

DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA ELECTRÓNICO DE
INFORMACIÓN AL USUARIO PARA AUTOBUSES DE LA UNIÓN TEMPORAL
CIUDAD SORPRESA

MARCIA GÓMEZ ORTEGA

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
PASTO
2012

DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA ELECTRÓNICO DE
INFORMACIÓN AL USUARIO PARA AUTOBUSES DE LA UNIÓN TEMPORAL
CIUDAD SORPRESA

MARCIA GÓMEZ ORTEGA

Trabajo de grado presentado como requisito parcial para optar por el título de
Ingeniera Electrónica

DIRECTOR
DARÍO FERNANDO FAJARDO FAJARDO
MAGÍSTER EN AUTOMATIZACIÓN Y CONTROL

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
PASTO
2012

“Las ideas y conclusiones aportadas en la tesis de grado, son responsabilidad exclusiva de los autores.”

Artículo 1°. Acuerdo No. 324 de Octubre 11 de 1966, emanado del Honorable Consejo Directivo de la Universidad de Nariño

Nota de aceptación:

Firma del Presidente del Jurado

Firma del Jurado

Firma del Jurado

Pasto, Noviembre de 2012.

DEDICATORIA

A Dios, porque su fuerza y energía han sido la luz constante en mi camino.

A mis padres, Pedro y María por su amor incondicional y su abnegado apoyo.

A mis hermanos, Margarita, Pablo, Maribel y Rosa por ser mis compañeros y amigos.

A mi esposo, Oscar, por todo su amor.

A mi querida Laura, por ser el motivo de mi superación diaria.

Al amigo por siempre, Ing. José Dolores Rodríguez q.e.p.d por todos sus consejos.

Especial agradecimiento a los docentes del departamento de Ingeniería Electrónica de la Universidad de Nariño especialmente al Ing. Darío Fajardo Fajardo por su colaboración.

Marcia Gómez Ortega

RESUMEN

Teniendo en cuenta los retos de la contemporaneidad y dando solución a las necesidades que en nuestra región surgen con la ejecución del Sistema Estratégico de Transporte (SETP) y, acorde a los lineamientos nacionales contemplados en el Plan Nacional de Desarrollo con la estrategia “Ciudades Amables”, se desarrolla el presente trabajo con la finalidad de implementar una herramienta que promueva el mejoramiento en la prestación del servicio de transporte público colectivo.

Se trata de un sistema de señalización para los autobuses de la Unión Temporal de Pasto, aplicable a otros sistemas estratégicos de ciudades intermedias, el cual tiene correspondencia con los objetivos del SETP.

El sistema de información al usuario consta de tres partes funcionales: se dispone de una unidad de control desarrollada en torno a microcontroladores PIC, para gestionar las operaciones de voz y datos del sistema y el posicionamiento global a través de un receptor GPS, igualmente, de tres paneles electrónicos de matriz de leds de alta luminosidad, los cuales muestran de manera visual la información concerniente a próxima parada e itinerarios de ruta y finalmente la interfaz de usuario que corresponde a la unidad de mando del conductor.

Palabras claves: amplificador, archivos WAV, GPS, matriz de leds, microcontroladores PIC, posicionamiento global, señalización visual y audible, sonido digital, SETP, panel electrónico.

ABSTRACT

This work is aimed at implementing an encouraging tool to improve the provision of public transport service considering the current challenges and sorting out our region necessities when implementing “Sistema Estratégico de Transporte (SETP)” as well as the national guidelines stated in “Plan Nacional de Desarrollo” with the strategy “Ciudades Amables”.

This idea refers to a signaling system for the “Unión Temporal de Pasto” buses, applicable to other strategic systems in medium cities, in agreement on the SETP objectives.

The user information system consists of three functional parts: a control unity developed around a PIC micro-controller to arrange the voice procedures, system data, and global position through both a GPS receiver and three Leds Matrix electronic panels, high brightness, which shows, in a visual, way the information about the following stop and the itinerary. Finally, the users’ interface corresponding to the driver unity control.

Key words: amplifier, WAV files, GPS, array of LEDs, PIC microcontrollers, global positioning, visual and audible signaling, digital sound, SETP, electronic panel.

CONTENIDO

	pág.
INTRODUCCIÓN	18
1. OBJETIVOS	20
1.1 OBJETIVO GENERAL	20
1.2 OBJETIVOS ESPECÍFICOS	20
2. JUSTIFICACIÓN	21
3. MARCO CONCEPTUAL	23
3.1 MICROCONTROLADORES PIC	23
3.2 SISTEMA DE INFORMACIÓN AL USUARIO	24
3.2.1 SEÑALÉTICA DE LA FLOTA	25
3.3 INFORMACION VISUAL	27
3.3.1 PANEL ELECTRÓNICO DE INFORMACIÓN AL USUARIO	27
3.3.2 CARACTERÍSTICAS MATRIZ LEDS	27
3.3.2.1 Resolución	27
3.3.2.2 Diámetro del punto.....	27
3.3.2.3 Visibilidad.....	27
3.3.2.4 Brillo	28
3.3.2.5 Ángulo de Visualización	28
3.3.2.6 Caracteres.	28
3.3.2.7 Rango de lectura.....	28

3.3.3 TÉCNICA DE MULTIPLEXACIÓN	28
3.4 INFORMACIÓN AUDIBLE	29
3.4.1 CODIFICACIÓN PCM	29
3.4.1.1 Muestreo	30
3.4.1.2 Cuantificación	31
3.4.1.3 Codificación	32
3.4.2 FORMATOS DE ARCHIVOS DE AUDIO	33
3.4.2.1 Formato de archivo Wave	34
3.4.3 CONVERSIÓN DIGITAL ANALÓGICA	38
3.4.3.1 Características de los conversores D/A	40
3.4.4 ESTÁNDAR SD	40
3.4.4.1 Interfaz	41
3.4.4.2 Bus en el modo SPI	43
3.4.4.3 Comandos.....	44
3.4.4.4 Formato de los bloques de datos.....	47
3.4.4.5 Respuestas	48
3.4.5 SISTEMA DE POSICIONAMIENTO GLOBAL	49
3.4.5.1 Sistema satelital.....	49
3.4.5.2 Sistema de control terrestre	50
3.4.5.3 Sistema de usuario	51
3.4.5.4 Funcionamiento del sistema GPS.....	51
3.4.5.5 Estándar NMEA 0183	55
4. DESARROLLO	58
4.1 DESCRIPCIÓN GENERAL	58
4.1.1 SEÑALIZACIÓN VISUAL	58

4.1.1.1 Panel electrónico de matriz de leds	58
4.1.1.2 Canal de datos	61
4.1.1.3 Técnica de multiplexación.....	62
4.1.1.4 Rutina de visualización	63
4.1.1.5 Rutina memoria eeprom.....	65
4.1.2 SEÑALIZACIÓN ACÚSTICA	68
4.1.2.1 Bloque funcional tarjeta de memoria SD.....	70
4.1.2.2 Bloque funcional GPS.....	77
4.1.2.3 Bloque funcional conversor D/A.....	84
5. RESULTADOS	93
5.1 HARDWARE	93
5.1.1 MÓDULO DE SEÑALIZACIÓN VISUAL	94
5.1.2 MÓDULO DE SEÑALIZACIÓN ACÚSTICA	96
5.1.3 INTERFAZ DEL CONDUCTOR	99
5.2 CONEXIÓN AL PC	100
5.3 SOFTWARE	1002
5.3.1 SOFTWARE EDITOR MATRIZ DE LEDS	102
5.3.2 SOFTWARE INTERFAZ SISTEMA GPS	105
6. CONCLUSIONES	108
7. RECOMENDACIONES	110
8. BIBLIOGRAFIA	111

LISTA DE TABLAS

	pág.
TABLA 1. ARQUITECTURA DEL ARCHIVO WAVE EN SU FORMA CANÓNICA	35
TABLA 2. VALORES DEL SEGMENTO RIFF PARA UN ARCHIVO WAVE	36
TABLA 3. FORMATO DE VALORES DEL SEGMENTO WAVE	36
TABLA 4. TIPOS DE COMPRESIÓN DEL FORMATO WAVE	37
TABLA 5. FORMATO DE VALORES DEL SEGMENTO DATOS	38
TABLA 6. ASIGNACIÓN DE PINES EN UNA TARJETA SD MODOS SD Y SPI	42
TABLA 7. ESTRUCTURA DE UN COMANDO DEL ESTÁNDAR SD	45
TABLA 8. COMANDOS DEL PROTOCOLO DE ACCESO MEDIANTE SPI	45
TABLA 9. FORMATO DE LA TRAMA DE DATOS	48
TABLA 10. DESCRIPCIÓN DE LA TRAMA DE DATOS	48
TABLA 11. TIPOS DE RESPUESTA EN EL MODO SPI	49
TABLA 12. FORMATO DE LA SENTENCIA GGA	57
TABLA 13. TABLA DE VERDAD DEL REGISTRO CMOS 74374	59
TABLA 14. BYTES DE INFORMACIÓN DE UN ARCHIVO DE AUDIO	70
TABLA 15. FORMATO DE UN PAQUETE DE DATOS	74
TABLA 16. CARACTERÍSTICAS GENERALES DEL MÓDULO GPS	79
TABLA 17. CONFIGURACIÓN MÍNIMA DEL GPS A1029-C	79
TABLA 18. VALORES REALES DE UNA TRAMA GPGGA	81
TABLA 19. VALORES ASCII EXTRAIDOS DE UNA TRAMA GPGGA	82
TABLA 20. VARIABLES DEL ÍNDICE DE UN ARCHIVO DE AUDIO	84
TABLA 21. ESTRUCTURA REGISTRO DE ENTRADA DEL MCP4821	86

