe tener saldo disponible para realizar una comunicación lo que significa mayor economía para el usuario, quien incluso puede comunicarse completamente gratis. Además de la disponibilidad de comunicación en tiempo real por mensajes escritos en lugares donde aún no están desplegadas redes de tercera y cuarta generación.

El uso de la tecnología Bluetooth permite que el dispositivo sea altamente compatible con muchos modelos de teléfonos celulares disponibles en el mercado, garantizando la flexibilidad del dispositivo en cuanto a las preferencias del usuario en el modelo de teléfono a utilizar.

El costo de fabricación para cada dispositivo electrónico es muy bajo en comparación con otros dispositivos existentes en el mercado nacional y que no prestan los beneficios otorgados por el dispositivo desarrollado en este trabajo.

El aula de apoyos tecnológicos para personas con discapacidad de la universidad de Nariño adquiere dos dispositivos electrónicos que permiten que los asistentes con discapacidad auditiva se comuniquen fuera del aula y desde casi cualquier lugar.

RECOMENDACIONES

Mejorar la interfaz de usuario implementando una pantalla grafica para lograr una mejor señalización de los mensajes y que los usuarios puedan enviar emoticones y otro tipo de caracteres especiales.

Adaptar los códigos de programación de los microcontroladores para que identifiquen el fabricante del teléfono celular y adapten los comandos AT compatibles con cada fabricante.

Desarrollar una aplicación para Smartphone con fin de integrar la interfaz de usuario del teléfono celular con el dispositivo electrónico y reemplazar algunas funciones de los microcontroladores en cuanto al envío de tonos DTMF.

Continuar con investigaciones en el aula de apoyos tecnológicos para personas con discapacidad de la Universidad de Nariño, para ofrecer apoyo tecnológico y satisfacer las múltiples necesidades de la sociedad discapacitada.

BIBLIOGRAFÍA

ALULEMA QUITAQUIS, Martha. Estudio de la comunicación con comandos AT y microcontroladores caso práctico implementación de un prototipo sistema de gestión de alarma para viviendas con monitoreo mediante telefonía celular. Riobamba, 2010, 118p. Tesis de grado (Ingeniería en Electrónica y Computación). Escuela Superior Politécnica de Chimborazo. Disponible en el repositorio de datos DSpace de la Escuela Superior Politécnica de Chimborazo:
<<http://dspace.esoch.edu.ec/>>

ARCHUNDIA, Francisco. Wireless Personal Area Network (WPAN) & Home Networking. Puebla, México, 2003. Tesis profesional (Licenciatura en Ingeniería en Electrónica y Comunicaciones). Universidad de las Américas Puebla. Departamento de Ingeniería Electrónica. Disponible en la colección de Tesis digitales de la Universidad de las Américas Puebla:
<http://catarina.udlap.mx/u_dl_a/tales/documentos>

GUTIÉRREZ REINA, Daniel. Sistema pasarela Bluetooth para una red de sensores zigbee. En: Capítulo 4 Tecnología Bluetooth. Sevilla, España, 2009. Proyecto fin de carrera. Universidad de Sevilla. Escuela Técnica Superior de Ingeniería. Disponible en el repositorio de trabajos y proyectos de fin de estudios de la Escuela Técnica Superior de Ingeniería:
<<http://bibing.us.es/proyectos/abreproy/40048/fichero/VOLUMEN+1.+MEMORIA%252F4.+Tecnolog%C3%ADa+Bluetooth.pdf>>

HEINE, Gunnar. GSM Networks: Protocols, Terminology, and Implementation. Londres: Artech House, 1999. 417p.

MARTÍNEZ, Evelio. La evolución de la telefonía móvil [en línea]. <<http://www.eveliux.com/mx/la-evolucion-de-la-telefonía-movil.php>>

NIÑO, Diego. (2011). Telefonía Móvil. [en línea]. <http://www.slideshare.net/diego15289/tecnologías-2g-y-25g#btnPrevious>

PALACIOS ROJAS, Diego. ROJAS PAZ, Carlos. Diseño e implementación de una interfaz electrónica para el monitoreo de vehículos de transporte público en tiempo real en la ciudad de Pasto. Pasto, 2010, 120p. Trabajo de Grado (Ingeniería Electrónica). Universidad de Nariño. Facultad de ingeniería. Disponible en el

catálogo en línea de la biblioteca de la Universidad de Nariño:
<<http://biblioteca.udenar.edu.co/atenea/>>

PALACIOS, Enrique, DOMÍNGUEZ, Fernando, LÓPEZ, Lucas. Microcontrolador PIC16F84 Desarrollo de proyectos. Madrid: RA-MA Editorial, 2006.621p.

PATRÓN, David. Simulación de tramas de WCDMA. Puebla, México, 2004. Tesis profesional (Licenciatura en Ingeniería en Electrónica y Comunicaciones). Universidad de las Américas Puebla. Departamento de Ingeniería Electrónica. Disponible en la colección de Tesis digitales de la Universidad de las Américas Puebla: <http://catarina.udlap.mx/u_dl_a/tales/documentos/>

Reocities. Tonos duales de multi frecuencia (DTMF) [en línea].
<http://www.reocities.com/SiliconValley/bridge/6429/derecho.html>

RUIZ DÍEZ, Pablo. Diseño y construcción de un vehículo de inspección remota mediante marcación por tonos. Madrid, 2009, 72p. Proyecto fin de carrera (Ingeniería de Sistemas y Automática). Universidad Carlos III de Madrid. Documento en línea disponible en el repositorio Institucional de la Universidad Carlos III de Madrid: <<http://e-archivo.uc3m.es/>>

Zarlink Semiconductor. MT8870D/MT8870D-1 Integrated DTMF Receiver [en línea]. <<http://datasheet.elcodis.com/pdf2/96/81/968170/mt8870dede-1.pdf>>

ANEXOS

Anexo A. Especificaciones del microcontrolador PIC16F876A



PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input
DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference (V_{REF}) module
 - Programmable input multiplexing from device inputs and internal voltage reference
 - Comparator outputs are externally accessible

Special Microcontroller Features:


- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I ² C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

Anexo B. Especificaciones del microcontrolador PIC16F628A



MICROCHIP PIC16F627A/628A/648A

18-pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

High-Performance RISC CPU:

- Operating speeds from DC – 20 MHz
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes
- 35 single-word instructions:
 - All instructions single cycle except branches

Special Microcontroller Features:

- Internal and external oscillator options:
 - Precision internal 4 MHz oscillator factory calibrated to $\pm 1\%$
 - Low-power internal 48 kHz oscillator
 - External Oscillator support for crystals and resonators
- Power-saving Sleep mode
- Programmable weak pull-ups on PORTB
- Multiplexed Master Clear/Input-pin
- Watchdog Timer with independent oscillator for reliable operation
- Low-voltage programming
- In-Circuit Serial Programming™ (via two pins)
- Programmable code protection
- Brown-out Reset
- Power-on Reset
- Power-up Timer and Oscillator Start-up Timer
- Wide operating voltage range (2.0-5.5V)
- Industrial and extended temperature range
- High-Endurance Flash/EEPROM cell:
 - 100,000 write Flash endurance
 - 1,000,000 write EEPROM endurance
 - 40 year data retention

Low-Power Features:

- Standby Current:
 - 100 nA @ 2.0V, typical
- Operating Current:
 - 12 μA @ 32 kHz, 2.0V, typical
 - 120 μA @ 1 MHz, 2.0V, typical
- Watchdog Timer Current:
 - 1 μA @ 2.0V, typical
- Timer1 Oscillator Current:
 - 1.2 μA @ 32 kHz, 2.0V, typical
- Dual-speed Internal Oscillator:
 - Run-time selectable between 4 MHz and 48 kHz
 - 4 μs wake-up from Sleep, 3.0V, typical

Peripheral Features:

- 16 I/O pins with individual direction control
- High current sink/source for direct LED drive
- Analog comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference (V_{REF}) module
 - Selectable internal or external reference
 - Comparator outputs are externally accessible
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Timer1: 16-bit timer/counter with external crystal/clock capability
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Capture, Compare, PWM module:
 - 16-bit Capture/Compare
 - 10-bit PWM
- Addressable Universal Synchronous/Asynchronous Receiver/Transmitter USART/SCI

Device	Program Memory	Data Memory		I/O	CCP (PWM)	USART	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)					
PIC16F627A	1024	224	128	16	1	Y	2	2/1
PIC16F628A	2048	224	128	16	1	Y	2	2/1
PIC16F648A	4096	256	256	16	1	Y	2	2/1

Anexo C. Código en lenguaje C para el PIC16F876A

```
#include <16f876a.h> // se define el pic en el que se grabara el programa
// parámetros de configuración del pic
#fuses XT, NOWDT, NOPROTECT, NOLVP, NOBROWNOUT
// puertos a utilizar se definen como digitales
#use standard_io(A)
#use standard_io(B)
#use standard_io(C)
#use delay(clock=4000000) // velocidad del cristal oscilador
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7) // define parámetros de comunicación serial
//pines a usar con el display LCD
#define LCD_DB4 PIN_C2
#define LCD_DB5 PIN_C3
#define LCD_DB6 PIN_C4
#define LCD_DB7 PIN_C5
#define LCD_RS PIN_B6
#define LCD_RW PIN_C0
#define LCD_E PIN_C1
#include <flex_lcd.c> // librería para la utilización del display LCD
// se declaran las subrutinas del programa
void fila_5(void);
#SEPARATE
void corre_lcd(void);
#SEPARATE
void fila_1(void);
#SEPARATE
void fila_2(void);
#SEPARATE
void fila_3(void);
#SEPARATE
void fila_4(void);
#SEPARATE
void fila_6(void);
#SEPARATE
void envia_caracter(void);
#SEPARATE
void sale_borra(void);
#SEPARATE
void corre_lcd_rx(void);
#SEPARATE
void cuelga(void);
#SEPARATE
void retardo(void);
// Se definen las variables a utilizar
int cont; // variable para escanear el teclado matricial 6x6
int MAYUS=0; // variable para cambiar de mayúsculas a minúsculas
int alt=0; // imprimir caracteres como números y algunos símbolos
int fin2=1; // corre los caracteres de la lcd una vez se ha llenado la pantalla
int fin=0; // variable para saber cuándo se llena la pantalla
int fin_rx=1; // variable para saber cuándo se llena la pantalla en recepción
char DTMF; // carácter DTMF a enviar
int veces; // establece las veces que se envía un tono DTMF, depende del carácter
char tecla; // guarda el carácter recibido por el puerto serie
int CONTESTA=0; // para saber si se ha contestado una llamada entrante
int ACTIVO=0; // para saber si hay una llamada en curso
int llamando=0; // para saber cuándo está entrando una llamada
int borra=0; // borrar carácter
// variable para almacenar los caracteres que se encuentran impresos en la LCD en el momento de correrlos
char caracter[17];
int ocupado=0; // para saber si el programa está ocupado enviando un tono DTMF
int error=0; // para saber si el modulo celular responde con error
int cont_error=0; // cuenta errores consecutivos para determinar perdida de conexión
int posicion=0; // posición de la dirección bluetooth
```

```

char direccion[20]; // datos de la dirección bluetooth
// rutina de interrupción al recibir un carácter por el pin rx de la comunicación serial
#INT_RDA
void RDA_isr() {
// espera a que llegue un carácter por el pin rx del puerto serie y cuando llega lo guarda en la variable tecla
tecla=getc();
// rutina que borra un carácter de rx en caso de recibir el DTMF de borrar
if (tecla=='\') {
    if (fin_rx>0) {
        fin_rx=fin_rx-1;
        lcd_gotoxy(fin_rx,2); // devuelve una posición en la línea de recepción de la LCD
        lcd_putc(' '); // imprime carácter espacio simulando un borrado de carácter
// después de realizar el proceso vuelve a la línea de transmisión de la LCD en el último punto donde estaba
// antes de recibir el carácter
        lcd_gotoxy((fin+1),1);
    }
}
else {
// si no se recibe caracteres especiales, se imprime en la LCD el carácter recibido
if (tecla!='°' && tecla!='~' && tecla!='¿' && tecla!='¡' && tecla!=0 && tecla!='\']) {
// llama a rutina para correr los caracteres de la lcd una vez se halla llenado la pantalla
corre_lcd_rx();
lcd_gotoxy(fin_rx,2);
printf(lcd_putc,"%c",tecla);
lcd_gotoxy((fin+1),1);
fin_rx=fin_rx+1;
}
}
// rutina para saber que está entrando una llamada y así habilitar la comunicación
if (tecla=='°') {
    llamando=1; // pone a 1 la variable indicando que está entrando una llamada
// si la tecla habilitar conexión ha sido presionada entonces procede a contestar
if (CONTESTA==1) {
    output_high(PIN_B7); // indica al usuario que el módulo está ocupado
// comando para contestar llamada
    putc('A');
    putc('T');
    putc('A');
    putc('\r');
    delay_ms(1000);
//comando para que el pic de decodificación se entere que se ha contestado la llamada y habilite la recepción
    putc('A');
    putc('T');
    putc('+');
    putc('C');
    putc('P');
    putc('A');
    putc('S');
    putc('\r');
    tecla=0;
    lcd_putc("\f"); // vacía la LCD
// reinicio todas las variables involucradas en el proceso
    MAYUS=0;
    alt=0;
    fin2=1;
    fin=0;
    fin_rx=1;
    borra=0;
    CONTESTA=0;
    llamando=0;
    output_low(PIN_B7); // indica al usuario que el módulo celular está desocupado
}
}
// rutina para saber cuándo se ha recibido un ok por parte del módulo celular
if (tecla=='¡') {
    if (ACTIVO==1 && ocupado==1) {
        cont_error=0;
    }
}