LISTA DE FIGURAS

	pág.
FIGURA 1. ARQUITECTURA SISTEMA DE INFORMACIÓN AL VIAJERO	26
FIGURA 2. DIAGRAMA DE BLOQUES CONVERTOR ANALÓGICO DIGITAL	30
FIGURA 3. MUESTRAS TOMADAS DE UNA SEÑAL ANALÓGICA	30
FIGURA 4. ASIGNACIÓN DE VALORES ANALÓGICOS A LAS MUESTRAS	31
FIGURA 5. DIFERENCIA DE RESOLUCIÓN DE UNA SEÑAL MUESTREADA	32
FIGURA 6. SALIDA DE UN CONVERTIDOR A/D EN LA CODIFICACIÓN	33
FIGURA 7. RECUPERACIÓN DE LA SEÑAL POR CONVERSIÓN D/A	39
FIGURA 8. ARQUITECTURA INTERNA DE LA TARJETA SD	41
FIGURA 9. DIAGRAMA DE PINES DE UNA TARJETA SD	42
FIGURA 10. CONEXIÓN DE LA TARJETA DE MEMORIA SD EN EL MODO SPI	44
FIGURA 11. LECTURA DE MÚLTIPLES BLOQUES	46
FIGURA 12. ESCRITURA DE MÚLTIPLES BLOQUES	47
FIGURA 13. BITS DE ESTADO DE RESPUESTA	48
FIGURA 14. CONSTELACIÓN DE SATELITES NAVSTAR	50
FIGURA 15. CÓDIGO PSEUDOALEATORIO DE TIEMPO GPS	52
FIGURA 16. SISTEMA DE COORDENADAS GPS	53
FIGURA 17. UBICACIÓN ESTACIÓN RECEPTORA RESPECTO 1 SATÉLITE	54
FIGURA 18. UBICACIÓN ESTACIÓN RECEPTORA RESPECTO 3 SATÉLITES	55
FIGURA 19. DIAGRAMA DE BLOQUES SISTEMA DE INFORMACIÓN AL USUARIO	58
FIGURA 20. DIAGRAMA DE BLOQUES CIRCUITO DE CONTROL MATRIZ	58

FIGURA 21. DIAGRAMA ESQUEMÁTICO DEL CIRCUITO DE CONTROL MATRIZ DE LEDS	60
FIGURA 22. CANAL PARA LA TRANSFERENCIA DE DATOS	61
FIGURA 23. DIAGRAMA CIRCUITAL Y ESQUEMÁTICO DE UNA MATRIZ INDIVIDUAL 8X8	62
FIGURA 24. VISUALIZACIÓN DE UN CARACTER EN UNA MATRIZ 8X8	63
FIGURA 25. DIAGRAMA DE FLUJO UTILIZADO PARA LA VISUALIZACIÓN	63
FIGURA 26. DIAGRAMA DE FLUJO DE LA SUBROUTINA CLOCK_LATCH	64
FIGURA 27. DIAGRAMA DE FLUJO DE LA SUBROUTINA ACTIVA_COLUMNNA	65
FIGURA 28. PATILLAJE DE LA MEMORIA SERIE 24LC256	66
FIGURA 29. DIAGRAMA DE FLUJO LECTURA DE ITINERARIOS DE RUTA	67
FIGURA 30. DIAGRAMA DE BLOQUES DEL COMPONENTE DE SEÑALIZACIÓN ACÚSTICA	68
FIGURA 31. DIAGRAMA DE PINES ENTRADAS Y SALIDAS DEL PIC18F2550	68
FIGURA 32. DIAGRAMA DE PINES DEL CONECTOR SD	70
FIGURA 33. DIAGRAMA DE PINES DE LA MEMORIA RAM	72
FIGURA 34. SECUENCIA DE INICIALIZACIÓN DE LA TARJETA SD	73
FIGURA 35. SECUENCIA DE INICIALIZACIÓN COMPLETA	74
FIGURA 36. TRANSFERENCIA DE DATOS DURANTE LA LECTURA DE BLOQUES	75
FIGURA 37. TRANSFERENCIA DE DATOS DURANTE LA ESCRITURA	76
FIGURA 38. MÓDULO RECEPTOR GPS A1029-C DE TYCO ELECTRONICS	78
FIGURA 39. FORMA DE ONDA DE LA SEÑAL 1PPS	78
FIGURA 40. PINES ENTRADA-SALIDA DEL MÓDULO RECEPTOR GPS	79
FIGURA 41. ANTENA ACTIVA DEL RECEPTOR GPS	80
FIGURA 42. CONEXIÓN DEL MÓDULO GPS AL PIC18F2550	80

FIGURA 43. MÁQUINA DE ESTADO PARA DISCRIMINACIÓN DE LA SENTENCIA GPGGA	82
FIGURA 44. BLOQUE FUNCIONAL ETAPA DE AMPLIFICACIÓN	85
FIGURA 45. DIAGRAMA DE PINES Y CONEXIONES DEL DAC MCP4821	85
FIGURA 46. COMANDOS DE ESCRITURA DEL CONVERTOR D/A	87
FIGURA 47. CIRCUITO DE AMPLIFICACIÓN CON GANANCIA DE 46DB	88
FIGURA 48. COMUNICACIÓN ENTRE LOS MÓDULOS DEL SISTEMA	89
FIGURA 49. FUNCIÓN DE ATENCIÓN DEL PUERTO SERIE RS232	90
FIGURA 50. SISTEMA DE INFORMACIÓN AL USUARIO	93
FIGURA 51. DIAGRAMA DEL MÓDULO DE SEÑALIZACIÓN VISUAL	94
FIGURA 52. INSTALACIÓN PANEL FRONTAL	95
FIGURA 52A. INSTALACIÓN PANEL POSTERIOR	95
FIGURA 52B. INSTALACIÓN PANEL INTERIOR	95
FIGURA 53. CIRCUITO CONMUTADO PARA LA CONVERSIÓN DE VOLTAJE	96
FIGURA 54. SECCIONES DE TARJETA ELECTRÓNICA DEL MÓDULO DE SEÑALIZACIÓN ACÚSTICA P.1	97
FIGURA 55. SECCIONES DE TARJETA ELECTRÓNICA DEL MÓDULO DE SEÑALIZACIÓN ACÚSTICA P.2	98
FIGURA 56. MODOS DE OPERACIÓN DEL SISTEMA	99
FIGURA 57. INTERFAZ CONDUCTOR	100
FIGURA 58. VENTANA PRINCIPAL EDITOR MATRIZ DE LED	103
FIGURA 59. VENTANA CREAR-MODIFICAR FUENTES DE TEXTO	104
FIGURA 60. VENTANA PRINCIPAL DEL SOFTWARE SISTEMA GPS	105
FIGURA 61. VENTANA PROPIEDADES DEL ARCHIVO WAV	106
FIGURA 62. VENTANA COORDENADAS ACTUALES	107
FIGURA 63. VENTANA ENVIAR WAV/COORDENADAS	107

GLOSARIO

A/D: Conversión de una señal analógica a digital.

ALTAVOZ: Transductor electro-acústico que convierte una señal eléctrica en sonido, utilizado para reproducir sonido desde un dispositivo electrónico.

COORDENADAS GEOGRÁFICAS: Es un sistema de referencia que utiliza las dos coordenadas angulares, latitud (Norte y Sur) y longitud (Este y Oeste) para determinar los ángulos laterales de la superficie terrestre o en general de un círculo o un esferoide. Se expresa en grados sexagesimales.

D/A: Conversión de una señal digital a analógica.

GGA (*Global Positioning System Fix Data*): Sentencia que incluye datos de tiempo, latitud, altitud y datos del estado del receptor.

GPS (*Global Positioning System*): El sistema de posicionamiento global es un sistema compuesto por una red de satélites, radio bases terrestres y receptores GPS, que permite casi todas las posibilidades de navegación y posicionamiento en cualquier parte del mundo. En la ubicación un punto se utiliza como mínimo cuatro satélites, el dispositivo GPS recibe las señales de cada uno de ellos y mediante triangulación calcula la posición.

I2C (*Inter Integrated Circuit*): Es un bus de comunicaciones en serie donde la transmisión de la información se realiza mediante dos líneas, una para los datos y por otra la señal de reloj.

LITTLE ENDIAN: designa el formato en el que se almacenan los datos de más de un byte en un ordenador. Se almacena de manera inversa, es decir, primero los bit menos significativos LSB seguidos de los mas significativos MSB.

LSB (*Less Significant Bit*): El bit menos significativo, es la posición de bit en un número binario que tiene el menor valor.

MMC (*Multi Media Card*): Es un estándar de tarjeta de memoria. Prácticamente igual a la SD, carece de la pestaña de seguridad que evita sobrescribir la información grabada en ella.

MSB (*Most Significant Bit*), El bit más significativo, es la posición de bit en número binario que tiene el mayor valor.

MUESTREO: proceso mediante el cual se representa una señal continua por medio de valores discretos de la misma (es decir en intervalos de tiempo), llamados muestras.

NMEA: Es una especificación eléctrica y de datos entre dispositivos electrónicos marinos y entre receptores GPS. El protocolo NMEA 0183 es un medio a través del cual los instrumentos marítimos y también la mayoría de los receptores GPS pueden comunicarse los unos con los otros. Ha sido definido, y está controlado, por la organización estadounidense National Marine Electronics Association.

PCM (*Pulse Code Modulation*): Modulación por codificación de impulsos, es un procedimiento de modulación utilizado para transformar una señal analógica en una secuencia de bits, es decir en una señal digital. Una trama o stream PCM es una representación digital de una señal analógica en donde la magnitud de la onda analógica es tomada en intervalos uniformes, cada muestra puede tomar un conjunto finito de valores, los cuales se encuentran codificados.

RIFF (*Resource Interchange File Format*): formato de archivo informático para intercambio de recursos, predeterminado de los archivos multimedia de Windows. Los enteros de más de un byte se representan en el sistema Little Endian. La implementación de Microsoft es conocida principalmente por los formatos AVI, ANI y WAV, que emplean el formato RIFF como base.

SD (*Secure Digital*): Es un formato de tarjeta de memoria flash desarrollado por Matsushita, SanDisk y Toshiba (luego asociadas en la SD Card Association) para ser usado en dispositivos portátiles.

SEÑALÉTICA: Técnica que permite optimizar, en un sistema de codificación, las señales y los signos destinados a orientar y facilitar a las personas respecto a su ubicación y a sus acciones en los distintos espacios y ambientes.

SLEW RATE: Es un efecto no lineal en los amplificadores, se define como la máxima pendiente de la salida ante una conmutación del código de entrada. También se define como la velocidad de cambio del voltaje de salida con respecto a las variaciones en el voltaje de entrada

SPI (*Serial Peripheral Interface*): es un bus de tres líneas, sobre el cual se transmiten paquetes de información de 8 bits. Cada una de estas tres líneas porta la información entre los diferentes dispositivos conectados al bus. Cada dispositivo conectado al bus puede actuar como transmisor y receptor al mismo tiempo, por lo que este tipo de comunicación serial es full duplex. Dos de estas líneas transfieren los datos (una en cada dirección) y la tercer línea es la del reloj.

USART (*Universal Synchronous/Asynchronous Receiver Transmitter*): periférico para la transmisión de datos en formato serie, usa técnicas de transmisión síncrona o asíncrona, según se configure el periférico.

USB (*Universal Serial Bus*): El Bus de Serie Universal es una interfaz que provee un estándar de bus serie para conectar dispositivos a un ordenador personal (generalmente a un PC). Un sistema USB tiene un diseño asimétrico, que consiste en un solo servidor y múltiples dispositivos conectados en una estructura de árbol utilizando concentradores especiales.

UTC (*Universal Time Coordinated*): Tiempo universal coordinado. Es el tiempo de la zona horaria de referencia respecto a la cual son calculadas todas las otras zonas del mundo. El 1 de enero de 1972 pasó a ser el sucesor del GMT, tiempo promedio del Observatorio de Greenwich, en Londres. La nueva denominación fue acuñada para eliminar la inclusión de una ubicación específica en un estándar internacional, así como para basar la medida del tiempo en los estándares atómicos, más que en los celestes.

UTM (*Universal Transverse Mercator*): El sistema de Coordenadas Universal Transversal de Mercator, está basado en la proyección cartográfica transversa de Mercator y, a diferencia del sistema de coordenadas geográficas, se expresa en metros. Actualmente se usa el elipsoide WGS84 como modelo base para este sistema.

WAV: Formato de audio digital normalmente sin compresión de datos desarrollado y propiedad de Microsoft e IBM, utilizado para almacenar archivos de audio en un PC, admite archivos mono y estéreo a diversas resoluciones y velocidades de muestreo, su extensión es *.wav. A pesar de que el formato WAV es compatible con casi cualquier códec de audio, se utiliza principalmente con el formato PCM (no comprimido) y, al no tener pérdida de calidad, es adecuado para uso profesional.

INTRODUCCIÓN

Los procesos de urbanización que el país ha experimentado durante su historia reciente han exigido el desarrollo de nuevas capacidades en la prestación del servicio de transporte público para su adecuado funcionamiento. A medida que las ciudades crecen en población y en superficie urbanizada, el número de viajes motorizados aumenta, requiriendo sistemas de transporte eficientes. Sin embargo, en las últimas décadas, ha surgido una fuerte tendencia al deterioro en la calidad de este servicio.

Dentro de las dificultades que experimenta, cabe resaltar, que parte importante de la malla vial y los elementos de señalización de las ciudades no son los apropiados o se encuentran en estado deficiente. En este sentido, las políticas nacionales han centrado sus esfuerzos en propender el establecimiento de sistemas de transporte eficientes que atiendan dichas necesidades.

Es así como, en la ciudad de Pasto, se viene adoptando las medidas de reestructuración planteadas, con la implementación del Sistema Estratégico de Transporte, cuyo proceso pre-operativo, hasta la fecha, se encuentra en período de prueba. La aplicación de estas medidas implica, entre otros factores, la modernización del sistema semafórico, de la señalización e infraestructura vial y de transporte y la localización y operación de paraderos con criterios técnicos, mediante la creación de paradas fijas con equipamiento y señalización adecuados.¹

De esta forma, surge el presente trabajo con el fin de desarrollar un sistema de información para los usuarios del transporte público, destinado a mejorar la calidad del servicio y facilitar los desplazamientos dentro del sistema. Como objetivo principal se plantea proveer una solución tecnológica acorde a los requerimientos, contemplados en el decreto 0735 de 2009, teniendo en cuenta que su realización puede ser desarrollada con recursos y mano de obra regional desde la parte inicial de diseño hasta su posterior instalación y mantenimiento.

En este sentido, se diseña un sistema de información que proporciona la señalización visual y acústica requerida por un usuario de transporte público, antes y después del ingreso al autobús, permitiendo dar lectura del número de ruta y destino, en cualquier hora del día o de la noche durante la operación del automotor. Puesto que el objetivo principal del sistema consiste en brindar información oportuna durante el recorrido, se convierte en una herramienta especialmente útil para personas con discapacidad auditiva o visual, ya que actúa como un asistente de viaje.

¹ COLOMBIA. MINISTERIO DE TRANSPORTE. Conpes 3167. Política para mejorar el servicio de transporte público urbano de pasajeros. Bogotá. 2002. 50p.

Se procede entonces a construir un prototipo equipado con los siguientes elementos: un primer componente de señalización visual que consta de tres paneles electrónicos de matriz de leds, dos de visualización externa para información sobre la ruta del autobús y los itinerarios de ruta y, uno de visualización interna para información sobre la próxima parada; un segundo componente que proporciona la señalización audible, el cual complementa la información visual disponible en el interior del autobús con el anuncio de la próxima parada; un tercer componente diseñado para el conductor del autobús, que consta de una pantalla LCD y teclado, el cual permite elegir la ruta que se pondrá en operación.

Finalmente, el sistema de información cuenta con una aplicación software para la edición de texto y el envío de coordenadas correspondientes a los paraderos; el texto es creado en un PC bajo entorno Windows y tanto los mensajes como las coordenadas que registra un receptor GPS son enviados al sistema de información vía USB.

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Diseñar y construir un sistema electrónico que permita al usuario del transporte público colectivo un acceso confiable y oportuno a la información concerniente a la ruta que aborda.

1.2 OBJETIVOS ESPECÍFICOS

Suministrar al usuario del transporte público, antes de su ingreso al autobús, la información necesaria que le permita como mínimo identificar ruta, origen y destino, mediante paneles electrónicos de visualización de mensajes externos.

Proporcionar a los usuarios dentro del bus el acceso a la información de manera visual y audible mediante un panel electrónico de visualización de mensajes.

Diseñar el sistema electrónico capaz de almacenar datos, gestionar información y controlar de forma eficiente la herramienta de información.

Articular una aplicación software de edición de texto, que permita de una manera fácil y dinámica, la edición de los mensajes que se van a proyectar sobre los tableros electrónicos.

2. JUSTIFICACIÓN

El municipio de Pasto es una de las ciudades intermedias más importantes del sur del país, con una población de 312.377 habitantes en la zona urbana y 70.241 en la zona rural,² es paso obligado a la República del Ecuador, a la región del alto Putumayo y a la costa sur de Colombia, lo que lo constituye en un polo de desarrollo en la región sur occidental de Colombia.