```

```

        error=0;
        ocupado=0;
    }
}
// rutina para saber cuándo se ha recibido un error por parte del módulo celular
if (tecla=='I') {
    if (ACTIVO==1 && ocupado==1) {
        error=1;
        ocupado=0;
    }
}
if (tecla=='~') { // rutina de emparejamiento con el dispositivo bluetooth
    putc('I');
    putc('\r');
    while (posicion<12) {
        direccion[posicion]=getch();
        posicion=posicion+1;
    }
    posicion=0;

printf("C,%c%c%c%c%c%c%c%c%c%c%c\r",direccion[0],direccion[1],direccion[2],direccion[3],direccion[4],direccion[5],di
reccion[6],direccion[7],direccion[8],direccion[9],direccion[10],direccion[11]);
    delay_ms(20000);
    putc('A');
    putc('T');
    putc('E');
    putc('0');
    putc('\r');
    delay_ms(100);
    puts("AT+VTD=1");
    output_low(PIN_B7);
}
if (tecla=='¿') {
    putc('C');
    putc('\r');
    delay_ms(10000);
    putc('A');
    putc('T');
    putc('E');
    putc('0');
    putc('\r');
    delay_ms(100);
    puts("AT+VTD=1");
    output_low(PIN_B7);
}
}
// programa principal
void main() {
    lcd_init();
    delay_ms(2000);
    lcd_putc("\f");
    // comando para que el modulo bluetooth entre en modo de configuración
    putc('$');
    putc('$');
    putc('$');
    // enciende el led de estado para indicar que el dispositivo está ocupado
    output_high(PIN_B7);
    enable_interrupts(int_rda); // habilita interrupción por recepción en el puerto serie
    enable_interrupts(GLOBAL); // habilita interrupciones globales
    cont=0;
    MAYUS=0;
    port_b_pullups(TRUE); // habilita resistencias pull up para el teclado matricial
    while(true){ // rutina para escanear la tecla pulsada en el teclado matricial de 6x6
    // por medio de la variable cont se envía un cero en cada fila y luego se pasa a sensar la entrada de cada una
    // de las 6 columnas. en caso de que la entrada de alguna columna sea cero significa que se ha pulsado una
    // tecla de la fila que este con salida en cero
        if (cont==0) {

```



```

    output_A(0b00111110);
    fila_1(); //
    cont=cont+1;
}
if (cont==1) {
    output_A(0b00111101);
    fila_2();
    cont=cont+1;
}
if (cont==2) {
    output_A(0b00111011);
    fila_3();
    cont=cont+1;
}
if (cont==3) {
    output_A(0b00110111);
    fila_4();
    cont=cont+1;
}
if (cont==4) {
    output_A(0b00101111);
    fila_5();
    cont=cont+1;
}
if (cont==5) {
    output_A(0b00011111);
    fila_6();
    cont=0;
}
}
}
}
void fila_1() { // rutina para sensar que tecla de la fila 1 se ha presionado
    while(input(PIN_B0)==0){ // primera columna
        fin=fin+1; // indica que aumenta una posición en la LCD
        sale_borra(); // llama a subrutina para saber si viene de borrar un carácter
        corre_lcd(); // si se llenó la pantalla, va a correr los caracteres
    // rutina para imprimir el carácter de la tecla que se presionó detectando si es mayúscula minúscula o símbolo
        veces=2; // carga el número de tonos DTMF necesarios para representar la tecla
        DTMF=2; // carga el número DTMF a enviar
        if (MAYUS==1) {
            lcd_putc('Q');
        }
        else {
            if (alt==1){
                lcd_putc('#');
            }
            else {
                lcd_putc('q');
            }
        }
        while(input(PIN_B0)==0){ // espera a que deje de pulsar la tecla para salir
            retardo(); // se llama a un retardo para evitar rebotes del teclado
        }
    // llama a subrutina para enviar el carácter por medio de la red celular
        envia_caracter();
    }
}
// repite el procedimiento para las demás columnas
while(input(PIN_B1)==0){ // segunda columna
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=2;
    DTMF='8';
    if (MAYUS==1) {
        lcd_putc('W');
    }
    else {

```

```

    if (alt==1){
        lcd_putc('1');
    }
    else {
        lcd_putc('w');
    }
}
while(input(PIN_B1)==0){
    retardo();
}
envia_caracter();
}
while(input(PIN_B2)==0){ // tercera columna
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=1;
    DTMF='5';
    if (MAYUS==1) {
        lcd_putc('E');
    }
    else {
        if (alt==1){
            lcd_putc('2');
        }
        else {
            lcd_putc('e');
        }
    }
    while(input(PIN_B2)==0){
        retardo();
    }
    envia_caracter();
}
while(input(PIN_B3)==0){ // cuarta columna
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=2;
    DTMF='3';
    if (MAYUS==1) {
        lcd_putc('R');
    }
    else {
        if (alt==1){
            lcd_putc('3');
        }
        else {
            lcd_putc('r');
        }
    }
    while(input(PIN_B3)==0){
        retardo();
    }
    envia_caracter();
}
while(input(PIN_B4)==0){ // quinta columna
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=2;
    DTMF='5';
    if (MAYUS==1) {
        lcd_putc('T');
    }
    else {
        if (alt==1){

```

```

        lcd_putc('(');
    }
    else {
        lcd_putc('t');
    }
}
while(input(PIN_B4)==0){
    retardo();
}
envia_caracter();
}
while(input(PIN_B5)==0){ // sexta columna
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=2;
    DTMF='0';
    if (MAYUS==1) {
        lcd_putc('Y');
    }
    else {
        if (alt==1){
            lcd_putc(')');
        }
        else {
            lcd_putc('y');
        }
    }
    while(input(PIN_B5)==0){
        retardo();
    }
    envia_caracter();
}
return;
}
}
void fila_2() { // rutina para sensar que tecla de la fila 2 se ha presionado
    while(input(PIN_B0)==0){
        fin=fin+1;
        sale_borra();
        corre_lcd();
        veces=1;
        DTMF='1';
        if (MAYUS==1) {
            lcd_putc('A');
        }
        else {
            if (alt==1){
                lcd_putc('*');
            }
            else {
                lcd_putc('a');
            }
        }
    }
    while(input(PIN_B0)==0){
        retardo();
    }
    envia_caracter();
}
while(input(PIN_B1)==0){
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=1;
    DTMF='7';
    if (MAYUS==1) {
        lcd_putc('S');
    }
}
}

```

```

else {
    if (alt==1){
        lcd_putc('4');
    }
    else {
        lcd_putc('s');
    }
}
while(input(PIN_B1)==0){
    retardo();
}
envia_caracter();
}
while(input(PIN_B2)==0){
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=1;
    DTMF='4';
    if (MAYUS==1) {
        lcd_putc('D');
    }
    else {
        if (alt==1){
            lcd_putc('5');
        }
        else {
            lcd_putc('d');
        }
    }
}
while(input(PIN_B2)==0){
    retardo();
}
envia_caracter();
}
while(input(PIN_B3)==0){
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=1;
    DTMF='6';
    if (MAYUS==1) {
        lcd_putc('F');
    }
    else {
        if (alt==1){
            lcd_putc('6');
        }
        else {
            lcd_putc('f');
        }
    }
}
while(input(PIN_B3)==0){
    retardo();
}
envia_caracter();
}
while(input(PIN_B4)==0){
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=2;
    DTMF='4';
    if (MAYUS==1) {
        lcd_putc('G');
    }
    else {

```

```

        if (alt==1){
            lcd_putc('$');
        }
        else {
            lcd_putc('g');
        }
    }
    while(input(PIN_B4)==0){
        retardo();
    }
    envia_caracter();
}
while(input(PIN_B5)==0){
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=2;
    DTMF='C';
    if (MAYUS==1) {
        lcd_putc('H');
    }
    else {
        if (alt==1){
            lcd_putc(':');
        }
        else {
            lcd_putc('h');
        }
    }
    while(input(PIN_B5)==0){
        retardo();
    }
    envia_caracter();
}
return;
}
void fila_3() { // rutina para sensar que tecla de la fila 3 se ha presionado
    while(input(PIN_B0)==0){ // tecla para introducir caracteres especiales
        veces=2;
        DTMF='A';
        if (alt==0) {
            alt=1;
            MAYUS=0;
        }
        else {
            alt=0;
            MAYUS=0;
        }
        while(input(PIN_B0)==0){
            retardo();
        }
        envia_caracter();
    }
    while(input(PIN_B1)==0){
        fin=fin+1;
        sale_borra();
        corre_lcd();
        veces=2;
        DTMF='*';
        if (MAYUS==1) {
            lcd_putc('Z');
        }
        else {
            if (alt==1){
                lcd_putc('7');
            }
            else {

```

```

    lcd_putc('z');
  }
}
while(input(PIN_B1)==0){
  retardo();
}
envia_caracter();
}
while(input(PIN_B2)==0){
  fin=fin+1;
  sale_borra();
  corre_lcd();
  veces=2;
  DTMF='9';
  if (MAYUS==1) {
    lcd_putc('X');
  }
  else {
    if (alt==1){
      lcd_putc('8');
    }
    else {
      lcd_putc('x');
    }
  }
}
while(input(PIN_B2)==0){
  retardo();
}
envia_caracter();
}
while(input(PIN_B3)==0){
  fin=fin+1;
  sale_borra();
  corre_lcd();
  veces=1;
  DTMF='3';
  if (MAYUS==1) {
    lcd_putc('C');
  }
  else {
    if (alt==1){
      lcd_putc('9');
    }
    else {
      lcd_putc('c');
    }
  }
}
while(input(PIN_B3)==0){
  retardo();
}
envia_caracter();
}
while(input(PIN_B4)==0){
  fin=fin+1;
  sale_borra();
  corre_lcd();
  veces=2;
  DTMF='7';
  if (MAYUS==1) {
    lcd_putc('V');
  }
  else {
    if (alt==1){
      lcd_putc('?');
    }
    else {
      lcd_putc('v');
    }
  }
}

```

```

    }
  }
  while(input(PIN_B4)==0){
    retardo();
  }
  envia_caracter();
}
while(input(PIN_B5)==0){
  fin=fin+1;
  sale_borra();
  corre_lcd();
  veces=1;
  DTMF='2';
  if (MAYUS==1) {
    lcd_putc('B');
  }
  else {
    if (alt==1){
      lcd_putc('!');
    }
    else {
      lcd_putc('b');
    }
  }
  while(input(PIN_B5)==0){
    retardo();
  }
  envia_caracter();
}
return;
}
void fila_4() { // rutina para sensar que tecla de la fila 4 se ha presionado
  while(input(PIN_B0)==0){ // tecla para cambiar de mayúsculas a minúsculas
    veces=2;
    DTMF='#';
    if (alt==1){
      fin=fin+1;
      sale_borra();
      corre_lcd();
      lcd_putc('0');
    }
    else {
      if (MAYUS==0) {
        MAYUS=1;
      }
      else {
        MAYUS=0;
      }
    }
  }
  while(input(PIN_B0)==0){
    retardo();
  }
  envia_caracter();
}
while(input(PIN_B1)==0){
  while(input(PIN_B1)==0){
  }
}
while(input(PIN_B2)==0){
  veces=1;
  DTMF='8';
  if (alt==0) {
    fin=fin+1;
    sale_borra();
    corre_lcd();
    lcd_putc(' ');
  }
}

```

```

else {
    fin=fin+1;
    sale_borra();
    corre_lcd();
    lcd_putc('&');
}
while(input(PIN_B2)==0){
    retardo();
}
envia_caracter();
}
while(input(PIN_B3)==0){
    while(input(PIN_B3)==0){
    }
}
while(input(PIN_B4)==0){ // tecla para cambiar de mayúsculas a minúsculas
    veces=2;
    DTMF='#';
    if (alt==1){
        fin=fin+1;
        sale_borra();
        corre_lcd();
        lcd_putc('0');
    }
    else {
        if (MAYUS==0) {
            MAYUS=1;
        }
        else {
            MAYUS=0;
        }
    }
    while(input(PIN_B4)==0){
        retardo();
    }
    envia_caracter();
}
while(input(PIN_B5)==0){ // tecla para habilitar comunicación
ESPERA_CONTESTA:
    if (llamando==1) { //entra si hay una llamada entrante
// esta variable se utiliza en la interrupción serial para contestar si se ha presionado la tecla
        CONTESTA=1;
        ACTIVO=1;
    }
    else {
        output_high(PIN_B7); // indica al usuario que el modulo está ocupado
        delay_ms(200);
        putc('A');
        putc('T');
        putc('+');
        putc('C');
        putc('P');
        putc('A');
        putc('S');
        putc('\r');
        ACTIVO=1;
    }
}
// reinicio todas las variables involucradas en el proceso
MAYUS=0;
alt=0;
fin2=1;
fin=0;
fin_rx=1;
borra=0;
llamando=0;
delay_ms(200);
lcd_putc("\f"); // vacía la LCD

```



```

    output_low(PIN_B7); // indica al usuario que el modulo está desocupado
    while(input(PIN_B5)==0){
        retardo();
    }
}
return;
}
}
void fila_5() { // rutina para sensar que tecla de la fila 5 se ha presionado
    while(input(PIN_B0)==0){
        fin=fin+1;
        sale_borra();
        corre_lcd();
        veces=1;
        DTMF="*";
        if (MAYUS==1) {
            lcd_putc('U');
        }
        else {
            if (alt==1){
                lcd_putc('_');
            }
            else {
                lcd_putc('u');
            }
        }
        while(input(PIN_B0)==0){
            retardo();
        }
        envia_caracter();
    }
    while(input(PIN_B1)==0){
        fin=fin+1;
        sale_borra();
        corre_lcd();
        veces=1;
        DTMF="9";
        if (MAYUS==1) {
            lcd_putc('I');
        }
        else {
            if (alt==1){
                lcd_putc('-');
            }
            else {
                lcd_putc('i');
            }
        }
        while(input(PIN_B1)==0){
            retardo();
        }
        envia_caracter();
    }
    while(input(PIN_B2)==0){
        fin=fin+1;
        sale_borra();
        corre_lcd();
        veces=1;
        DTMF="C";
        if (MAYUS==1) {
            lcd_putc('O');
        }
        else {
            if (alt==1){
                lcd_putc('+');
            }
            else {
                lcd_putc('o');
            }
        }
    }
}