El continuo y acelerado crecimiento de la ciudad, la motorización y la expansión del área urbana le generan crecientes desafíos al sistema de transporte, al punto que las soluciones que se adopten al respecto pueden condicionar su competitividad futura en un sistema económico crecientemente globalizado; de persistir la situación actual del transporte público colectivo de pasajeros, sus deficiencias seguirán limitando los niveles de productividad urbana y la calidad de vida de sus habitantes.

Por tal razón, desde hace varios años la Administración Municipal, entes privados y el Gobierno Nacional han venido adelantado el plan de movilidad y el Sistema Estratégico de Transporte Público colectivo, SETP, para la ciudad de Pasto, el cual comenzó a operar en el año 2010. El proceso pre-operativo que está en prueba, es una fase de aprendizaje y de implementación de infraestructura y ampliación de andenes y vías. En el componente de infraestructura se estableció 76 paraderos, diseñados para cada una de las rutas y son los únicos sitios autorizados para el ascenso y descenso de pasajeros. Cabe decir que las zonas de parada y abordaje no estaban delimitadas y su implementación es un logro importante del SETP que da consecuencia a los objetivos del proyecto que buscan proporcionar a los usuarios dentro del bus, información acerca del próximo destino.

En este sentido, se evidencia la necesidad de modernizar el transporte público de pasajeros mediante la utilización de herramientas automatizadas de gestión de la información, para mejorar la calidad de los servicios ofrecidos a los usuarios; el sistema de información al usuario, que corresponde a uno de los objetivos del sistema de programación, control y gestión, cuarto componente del SETP,³ permitirá gestionar la señalética del servicio de transporte y la socialización del SETP en la ciudad, en busca que los mismos usuarios sean multiplicadores, en sus comunidades, del cumplimiento de las normas de tránsito, y el desarrollo de procesos de corresponsabilidad entre la ciudadanía y la Administración municipal en relación con la seguridad vial.

² COLOMBIA. DEPARTAMENTO ADMINISTRATIVO NACIONAL DE ESTADÍSTICA. Proyecciones de población municipales por área 2005-2020. Bogotá, 2005

³ COLOMBIA. MINISTERIO DE TRANSPORTE. Conpes 3167. Política para mejorar el servicio de transporte público urbano de pasajeros. Bogotá, 2002. 50p.

La señalización audible, por otro lado, se convierte en una estrategia de difusión y cultura ciudadana para la movilidad, mediante la cual se fomenta no sólo el uso adecuado del sistema, sino que también brinda apoyo a los usuarios con necesidades diferentes, como es el caso de personas con capacidades reducidas que tienen mayor requerimiento de información y orientación en su recorrido.

3. MARCO CONCEPTUAL

3.1 MICROCONTROLADORES⁴

Un microcontrolador es un circuito integrado programable capaz de ejecutar las órdenes grabadas en su memoria, y cuya arquitectura interna se asemeja a la de un microcomputador. Se compone de CPU, memoria RAM y ROM y de varios bloques funcionales destinados a tareas específicas, tales como los circuitos de entrada y salida. Entre los microcontroladores más extendidos se puede mencionar los fabricados por Motorola, Siemens, Microchip e Intel.

En cuanto los microcontroladores de Microchip, éstos se dividen en PICs de 8 y 16 bits. Los de 8 bits se clasifican en tres gamas a saber, base, media y alta; se caracterizan porque emplean un juego de instrucciones tipo RISC cuyo número varía de 33 para PICs de gama base hasta 77 para PICs de gama alta. Estos dispositivos poseen una amplia gama de avances hardware incorporados, tales como memoria flash y ROM disponible desde 256 bytes a 256 kilobytes, periféricos serie síncronos y asíncronos, conversores analógicos y digitales de 8, 10 y 12 bits, comparadores de tensión, módulos de captura y comparación PWM, periférico MSSP (Master Synchronous Serial Port) para comunicaciones IIC y SPI, memoria EEPROM interna con duración de hasta un millón de ciclos de lectura/escritura, soporte de interfaz USB, controlador Ethernet, controlador CAN y controlador LIN, entre otros.

Por otra parte, en la programación de microcontroladores, la utilización de los lenguajes más cercanos a la máquina (de bajo nivel) representan un considerable ahorro de código en la edición de los programas, lo que es muy importante dada la estricta limitación de la capacidad de la memoria de instrucciones. Los programas bien realizados en lenguaje ensamblador optimizan el tamaño de la memoria que ocupan y su ejecución es muy rápida. Sin embargo, los lenguajes de alto nivel tienen instrucciones más potentes y se puede realizar funciones y operaciones mucho más complejas que en los lenguajes de bajo nivel. Los lenguajes de alto nivel más empleados en microcontroladores son el C y el BASIC, de los que existen varias empresas que comercializan versiones de compiladores e intérpretes para diversas familias de microcontroladores, como es el caso del compilador C PCM de la empresa CCS.⁵

Respecto a las herramientas hardware de programación, una indispensable es el programador, encargado de escribir el programa en la memoria del microcontrolador. Uno de los programadores más extendidos, fabricado por

⁴ GONZALEZ, José Adolfo. Introducción a los microcontroladores. Madrid: McGraw Hill, 2002.

⁵ ANGULO, José María. Microcontroladores Pic: Diseño práctico de aplicaciones. Madrid: McGraw Hill, 2003. p.9.

Microchip, es el Pickit 3, el cual permite la programación de la mayoría de los microcontroladores PIC y DSPIC, al igual que memorias eeprom serie de Microchip. La programación se realiza desde un computador personal mediante el puerto USB directamente en el zócalo del programador o en la placa del circuito de control con conexión de 6 líneas mediante programación *In Circuit* (ICSP).⁶

3.2 SISTEMA DE INFORMACIÓN AL USUARIO

Frente al desarrollo de Sistemas Integrados de Transporte Masivo (SITM) en las grandes ciudades, surgió la necesidad de responder a demandas de sistemas eficientes de transporte para ciudades intermedias a través de los denominados Sistemas Estratégicos de Transporte Público (SETP), los cuales corresponden a la estrategia de Ciudades Amables consignada en el Plan Nacional de Desarrollo (PND) 2006 – 2010, “Estado comunitario desarrollo para todos”. Los SETP buscan mejorar la prestación del servicio de transporte público colectivo en ciudades intermedias con el fin de estructurar urbes competitivas, eficientes y equitativas que permitan a los ciudadanos tener oportunidades seguras de movilidad.⁷

El SETP de la ciudad de Pasto, se ha ido constituyendo desde el año 2004; para esto se ha basado en los lineamientos definidos en el Plan Nacional mencionado, y su objetivo primordial es permitir a la población contar con servicios de transporte público que mejoren las condiciones de accesibilidad de manera confiable y con costos acordes con la calidad de la prestación del servicio.

Desde el punto de vista operacional, el SETP de Pasto, ha establecido un reordenamiento de la totalidad de las rutas, proponiendo niveles jerárquicos acordes con los niveles de demanda que tiene la ciudad. La adquisición, operación, mantenimiento y dotación de equipos se realiza a través del sector privado, mientras que el sector público realiza la inversión en infraestructura y tiene a su cargo la planificación, regulación y control del sistema.

Para la ejecución del SETP de Pasto se ha establecido dos fases y se ha jerarquizado de acuerdo con su funcionalidad, para atender los distintos niveles de demanda; se ha establecido 7 rutas estratégicas en la primera fase y una adicional en la segunda para completar 8 rutas estratégicas que operarán por los ejes principales de la ciudad sobre carriles mixtos, con buses de mediana capacidad y, 16 rutas complementarias en la primera fase, disminuyendo en dos rutas en la segunda para contar finalmente con 14 rutas complementarias, que servirán la demanda no atendida por la rutas estratégicas dentro del área de influencia del SETP.

⁶ *Ibíd.*, p.11.

⁷ COLOMBIA. MINISTERIO DE TRANSPORTE. Conpes 3549: Sistema estratégico de transporte público de pasajeros para la ciudad de Pasto. Bogotá, 2008. 45p.

En cuanto a la flota, serán empleadas tres tipologías de vehículos⁸: buses con capacidad de sesenta pasajeros, vehículos adaptados para movilización de pasajeros con movilidad reducida y microbuses. Los vehículos deberán contar con tecnologías de baja contaminación y con los dispositivos de comunicación, control de gestión e ingreso, requeridos para el recaudo electrónico y el sistema de gestión y control de flota.

3.2.1 Señalética de la flota. La señalética es una técnica que permite optimizar, en un sistema de codificación, las señales y signos destinados a orientar y facilitar a las personas respecto a su ubicación y a sus acciones en los distintos espacios y ambientes. Aborda la elaboración de sistemas de señales y sus representaciones considerando factores tales como: ubicación geográfica, lenguaje de la localidad, identidad o elementos representativos del sitio, etc., a fin de que las señales que compongan el sistema señalético sean coherentes no sólo con aquello a lo que se refieren, si no a su entorno.

En el contexto regional, el plan de ordenamiento territorial del municipio de Pasto, en su artículo 78, define la señalización como un elemento complementario del espacio público perteneciente al mobiliario urbano, cuya función es informativa, normativa y de identificación, y cuya ubicación debe además, facilitar la accesibilidad y movilidad de los transeúntes.⁹ En este sentido, la herramienta de información al usuario de transporte público colectivo, que corresponde a uno de los objetivos del SETP, se enmarca en esta definición y se aplica, por tanto, al servicio del usuario de transporte, a su orientación y accesibilidad del servicio para una mayor seguridad en los desplazamientos.

El esquema básico, utilizado en la mayoría de aplicaciones, se compone de un computador a bordo y una antena GPS sólo receptora para evitar costos de transmisión satelital.¹⁰ De esta forma, el computador registra en intervalos de tiempo programables información de fecha, hora, posición (latitud y longitud) y velocidad.

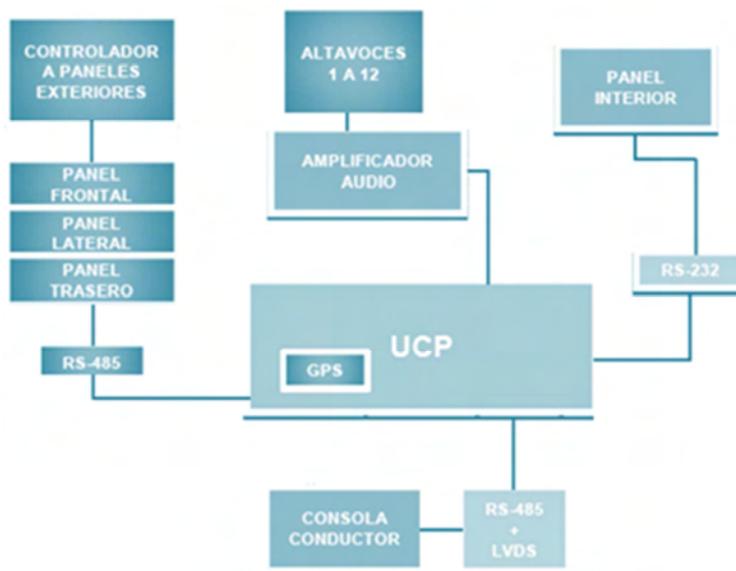
La información visual se presenta de manera dinámica e instantánea en paneles informativos a bordo del autobús, conteniendo información de próxima parada y destino, fecha y hora GPS, información adicional y complementaria e información acústica.

⁸ PASTO. ALCALDÍA MUNICIPAL. Decreto 0735 de 2009: Sistema Estratégico de Transporte Público en el municipio de Pasto. Pasto, 2009. 12p.

⁹ PASTO, ALCALDÍA MUNICIPAL. Acuerdo 026 de 2009. Revisión y ajustes del Plan de Ordenamiento Territorial del municipio de Pasto, 2009. 117p.

¹⁰ MARTÍNEZ, Arturo. Nuevas aplicaciones tecnológicas para la mejora de la eficiencia energética en el transporte: Información al usuario de transporte público. Pamplona. 2007.

Figura 1. Arquitectura embarcada de un sistema de información al viajero



Fuente: MARTÍNEZ, Arturo M. Información al usuario de transporte público. Pamplona, 2007.

Generalmente, los paneles pueden presentar diferentes alternativas en la comunicación; ésta puede ser dependiente o independiente.

En la operación dependiente se establece una comunicación permanente entre computador y el panel electrónico utilizando el estándar RS-232 que describe la interfaz física y el protocolo para la comunicación de datos en serie a baja velocidad entre dispositivos. También es posible realizarlo a través de RS-485. Ésta es una especificación eléctrica de una conexión en serie de dos cables, semidúplex y multipunto, que permite configurar un enlace de comunicaciones con varias segregaciones. Ofrece velocidades de transmisión de datos altas hasta 10 Mbps y, al utilizar una línea diferencial en un par trenzado abarca distancias relativamente grandes. RS-485 sólo especifica las características eléctricas del driver y el receptor pero no especifica un protocolo de datos.

Por otro lado, el modo de operación independiente funciona de forma autónoma sin necesidad de computador y a través de textos configurados en memoria. Sin embargo, al igual que en el modo dependiente, las funciones y mensajes son programados mediante conexión al computador, esto es necesario al momento de programarlo, posteriormente el sistema funciona de forma autónoma y puede ser desconectado del PC.

3.3 INFORMACION VISUAL

3.3.1 Panel electrónico de información al usuario.¹¹ Hace referencia a las pantallas de matriz de leds unilínea de instalación interna y de visualización externa para usuarios potenciales y de visualización interna para usuarios a bordo del bus. Permite informar al usuario potencial sobre el número de identificación, nombre del origen y nombre del destino de la ruta y ruta en servicio, en cualquier hora del día o de la noche, durante la operación del sistema. Al usuario a bordo del bus le suministra, publicidad dentro del autobús y la información necesaria para la identificación de paradas de manera visual y audible. Lo anterior, para facilitar un adecuado nivel de servicio al usuario. Sobre el panel se visualizan las siguientes variables:

- Código de la ruta en operación.
- Hito que identifica el origen o inicio operativo de la ruta.
- Hito que identifica el destino o final operativo de la ruta.
- Tipo de servicio (ruta estratégica, complementaria, otro).
- Próxima parada
- Parada anterior

3.3.2 Características Matriz leds.¹² Matriz es la cantidad de puntos luminosos que forman el panel electrónico de visualización de mensajes. Para su diseño se debe tener en cuenta los siguientes aspectos:

3.3.2.1 Resolución. Es el parámetro descriptivo más importante de un panel electrónico. Se expresa como el número de filas por el número de columnas y de ésta depende la visibilidad del cartel, la calidad de su tipografía y la cantidad de caracteres.

3.3.2.2 Diámetro del punto. Normalmente cada punto se forma de un solo led, cuyo diámetro puede variar de 3mm a 8mm para leds de alta luminosidad y generalmente es de 5 mm de diámetro. Para mejorar la visibilidad se puede optar por usar varios leds por punto, o leds de diámetro mayor.

3.3.2.3 Visibilidad. También llamado alcance visual, es el rango de distancia desde donde el panel electrónico puede ser leído por una persona con vista normal o visión 20/20. La información que se visualiza en un panel electrónico, se lee a mayor distancia que en uno impreso con la misma altura de letra, porque las letras son luminosas. El tamaño de los caracteres influye en la visibilidad a distancia, pero no tanto como la resolución de la matriz y el diámetro de cada

¹¹ CALI. APÉNDICE II DEL CONTRATO. Especificaciones técnicas de los autobuses del Sistema Integrado de Transporte Masivo de Occidente, MIO. Cali, 2007.

¹² DJORDJALIAN, Andrés. Características de los letreros electrónicos de matriz de Leds [en línea]. <<http://www.indicart.com.ar/letreros-electronicos.htm>> [Citado el 20 de Mayo de 2011]

punto luminoso. Estudios indican que una persona con una visión 20/20 a la luz del día puede leer una tipografía de 2.5cm a una distancia de 15 m.

3.3.2.4 Brillo. Se clasifica en brillo estándar, alto brillo, súper e híper brillo y depende de la intensidad de la luz que incide sobre el panel. Un brillo excesivo impide una lectura adecuada, además los leds de alta luminosidad generalmente concentran la luz, empeorando así un parámetro importante denominado ángulo de visualización. Por lo tanto, carteles con más brillo del necesario no son convenientes.

3.3.2.5 Ángulo de Visualización. Se trata del máximo ángulo, respecto al frente del panel electrónico, desde donde se puede posicionar una persona sin dejar de leer el mensaje. El campo normal o cono de visión utilizado en señalización cubre un ángulo de 60°, las áreas por fuera de esta cono tienden a verse con menor detalle.

3.3.2.6 Caracteres. Depende fundamentalmente de cuántas columnas tiene la matriz, es decir de la resolución. Existen técnicas para aumentar la cantidad de caracteres, tales como tipografías de ancho variable, en las cuales no todas las letras tienen el mismo ancho, lo que maximiza la cantidad de caracteres y le da un aspecto elegante al texto.

3.3.2.7 Rango de lectura. Aspecto físico que determina la visualización y percepción de la señal en su contexto de actuación. Entre el público normal existe una amplia variación de este rango que comprende 125 palabras por minuto hasta 500 e incluso 600; sin embargo, factores como la edad y el nivel de educación pueden influir dando como promedio 250 palabras. Considerando este rango de lectura, los señalamientos vehiculares que son expuestos por unos cuantos segundos no deben incluir más de 6 términos.

3.3.3 Técnica de multiplexación. Ésta técnica se sustenta en la capacidad teórica de la retina de guardar la última imagen que le llega, haciendo que un objeto sea percibido incluso cuando ya no está; se trata de un defecto visual de los seres humanos conocido como persistencia de la visión.¹³ El fenómeno óptico fue descubierto por el científico belga Joseph Plateau, y demuestra cómo una imagen permanece en la retina una décima de segundo antes de desaparecer completamente.