```

```

    }
  }
  while(input(PIN_B2)==0){
    retardo();
  }
  envia_caracter();
}
while(input(PIN_B3)==0){
  fin=fin+1;
  sale_borra();
  corre_lcd();
  veces=2;
  DTMF='1';
  if (MAYUS==1) {
    lcd_putc('P');
  }
  else {
    if (alt==1){
      lcd_putc('@');
    }
    else {
      lcd_putc('p');
    }
  }
}
while(input(PIN_B3)==0){
  retardo();
}
envia_caracter();
}
while(input(PIN_B4)==0){
  fin=fin+1;
  sale_borra();
  corre_lcd();
  veces=2;
  DTMF='B';
  if (MAYUS==1) {
    lcd_putc('J');
  }
  else {
    if (alt==1){
      lcd_putc(';');
    }
    else {
      lcd_putc('j');
    }
  }
}
while(input(PIN_B4)==0){
  retardo();
}
envia_caracter();
}
while(input(PIN_B5)==0){
  fin=fin+1;
  sale_borra();
  corre_lcd();
  veces=2;
  DTMF='6';
  if (MAYUS==1) {
    lcd_putc('K');
  }
  else {
    if (alt==1){
      lcd_putc('%');
    }
    else {
      lcd_putc('k');
    }
  }
}

```

```

    }
    while(input(PIN_B5)==0){
        retardo();
    }
    envia_caracter();
}
return;
}
void fila_6() { // rutina para sensar que tecla de la fila 6 se ha presionado
while(input(PIN_B0)==0){
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=1;
    DTMF='#';
    if (MAYUS==1) {
        lcd_putc('L');
    }
    else {
        if (alt==1){
            lcd_putc('^');
        }
        else {
            lcd_putc('I');
        }
    }
    while(input(PIN_B0)==0){
        retardo();
    }
    envia_caracter();
}
while(input(PIN_B1)==0){ //tecla para borrar carácter
    veces=1;
    DTMF='0';
    if (alt==0){
        if (fin>0) {
            lcd_gotoxy(fin,1);
            fin=fin-1;
            borra=1;
            corre_lcd();
            lcd_putc(' ');
        }
    }
    else {
        fin=fin+1;
        sale_borra();
        corre_lcd();
        lcd_putc('=');
    }
    while(input(PIN_B1)==0){
        retardo();
    }
    envia_caracter();
}
while(input(PIN_B2)==0){
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=1;
    DTMF='B';
    if (MAYUS==1) {
        lcd_putc('N');
    }
    else {
        if (alt==1){
            lcd_putc(',');
        }
    }
}

```

```

    else {
        lcd_putc('n');
    }
}
while(input(PIN_B2)==0){
    retardo();
}
envia_caracter();
}
while(input(PIN_B3)==0){
    fin=fin+1;
    sale_borra();
    corre_lcd();
    veces=1;
    DTMF='A';
    if (MAYUS==1) {
        lcd_putc('M');
    }
    else {
        if (alt==1){
            lcd_putc('.');
        }
        else {
            lcd_putc('m');
        }
    }
}
while(input(PIN_B3)==0){
    retardo();
}
envia_caracter();
}
while(input(PIN_B4)==0){
    while(input(PIN_B4)==0){
    }
}
while(input(PIN_B5)==0){ // tecla para deshabilitar la comunicación
CUELGA:
    cuelga(); // llama a rutina para colgar la llamada en curso
    while(input(PIN_B5)==0){
        retardo();
    }
}
return;
}
Void corre_lcd () { // si se llena la línea de transmisión de la LCD corre los caracteres
    if (fin==17){
        while (fin2<=15) {
            caracter[(fin2-1)]=lcd_getc((fin2+1),1);
            fin2=fin2+1;
        }
        lcd_gotoxy(1,1);
        printf(lcd_putc,"%s",caracter);
        lcd_gotoxy(16,1);
        fin=16;
        fin2=1;
    }
    if (borra==1) {
        lcd_gotoxy((fin+1),1);
    }
    else {
        lcd_gotoxy(fin,1);
    }
}
return;
}
void envia_caracter() { // envía tonos DTMF a través de la red celular
    if (ACTIVO==1){
        output_high(PIN_B7); // indica al usuario que el modulo está ocupado
    }
}

```

```

// se realiza esta rutina para la representación de caracteres mediante 2 tonos
if (veces==2){
repite_envio2:
    delay_ms(300);
    puts("AT+VTS=D");
    while(ocupado==1){ // espera a que se desocupe el modulo celular
        lcd_gotoxy(fin,1);
    }
    if (error==1){ // rutina para saber si el modulo contesta con error
        error=0;
        cont_error=cont_error+1;
        if (cont_error==3){ // si se dan 3 errores consecutivos, se finaliza la conexión
            cont_error=0;
            cuelga();
        }
        else {
            goto repite_envio2;
        }
    }
}
// rutina para los caracteres representados por un solo tono DTMF
repite_envio1:
    delay_ms(300);
    printf("AT+VTS=%c\r",DTMF);
    ocupado=1;
    while(ocupado==1){ // espera a que se desocupe el modulo celular
        lcd_gotoxy(fin,1);
    }
    if (error==1){
        error=0;
        cont_error=cont_error+1;
        if (cont_error==3){
            cont_error=0;
            cuelga();
        }
        else {
            goto repite_envio1;
        }
    }
    output_low(PIN_B7); // indica que el modulo esta libre para enviar otro carácter
}
return;
}
void sale_borra() { // rutina para saber cuándo se ha borrado un carácter en pantalla
    if (borra==1) {
        lcd_gotoxy(fin,1);
        borra=0;
    }
}
return;
}
Void corre_lcd_rx() { // si se llena la línea de recepción de la LCD corre los caracteres
    if (fin_rx==17) {
        while (fin2<=15) {
            caracter[(fin2-1)]=lcd_getc((fin2+1),2);
            fin2=fin2+1;
        }
        lcd_gotoxy(1,2);
        printf(lcd_putc,"%s",caracter);
        lcd_gotoxy(16,2);
        fin_rx=16;
        fin2=1;
    }
}
return;
}
void cuelga(){ // rutina para finalizar la comunicación
    delay_ms(100);
    putc('A');
}

```

```

putc('T');
putc('H');
putc('\r');
delay_ms(500);
lcd_putc("\f");
lcd_putc("Fin de LLamada");
delay_ms(2000);
putc('A');
putc('T');
putc('+');
putc('C');
putc('P');
putc('A');
putc('S');
putc('\r');
delay_ms(500);
ACTIVO=0;
MAYUS=0;
alt=0;
fin2=1;
fin=0;
fin_rx=1;
borra=0;
CONTESTA=0;
ocupado=0;
llamando=0;
lcd_putc("\f");
return;
}
// retardo utilizado para generar una secuencia de anti rebote para los
// pulsadores del teclado
void retardo() {
delay_ms(30);
return;
}
// fin del programa