A manera de ejemplo, si se tiene un oscilador que controla el encendido y apagado de un led, creando de esta manera una intermitencia, en el momento que ésta es lo suficientemente rápida ya no se aprecia que el led se apaga, sino que proporciona la sensación de estar permanentemente encendido.

¹³ ANÓNIMO. Persistencia retiniana [en línea]. <<http://www.elmulticine.com/glosario2.php?orden=130>> [Citado el 15 de mayo de 2011]

La técnica de multiplexación orientada a la visualización de caracteres de una matriz de leds, consiste en mostrar una columna o fila de la matriz a la vez, lo suficientemente rápido para permitir la percepción que todas se encuentran encendidas y limitar la percepción de parpadeos, por esta razón, la base de tiempo para la multiplexación es muy importante. En la práctica a frecuencias superiores a 200Hz ya no se logra apreciar el parpadeo.

La técnica de multiplexación permite utilizar sólo algunos pines de E/S de un microcontrolador para manejar una serie de circuitos integrados que se encargan de excitar los leds. Existen varias maneras para lograr ese propósito, entre estas, el empleo de arreglos de registros o Flip-Flops como elementos de memoria temporal.

3.4 INFORMACIÓN AUDIBLE

El proceso de codificación de audio tiene como objetivo la representación eficiente en formato digital de una señal de audio para su almacenamiento y/o transmisión, sin pérdida de calidad. Desde el punto de vista del almacenamiento de la señal de voz en formato digital, la codificación de audio permite minimizar el número de bits necesarios para su almacenamiento manteniendo un nivel de calidad adecuado.

El objetivo fundamental de la codificación de audio es la conversión de la señal de voz a una secuencia binaria o representación digital. Dado el carácter analógico (señal continua en tiempo y amplitud) de la señal de voz, la codificación de audio conlleva un proceso básico de muestreo (discretización en el tiempo) y cuantificación (discretización en amplitud) para lograr una representación digital.

La frecuencia de muestreo y el número de bits por muestra son dos de los parámetros fundamentales a elegir cuando se requiere procesar digitalmente una determinada señal de audio.

3.4.1 Codificación PCM. Una de las técnicas más utilizadas en la conversión de una señal analógica a digital, para un posterior proceso de digitalización, es la modulación por codificación de pulsos PCM (*Pulse Code Modulation*).

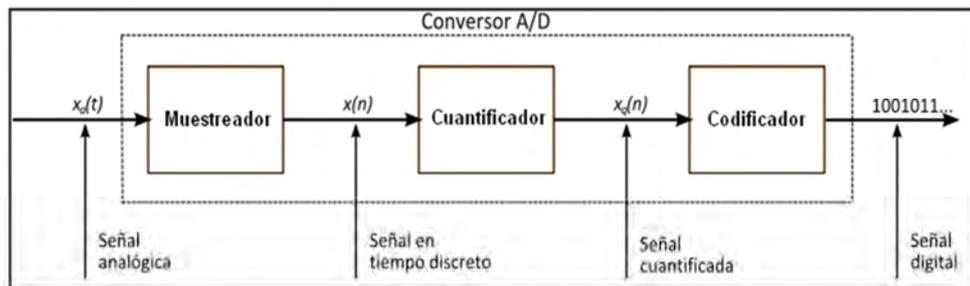
Para la digitalización del sonido mediante un PC, se emplea la placa de sonido y un micrófono conectado a ésta.¹⁴ En la conversión, estos dispositivos utilizan, como elemento intermedio, un dispositivo denominado conversor analógico-digital (A/D) o ADC (Analogic to Digital Converter), que recibe las señales eléctricas en

¹⁴ PROAKIS, J. G. Tratamiento digital de señales: Principios, algoritmos y aplicaciones. Madrid: Prentice Hall Internacional Ltd., 1998.

forma de onda senoidal analógica proporcionadas por el micrófono y a continuación las convierte en señales digitales, codificadas en valores numéricos binarios. El dispositivo digitalizador, en este caso, la placa de sonido del PC, emplea la técnica PCM.

Para realizar este proceso, el conversor A/D efectúa un proceso de muestreo de la señal analógica, cuantificación de la propia señal y codificación del resultado de la cuantificación, en código binario. Gráficamente, se puede representar de acuerdo al diagrama de bloques de la figura 2.

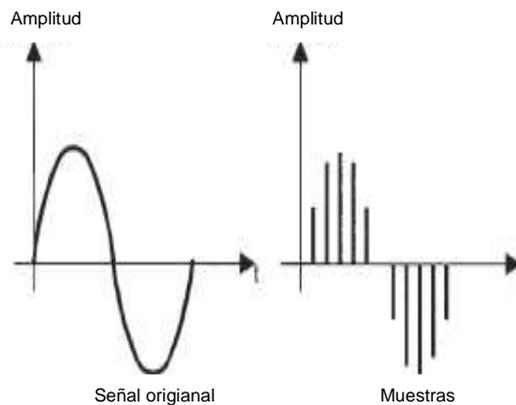
Figura 2. Diagrama de bloques de un conversor analógico digital



Fuente: Autor.

3.4.1.1 Muestreo. El muestreo toma un número discreto de valores de la señal analógica a intervalos de tiempo constante (tasa de muestreo).

Figura 3. Muestras tomadas de una señal analógica



Fuente: Autor.

El muestreo genera pulsos comprendidos en el rango de amplitudes de la señal, el conjunto de amplitudes puede ser infinito, con valores no enteros. Debido a que estos valores no pueden ser usados en el proceso de codificación, se procede a la

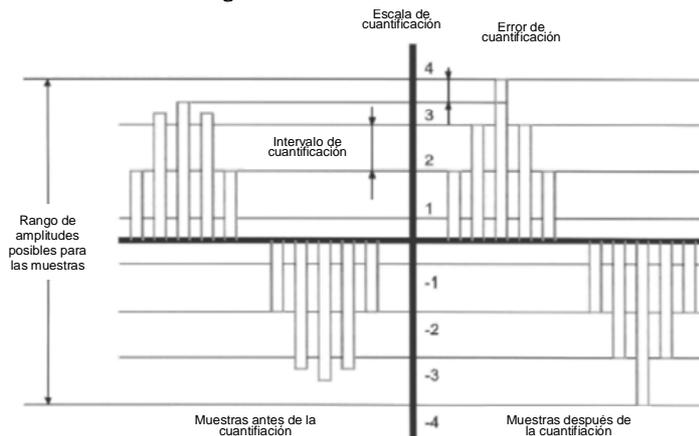
cuantificación. La figura 3 indica el resultado obtenido después de haber realizado un muestreo a la señal analógica.

Frecuencia de muestreo. También llamada tasa de muestreo, es la cantidad de muestras por unidad de tiempo. Las frecuencias de muestreo más utilizadas en audio digital son 8KHz, 11KHz, 16KHz, 22KHz, 32KHz, 44,1KHz y 48KHz. En el caso de una grabación digital de audio, a mayor cantidad de muestras tomadas, es decir, cuanto mayor es la tasa de muestreo, mejor es la calidad y fidelidad de la señal digital resultante.

Para que en el proceso de digitalización no exista pérdida de información, según el teorema de Nyquist, la frecuencia de muestreo debe ser como mínimo el doble de la frecuencia más alta presente en la señal a discretizar. En una señal de voz, corresponde a la frecuencia más aguda que se puede escuchar, por lo que la frecuencia óptima de muestreo, que es también la utilizada en los CD de música, es 44.1KHz. Sin embargo, si el propósito es digitalizar sonidos acústicos, no necesariamente se debe alcanzar una frecuencia de muestro estándar, ya que la voz humana no produce sonidos con frecuencias superiores a los 10KHz por lo que en estos casos se puede utilizar una frecuencia de 22KHz, o incluso una inferior.¹⁵

3.4.1.2 Cuantificación. Durante éste proceso se asigna valores numéricos equivalentes a la tensión existente en diferentes puntos de la sinusoide. Se trata de un proceso de discretización en amplitud, ya que divide el rango de amplitudes en un número limitado de intervalos de forma que las muestras que se encuentran en un mismo intervalo toman el mismo valor.

Figura 4. Asignación de valores analógicos discretos a las muestras



Fuente: COIMBRA, Edison. Digitalización de señal analógica PCM. Santa Cruz de la Sierra, Bolivia, 2006.