```

Anexo D. Código en lenguaje C para el PIC16F628A

```
#include <16f628a.h> // se define el pic en el que se grabara el programa
#fuses XT, NOWDT, NOPROTECT, NOLVP, NOBROWNOUT
#use delay(clock=4000000)
#use rs232(baud=9600, xmit=PIN_B2, rcv=PIN_B1) // definición del RS232
// variables a utilizar en el programa
int tono=0; // se utiliza en la detección del dato que ingresa el mt8870
int dos_veces=0; // bandera para representar un carácter con uno o dos tonos
int MAYUS=0; // bandera para cambiar de mayúsculas a minúsculas
int alt=0; // bandera para cambiar a símbolos
int recepcion=0; // habilita la recepción de tonos
int cuenta=0; // dirección para almacenar los datos recibidos por RS232
int ruido=0; // bandera para detectar pérdida de señal
char ch=0; // variable para recibir el dato que ingresa por RS232
char llamada[15]=""; // cadena para almacenar los datos recibidos por RS232
char direccion[30]=""; // variable para almacenar la dirección del dispositivo Bluetooth
// rutina para interrupción por recepción RS232
#INT_RDA
void RDA_isr() {
    ch=getch(); // espera a que comience una trama de datos
    if (ch!=0x0A) { // si el carácter no es 0X0A significa que aún no ha terminado de recibir un dato completo
        llamada[cuenta]=ch; // almacena el dato en la dirección "cuenta" de la cadena "llamada"
        cuenta=cuenta+1;
    }
    else { // si el carácter es 0X0A entonces ha terminado de recibir los datos
        cuenta=0;
    }
    if (llamada[0]=='C' && llamada[1]=='M' && llamada[2]=='D') { // entra a modo configuración de bluetooth
        cuenta=0;
        if (input(PIN_B5)==1) { // si el bit está en 1, el modulo bluetooth conecta con la última dirección
            // guardada
            putc('z');
        }
        else { // de lo contrario obtiene la dirección de un nuevo dispositivo
            putc('~');
            while (ch!=0x0A) {
                ch=getch();
            }
            ch=0;
            while (ch!=0x0A) {
                ch=getch();
            }
            ch=0;
            while (ch!=0x0A) {
                ch=getch();
            }
            ch=0;
            gets(direccion);
            while (ch!=0x0A) {
                ch=getch();
            }
            ch=0;
            while (ch!=0x0A) {
                ch=getch();
            }
            ch=0;
            // envía la dirección al pic 16F876A para que este solicite conexión
            printf("%c%c%c%c%c%c%c%c%c%c%c%c", direccion[0], direccion[1], direccion[2], direccion[3], direccion[4], direccion[5], direccion[6], direccion[7], direccion[8], direccion[9], direccion[10], direccion[11]);
        }
        llamada[0]=0;
        llamada[1]=0;
        llamada[2]=0;
    }
}
```

```

}
// rutina de llamada entrante
if (llamada[0]=='R' && llamada[1]=='I' && llamada[2]=='N' && llamada[3]=='G') {
    putc('o');
    llamada[0]=0;
    llamada[1]=0;
    llamada[2]=0;
    llamada[3]=0;
}
// rutina para conocer si el teléfono está conectado de una llamada
if (llamada[0]=='+' && llamada[1]=='C' && llamada[2]=='P' && llamada[3]=='A' && llamada[4]=='S' && llamada[5]==':'
&& llamada[6]==' ' && llamada[7]=='3') {
    output_high(PIN_B6); // enciende led de la LCD
    recepcion=1; // habilita la recepción
    llamada[0]=0;
    llamada[1]=0;
    llamada[2]=0;
    llamada[3]=0;
    llamada[4]=0;
    llamada[5]=0;
    llamada[6]=0;
    llamada[7]=0;
}
// rutina para conocer si el teléfono está conectado de una llamada
if (llamada[0]=='+' && llamada[1]=='C' && llamada[2]=='P' && llamada[3]=='A' && llamada[4]=='S' && llamada[5]==':'
&& llamada[6]==' ' && llamada[7]=='4') {
    output_high(PIN_B6); // enciende led de la LCD
    recepcion=1; // habilita la recepción
    llamada[0]=0;
    llamada[1]=0;
    llamada[2]=0;
    llamada[3]=0;
    llamada[4]=0;
    llamada[5]=0;
    llamada[6]=0;
    llamada[7]=0;
}
// rutina para conocer si el teléfono esta desconectado de una llamada
if (llamada[0]=='+' && llamada[1]=='C' && llamada[2]=='P' && llamada[3]=='A' && llamada[4]=='S' && llamada[5]==':'
&& llamada[6]==' ' && llamada[7]=='0') {
    output_low(PIN_B6); // apaga led de la LCD
    MAYUS=0;
    alt=0;
    recepcion=0;
    llamada[0]=0;
    llamada[1]=0;
    llamada[2]=0;
    llamada[3]=0;
    llamada[4]=0;
    llamada[5]=0;
    llamada[6]=0;
    llamada[7]=0;
}
// cuando se recibe un ok por parte del teléfono, este informa al PIC 16F876A
if (llamada[0]=='O' && llamada[1]=='K') {
    putc('i');
    llamada[0]=0;
    llamada[1]=0;
}
// cuando se recibe un error por parte del teléfono, este informa al PIC 16F876A
if (llamada[0]=='E' && llamada[1]=='R' && llamada[2]=='R' && llamada[3]=='O' && llamada[4]=='R') {
    putc('l');
    llamada[0]=0;
    llamada[1]=0;
    llamada[2]=0;
    llamada[3]=0;
    llamada[4]=0;
}

```



```

    }
}
// Interrupción utilizada para reconocer perdida de señal y evitar recibir caracteres no deseados
#INT_TIMER0
void TIMER0_isr(void){
ruido=1;
}
// programa principal, recibe el dato del tono decodificado y lo envía al PIC16F876 para mostrarlos en la LCD
#use delay(clock=4000000)
void main() {
    //configuración inicial.
    delay_ms(100);
    output_low(PIN_B6); // apaga led de la LCD
    input_A() & 0b00000000;
    // habilita interrupciones
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_256);
    enable_interrupts(int_rda);
    enable_interrupts(GLOBAL);
    while (true) {
        if (repcion==1) {
            // si hay un tono disponible para decodificar el pin B0 se pone en alto, de lo contrario está en bajo
            if (input(PIN_B0)==1) {
                delay_ms(10); // esperamos que establezca el dato
                tono=input_A();
                tono=tono & 0b00001111; // toma los 4 bits menos significativos del puerto
                if (tono==0){ // si se recibe un tono DTMF "D"
                    dos_veces=1; // para caracteres que necesitan dos tonos para ser representados
                    while (input(PIN_B0)==1) { // espera hasta que el tono ya no esté presente
                    }
                }
            }
            else {
                if (dos_veces==0) { // envía al PIC16F876A los caracteres representados por un solo tono
                    switch (tono) {
                        case 1:
                            if (MAYUS==1) {
                                putc('A');
                            }
                            else {
                                if (alt==1) {
                                    putc('*');
                                }
                                else{
                                    putc('a');
                                }
                            }
                            break;
                        case 2:
                            if (MAYUS==1) {
                                putc('B');
                            }
                            else {
                                if (alt==1) {
                                    putc('!');
                                }
                                else{
                                    putc('b');
                                }
                            }
                            break;
                        case 3:
                            if (MAYUS==1) {
                                putc('C');
                            }
                            else {
                                if (alt==1) {
                                    putc('9');
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        else{
            putchar('c');
        }
    }
    break;
case 4:
    if (MAYUS==1) {
        putchar('D');
    }
    else {
        if (alt==1) {
            putchar('5');
        }
        else{
            putchar('d');
        }
    }
}
break;
case 5:
    if (MAYUS==1) {
        putchar('E');
    }
    else {
        if (alt==1) {
            putchar('2');
        }
        else{
            putchar('e');
        }
    }
}
break;
case 6:
    if (MAYUS==1) {
        putchar('F');
    }
    else {
        if (alt==1) {
            putchar('6');
        }
        else{
            putchar('f');
        }
    }
}
break;
case 7:
    if (MAYUS==1) {
        putchar('S');
    }
    else {
        if (alt==1) {
            putchar('4');
        }
        else{
            putchar('s');
        }
    }
}
break;
case 8:
    if (alt==1) {
        putchar('&');
    }
    else{
        putchar(' ');
    }
}
break;
case 9:
    if (MAYUS==1) {