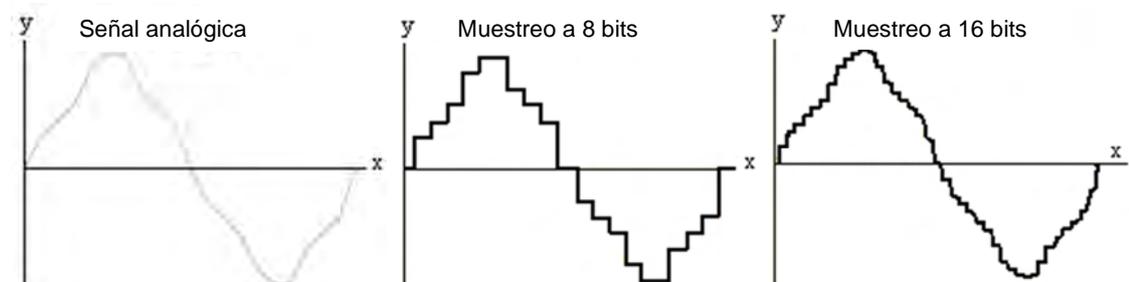
¹⁵ MIYARA, Federico. Preguntas frecuentes sobre audio digital [en línea]. <<http://www.cetear.com/FAQad.pdf>> [Citado el 22 de Mayo de 2011]

La asignación de valores analógicos discretos a las muestras supone una pérdida de información, conocido como error de cuantificación, debido a que la amplitud real de las muestras es sustituida por una amplitud aproximada. Este error deforma la señal reconstruida y da lugar a una distorsión denominada distorsión o ruido de cuantificación que debe ser imperceptible para el oído humano.

Bits de resolución. La resolución es la cantidad de bits que se utiliza para representar las muestras de audio; cuanto mayor es la resolución, mayor precisión se logra en la representación. El número de bits por muestra determina el número de niveles de cuantificación y éstos a su vez determinan la relación señal-ruido de pico de portadora que depende de forma cuadrática al número de bits por muestra.

La amplitud total se divide en tantos niveles como la resolución en bits lo permite, de acuerdo a la expresión 2^N , siendo N el número de bits. De esta forma, con una resolución de 8 bits, el rango de variación de la señal se divide en 256 subintervalos (2^8), mientras que con una resolución de 16 bits lo hace en 65.536 subintervalos (2^{16}), cuya amplitud es, por consiguiente, mucho menor. De acuerdo a la figura 5, cuanto mayor es el número de bits, más valores intermedios se obtienen, lo que hace que los niveles se reduzcan y la onda digitalizada se vuelva más parecida a la señal original.

Figura 5. Diferencia de resolución de una señal muestreada



Fuente: Autor.

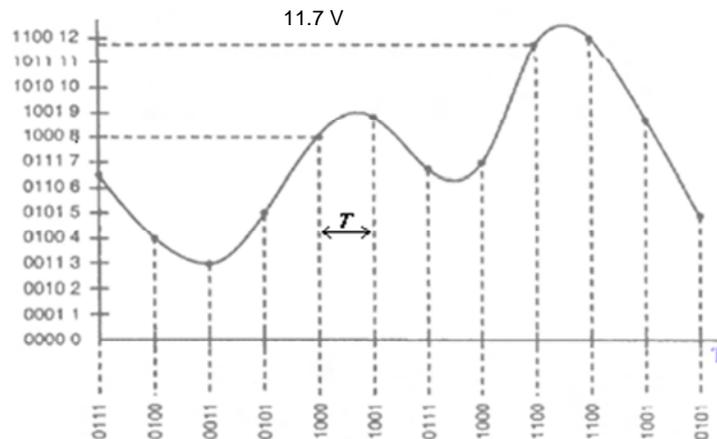
Un muestreo realizado a 16 bits tiene mejor respuesta en frecuencia, que otro realizado a menor resolución, por lo que se puede conseguir una calidad de sonido igual al de CD de audio, mientras que un muestreo a 8 bits, aunque no contiene las frecuencias más altas de la señal analógica originaria, es muy útil para grabar o reproducir voz, lográndose una calidad similar a la de radio.

3.4.1.3 Codificación. Es la última etapa en la modulación PCM. Mediante la codificación se representa las muestras cuantificadas por razón de una secuencia

binaria, esto es, se asigna una secuencia de bits a cada valor analógico discreto. La longitud de la secuencia de bits es función del número de niveles analógicos empleados en la cuantificación.

El proceso de codificación, propiamente dicho, toma como señal de entrada la obtenida en la conversión A/D utilizando ciertas propiedades de la señal de voz, y obteniendo una nueva codificación con una velocidad inferior a la inicial.

Figura 6. Salida de un convertidor A/D en la etapa de codificación



Fuente: COIMBRA, Edison. Digitalización de señal analógica PCM. Santa Cruz de la Sierra, Bolivia, 2006.

La figura 6 muestra los números binarios generados por un convertidor A/D para cada uno de los instantes de muestreo de la señal digitalizada a una frecuencia de muestreo $1/T$, con 16 niveles de cuantificación y una codificación de 4 bits por muestra.

3.4.2 Formatos de archivos de audio. El sonido o audio digital se obtiene mediante la aplicación de la técnica PCM descrita anteriormente, donde la señal analógica, después de atravesar un sistema de procesamiento analógico, se muestrea, cuantifica y codifica. Un sistema completo de audio digital termina con el proceso inverso al descrito: de la representación digital almacenada se obtiene el conjunto de muestras que representa, las cuales pasan por un proceso de conversión digital-analógico, proporcionando una señal analógica que tras un procesamiento (filtrado, amplificación, ecualización, etc.) incide sobre el transceptor de salida (altavoz).

Los formatos de archivos de audio son utilizados para almacenar audio digital; estos formatos contienen información sobre las características de la forma de

onda del audio y la manera en que ésta se almacena, permitiendo el almacenamiento y decodificación del sonido codificado digitalmente.

Un formato de archivo de audio, por lo general, especifica la frecuencia de muestreo, la resolución y el tipo de compresión utilizado, así mismo, información que indica la cantidad de datos que se debe leer por segundo, en el caso de una conversión D/A, etc. La mayor parte del contenido de un archivo de audio se utiliza para describir las muestras individuales de audio que son reproducidas en secuencia para reconstruir la forma de onda original.

Los formatos WAV, AIFF, SU, AU y RAW, se obtienen mediante modulación por pulsos digitales; la calidad del audio digital representado por PCM depende de la frecuencia de muestreo y el número de bits por muestra. Estos formatos contienen toda la información que sale del conversor A/D, sin ninguna omisión y por eso, tienen la denominación de formatos no comprimidos.

3.4.2.1 Formato de archivo Wave. El formato de archivo Wave (Waveform audio file), es el formato de archivo nativo de Windows para almacenar datos de audio digital y se ha convertido en uno de los formatos de archivo digital de audio más popular de Windows debido a que casi todos los programas modernos que pueden abrir o guardar archivos de audio digital son compatibles con éste, lo que lo hace extremadamente útil.¹⁶

Aunque los archivos Wave pueden tener un excelente sonido comparable al del CD (16 bytes y 44,1Khz. estéreo) el tamaño necesario para esa calidad es muy grande ya que por cada minuto de grabación de sonido se ocupa aproximadamente 10Mbytes de espacio en disco, limitando el tamaño del fichero a 4 gigabytes, equivalentes a 6,6 horas en calidad de CD de audio. Sin embargo, la sencillez de este formato lo hace ideal para el tratamiento digital del sonido.¹⁷ En un fichero Wave se almacena las muestras una tras otra, a continuación de la cabecera del fichero, sin ningún tipo de compresión de datos y con cuantificación uniforme.¹⁸

Este formato de archivo permite almacenar audio digital sin compresión de datos y debido a su flexibilidad es muy usado en el tratamiento digital del sonido, ya que permite grabaciones a distintas calidades y a diferentes frecuencias de muestreo (8000, 11025, 22050 y 44100 Hz). A pesar de que el formato Wave puede soportar

¹⁶ TIMONEDA, José Miguel. Archivos de sonido digital [en línea]. <<http://cursoryc.educa.aragon.es/comunes/modulo3/unidad2.htm>> [Citado el 26 de mayo de 2011]

¹⁷ TRONCHONI, Pepe. Formato de los ficheros de sonido wav [en línea]. http://www.tav.net/audio/formato_wav.htm [Citado el 26 de mayo de 2011]

¹⁸ BORN, Gunter. Wav file format description [en línea]. <http://www.tav.net/audio/wav_format.htm> [Citado el 26 de mayo de 2011]

casi cualquier códec de audio, se utiliza principalmente con el formato PCM lo que no representa pérdida de calidad.

Formato de datos. Por defecto, todos los valores de los datos son almacenados en Little-endian, esto es, los datos que ocupan más de un byte son representados según el formato Intel, primero los bytes LSB (Least Significant Bit) y a continuación los MSB (Most Significant Bit).

Strings. El archivo Wave contiene cadenas de texto para especificar las etiquetas de un punto de referencia, notas, etc., las cuales son almacenadas en un formato donde el primer byte especifica el número de bytes que contiene el texto ASCII de la cadena, por ejemplo: 7 'e' 'x' 'a' 'm' 'p' 'l' 'e'

Arquitectura del archivo Wave. El tipo de arquitectura utilizado para los archivos Wave es el denominado Formato de archivos de intercambio de recursos RIFF (*Resource Interchange File Format*) de Microsoft, el cual constituye un marco de referencia para el almacenamiento de segmentos ó "chunks" de información multimedia.