```

```

    putchar('l');
}
else {
    if (alt==1) {
        putchar('-');
    }
    else{
        putchar('i');
    }
}
}
break;
case 10:
    if (alt==1) {
        putchar('=');
    }
    else{
        putchar('l');
    }
}
break;
case 11:
    if (MAYUS==1) {
        putchar('U');
    }
    else {
        if (alt==1) {
            putchar('_');
        }
        else{
            putchar('u');
        }
    }
}
break;
case 12:
    if (MAYUS==1) {
        putchar('L');
    }
    else {
        if (alt==1) {
            putchar('^');
        }
        else{
            putchar('l');
        }
    }
}
break;
case 13:
    if (MAYUS==1) {
        putchar('M');
    }
    else {
        if (alt==1) {
            putchar('.');
        }
        else{
            putchar('m');
        }
    }
}
break;
case 14:
    if (MAYUS==1) {
        putchar('N');
    }
    else {
        if (alt==1) {
            putchar(',');
        }
        else{

```

```

        putc('n');
    }
}
break;
case 15:
    if (MAYUS==1) {
        putc('O');
    }
    else {
        if (alt==1) {
            putc('+');
        }
        else{
            putc('o');
        }
    }
}
break;
}
}
else { // envía al PIC16F876A los caracteres representados por dos tonos
    dos_veces=0;
    switch (tono) {
    case 1:
        if (MAYUS==1) {
            putc('P');
        }
        else {
            if (alt==1) {
                putc('@');
            }
            else{
                putc('p');
            }
        }
    }
    break;
    case 2:
        if (MAYUS==1) {
            putc('Q');
        }
        else {
            if (alt==1) {
                putc('#');
            }
            else{
                putc('q');
            }
        }
    }
    break;
    case 3:
        if (MAYUS==1) {
            putc('R');
        }
        else {
            if (alt==1) {
                putc('3');
            }
            else{
                putc('r');
            }
        }
    }
    break;
    case 4:
        if (MAYUS==1) {
            putc('G');
        }
        else {
            if (alt==1) {

```

```

        putc('$');
    }
    else{
        putc('g');
    }
}
break;
case 5:
    if (MAYUS==1) {
        putc('T');
    }
    else {
        if (alt==1) {
            putc('(');
        }
        else{
            putc('t');
        }
    }
}
break;
case 6:
    if (MAYUS==1) {
        putc('K');
    }
    else {
        if (alt==1) {
            putc('%');
        }
        else{
            putc('k');
        }
    }
}
break;
case 7:
    if (MAYUS==1) {
        putc('V');
    }
    else {
        if (alt==1) {
            putc('?');
        }
        else{
            putc('v');
        }
    }
}
break;
case 8:
    if (MAYUS==1) {
        putc('W');
    }
    else {
        if (alt==1) {
            putc('1');
        }
        else{
            putc('w');
        }
    }
}
break;
case 9:
    if (MAYUS==1) {
        putc('X');
    }
    else {
        if (alt==1) {
            putc('8');
        }
    }
}

```

```

        else{
            putchar('x');
        }
    }
break;
case 10:
    if (MAYUS==1) {
        putchar('Y');
    }
    else {
        if (alt==1) {
            putchar('');
        }
        else{
            putchar('y');
        }
    }
break;
case 11:
    if (MAYUS==1) {
        putchar('Z');
    }
    else {
        if (alt==1) {
            putchar('7');
        }
        else{
            putchar('z');
        }
    }
break;
case 12:
    if (alt==0){
        if (MAYUS==0){
            MAYUS=1;
        }
        else {
            MAYUS=0;
        }
    }
    else {
        putchar('0');
    }
break;
case 13:
    if (alt==0){
        alt=1;
        MAYUS=0;
    }
    else {
        alt=0;
        MAYUS=0;
    }
break;
case 14:
    if (MAYUS==1) {
        putchar('J');
    }
    else {
        if (alt==1) {
            putchar(';');
        }
        else{
            putchar('j');
        }
    }
break;

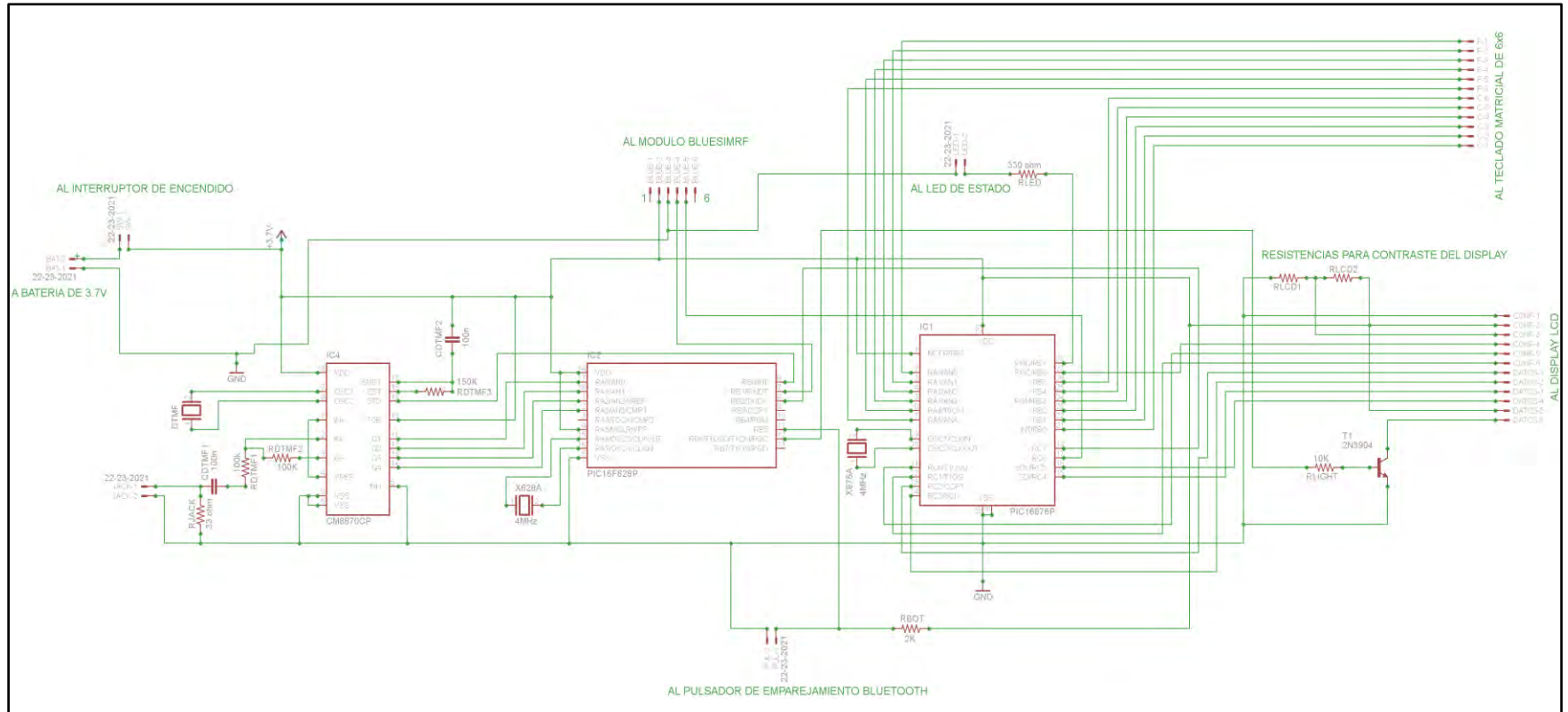
```

```

case 15:
  if (MAYUS==1) {
    putc('H');
  }
  else {
    if (alt==1) {
      putc(':');
    }
    else{
      putc('h');
    }
  }
  break;
}
}
senal2:
  while (input(PIN_B0)==1) { // espera mientras el tono este presente
  }
  // en caso de que el tono se pierda por mala señal se detecta mediante un temporizador para evitar
  // falsas lecturas de tonos
  ruido=0;
  //INICIA TIMER
  enable_interrupts(INT_TIMER0);
  set_timer0(30);
  // si el tono se pierde, pero antes de desbordar el timer vuelve a aparecer, entonces se considera
  // perdida de señal y el programa vuelve a esperar que finalice la duración del tono
  while (ruido==0) {
    if (input(PIN_B0)==1) {
      goto senal2;
    }
  }
}
}
}
}
// fin del programa

```

Anexo E. Circuito esquemático del dispositivo electrónico implementado



Anexo F. Formato de encuesta realizada a personas sordas



UNIVERSIDAD DE NARIÑO

Cuestionario anterior a la prueba de funcionamiento.

Estimado(a) voluntario(a), su opinión es muy importante para la realización y ejecución del proyecto “**Diseño e implementación de un dispositivo electrónico portátil para la comunicación en tiempo real, entre personas sordas a través de la red celular**” de la Facultad de Ingeniería del programa de Ingeniería Electrónica. A continuación se presentan una serie de aspectos relevantes para que por favor responda con la mayor sinceridad posible frente a cada aspecto.

Cuestionario.

1. De acuerdo a la explicación, ¿Qué opinión tiene usted acerca del dispositivo?

2. ¿considera que el uso del dispositivo representaría algún riesgo para su integridad física y mental?, ¿Cuál?

3. ¿Estaría usted dispuesto a realizar pruebas de funcionamiento con el dispositivo?

Gracias por su tiempo.



UNIVERSIDAD DE NARIÑO

Cuestionario posterior a la prueba de funcionamiento.

Estimado(a) voluntario(a), su opinión es muy importante para la realización y ejecución del proyecto “**Diseño e implementación de un dispositivo electrónico portátil para la comunicación en tiempo real, entre personas sordas a través de la red celular**” de la Facultad de Ingeniería del programa de Ingeniería Electrónica. A continuación se presentan una serie de aspectos relevantes para que por favor responda con la mayor sinceridad posible frente a cada aspecto.

Cuestionario.

1. ¿El uso del dispositivo afecta positiva o negativamente su forma de comunicarse a distancia?, ¿Por qué?

2. ¿Considera necesario algún cambio que ayude a mejorar su interacción con el dispositivo?

3. ¿Cree usted que el dispositivo ofrece una solución económica al problema de comunicación a distancia entre personas con discapacidad auditiva?, ¿Por qué?

Gracias por su tiempo.

Anexo G. Manual de usuario para el dispositivo electrónico.

El dispositivo electrónico permite la comunicación en tiempo real, entre personas sordas a través de la red celular mediante mensajes escritos que son transmitidos mientras este establecida una llamada de voz.

Sus partes principales se describen a continuación:



1. Pantalla LDC usada para visualizar el envío y recepción de los mensajes escritos.
2. Teclado alfanumérico tipo QWERTY usado para el envío de caracteres a través de la red celular.
3. Jack de 3.5mm macho usado para la recepción de mensajes que llegan a través de la red celular.
4. Interruptor de encendido/apagado para el dispositivo.
5. Led de estado que indica cuando el dispositivo se encuentra ocupado realizando alguna acción en específico.
6. Pulsador de sincronización usado para sincronizar el dispositivo con el teléfono por primera vez.

Para hacer uso del dispositivo se debe contar con un teléfono celular de gama media con tecnología Bluetooth compatible con el protocolo RFCOMM y perfil SPP que además soporte control por medio de comandos AT.

Además el teléfono debe contar con una salida de audio mediante Jack hembra de 3.5mm para que se conecte con el dispositivo.

Una vez activado el dispositivo Bluetooth del teléfono celular, se deben realizar los siguientes procesos para establecer la comunicación.

- **Sincronizando y conectando un teléfono con el dispositivo.** Para sincronizar el teléfono por primera vez con el dispositivo electrónico se mantiene presionado el pulsador de sincronización ubicado en la parte superior del dispositivo mientras se enciende el dispositivo mediante su interruptor de encendido. Así, el dispositivo enciende el led de estado durante 20 segundos para indicar que ha comenzado la búsqueda del nuevo teléfono y una vez encontrado, se debe permitir la conexión en el teléfono para completar la sincronización.

Después de sincronizar los equipos por primera vez, para posteriores conexiones basta con encender el dispositivo Bluetooth del teléfono celular y luego encender el dispositivo electrónico mediante su interruptor para realizar la conexión. Este proceso tarda aproximadamente 8 segundos durante los cuales el led de estado permanece encendido.

- **Estableciendo una llamada.** Se debe marcar el número celular desde el teléfono, luego se espera a que el teléfono indique que han contestado la llamada y entonces se presiona el botón verde ● del dispositivo para encender la pantalla y habilitar la comunicación. Cabe aclarar que el auricular del teléfono debe estar conectado al dispositivo.

- **Contestando una llamada.** Cuando el teléfono celular vibre para indicar una llamada entrante, se debe presionar el botón verde ● del dispositivo para contestar la llamada y a su vez, encender la pantalla LCD y habilitar la comunicación.

- **Comunicándose con el dispositivo.** Cuando se presiona alguna tecla del dispositivo, se enciende el LED de estado (Azul), indicando que el módulo se encuentra ocupado y no se puede presionar otra tecla mientras permanezca encendido. Una vez el LED se apague se puede seguir presionando las teclas.

La tecla “**Alt**” se utiliza para intercambiar entre caracteres alfanuméricos y símbolos especiales.

La tecla Flecha arriba ↑ se utiliza para intercambiar caracteres en mayúsculas o minúsculas.

- **Finalizando una llamada.** Para finalizar una llamada activa se debe presionar el botón rojo ● del dispositivo e inmediatamente se muestra el mensaje “Fin de Llamada” y la pantalla se apaga indicando que se ha finalizado la comunicación.

Nota: es necesario que el usuario mantenga el dispositivo encendido mientras lo porta, para permitir una comunicación oportuna. Así mismo el dispositivo debe encontrarse dentro de un rango menor a 18 metros para mantener su conexión con el teléfono celular.

En caso de perder la conexión por salir del rango de alcance, se debe reiniciar el dispositivo para restablecer la conexión.