Tabla 1. Arquitectura del archivo Wave en su forma canónica

Segmento identificación	
Byte	Descripción
0 - 3	"RIFF" (Identificador en ASCII)
4 - 7	Tamaño del fichero en bytes (Binario, little-endian)
8 - 11	"WAVE" (Identificador en ASCII)
Segmento formato	
0 - 3	"fmt" (Identificador en ASCII)
4 - 7	Tamaño de la cabecera (Binario)
8 - 9	Etiqueta de formato. (0x01)
10 - 11	Número de canales (0x01=Mono, 0x02= Estéreo)
12 - 15	Frecuencia de Muestreo (Binario, en Hz)
16 - 19	Número de bytes por segundo
20 - 21	Número de bytes por muestra (1=8 bit Mono, 2=8 bit Estéreo o 16 bit Mono, 4=16 bit Estéreo)
22 - 23	Bits de resolución (8,16,32)
Segmento datos	
0 - 3	"data" (Identificador en ASCII)
4 - 7	Número de bytes muestreados
8 - end	Muestras (Cuantificación uniforme)

Fuente: BORN, Gunter. Wav file format description. Boston, 1995.

Segmento Identificación. El archivo Wave se almacena dentro de un fichero, según el estándar RIFF, cuya cabecera de 8 bytes identifica el fichero y especifica la longitud de los datos a partir de la cabecera. La cabecera se compone de 4 bytes con el contenido "RIFF" y otros 4 que indican la longitud. Los 4 bytes posteriores a la cabecera RIFF, identifican el tipo de datos que contiene, que para el caso de un archivo Wave contiene la palabra "WAVE" en caracteres ASCII.

Tabla 2. Valores típicos del segmento RIFF para un archivo Wave

Offset	Tamaño	Descripción	Valor
0x00	4	Segmento ID	"RIFF" (0x52494646)
0x04	4	Longitud de los datos	(tamaño total) - 8
0x08	4	Tipo de formato RIFF	"WAVE"(0x57415645)
0x10		Segmento Wave	

Fuente: BORN, Gunter. Wav file format description. Boston, 1995.

Existen varios tipos de segmentos definidos para los archivos Wave. Muchos archivos Wave contienen sólo dos de ellos, específicamente el segmento formato y el segmento de datos. Estos son los dos segmentos necesarios para describir el formato de las muestras de audio digital y las propias muestras.

Segmento formato. El segmento formato se identifica por "fmt" y se compone de dos elementos a saber, el conjunto de campos comunes y el conjunto de campos específicos. En este segmento se identifican parámetros como tipo de compresión utilizado, número de canales, frecuencia de muestreo y bits por muestra y longitud de la muestra, entre otros.

Tabla 3. Formato de valores del segmento Wave

Offset	Tamaño	Descripción	Valor
0x00	4	Segmento ID	"fmt " (0x666D7420)
0x04	4	Longitud de los datos	16 + extra format bytes
0x08	2	Compresión de código	1 - 65,535
0x0a	2	Número de canales	1 - 65,535
0x0c	4	Frecuencia de muestreo	1 - 0xFFFFFFFF
0x10	4	Promedio de bytes por segundo	1 - 0xFFFFFFFF
0x14	2	Alineación de bloque	1 - 65,535
0x16	2	Bits significativos por muestra	2 - 65,535
0x18	2	Bytes extra del formato	0 - 65,535

Fuente: BORN, Gunter. Wav file format description. Boston, 1995.

Compresión de código. Los datos de un formato Wave pueden ser comprimidos de diferentes maneras para reducir el tamaño de las muestras de audio. Cada segundo de calidad CD de audio sin comprimir (44,1 kHz, 16 bits, estéreo) requieren alrededor de 172 kilobytes.

La primera palabra del segmento datos, especifica el tipo de compresión utilizado en los datos incluidos en el segmento "RIFF". La Tabla 4 contiene una lista de los códigos de compresión más comunes.

Tabla 4. Tipos de compresión del formato Wave

Código	Descripción
0 (0x0000)	Desconocido
1 (0x0001)	PCM (sin compresión)
2 (0x0002)	Microsoft ADPCM
6 (0x0006)	ITU G.711 a-law
7 (0x0007)	ITU G.711 μ -law
17 (0x0011)	IMA ADPCM
20 (0x0016)	ITU G.723 ADPCM (Yamaha)
49 (0x0031)	GSM 6.10
64 (0x0040)	ITU G.721 ADPCM
80 (0x0050)	MPEG
65,536 (0xFFFF)	Experimental

Fuente: LÓPEZ, Alberto. Ingeniería de Ondas formatos de audio digital. Valladolid, 2009.

Número de canales. El número de canales especifica el número de señales de audio por separado que son codificadas en el segmento datos. Un valor de 1 representa una señal mono y un valor de 2 representa una señal estéreo.

Frecuencia de muestreo. Es el número de muestras por unidad de tiempo tomadas durante el proceso de conversión, se expresa en hercios.

Promedio de bytes por segundo. Indica cuántos bytes de datos deben ser transmitidos a un convertor D/A por segundo para poder reproducir el archivo Wave. Esta información es útil para determinar si los datos pueden transmitirse lo suficientemente rápido para reproducirse. Se puede calcular con la expresión:

$$\text{Promedio bytes por segundo} = \text{Frecuencia muestreo} * \text{Alineación Bloque} \quad (1)$$

Alineación de bloque. Es el número de bytes por segmento de la muestra. Este valor se ve afectado por el número de canales y se puede calcular con la expresión:

$$\text{Alineación} = \frac{SBPS}{8 * \text{Número Canales}} \quad (2)$$

Bits significativos por muestra (SBPS). Especifica el número de bits utilizados para definir cada muestra, suele ser de 8, 16, 24 o 32. Si el número de bits no es un byte alineado (múltiplo de 8), el número de bytes utilizados por cada muestra se redondea al tamaño del byte más próximo, los bytes sin uso se ponen a 0 y son ignorados.

Bytes extras del formato. Define el número de bytes adicionales del formato. Su valor es nulo si el código de compresión es de 0 (archivo sin comprimir PCM), pero puede tomar cualquier valor para los tipos de compresión en función de la información que la compresión necesite para decodificar los datos.

Segmento datos. Contiene las muestras reales. Puede ser de dos tipos: tipo data o tipo data-list. El primero es una secuencia de datos sin más elementos. El segundo en una secuencia de dos tipos de segmentos: datos o silencio. Un segmento tipo dato se identifica por "data", mientras que un segmento tipo data-list se identifica por "wavl". Los segmentos de silencio ("slnt") sólo contienen un campo que es una doble palabra con el número de muestras a mantener el silencio.

El segmento de datos Wave contiene los datos de audio digital de la muestra que pueden ser decodificados con el método y el formato de compresión especificado en el segmento Formato. Si el código de compresión es 1 (PCM sin comprimir), entonces la onda contiene valores de datos en bruto de la muestra.

Tabla 5. Formato de valores del segmento Datos

Offset	Longitud	Tipo	Descripción	Valor
0x00	4	char[4]	Segmento ID	"data" (0x64617461)
0x04	4	dword	Tamaño del segmento	Depende de las muestras y la compresión
0x08	Datos de la muestra			

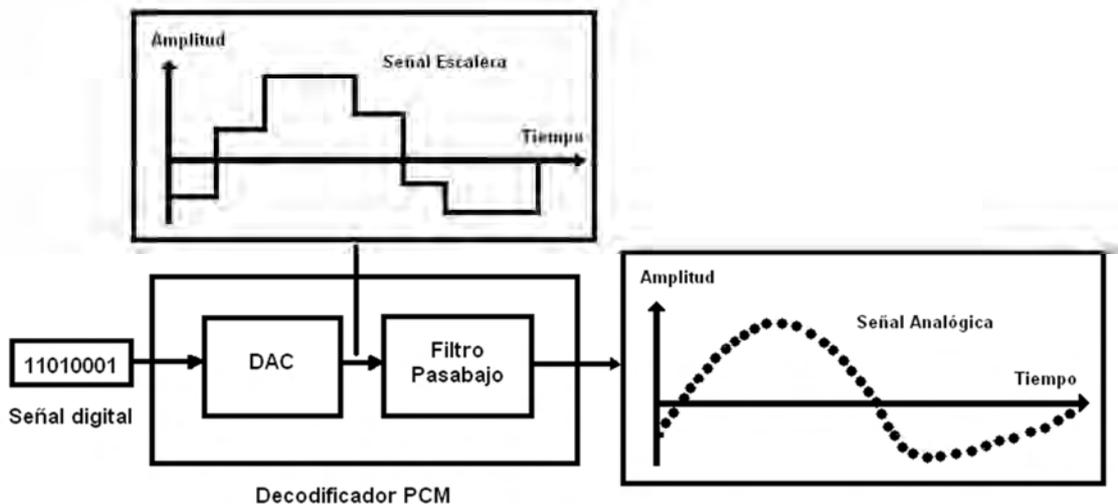
Fuente: BORN, Gunter. Wav file format description. Boston, 1995.

3.4.3 Conversión digital analógica. En muchas aplicaciones se digitaliza una señal analógica, lo cual significa que varios puntos sucesivos de ésta se convierten en su equivalente digital y son guardados en memoria. Esta conversión se lleva a cabo con un conversor analógico-digital. El conversor digital analógico se puede utilizar para convertir los datos digitales, de un punto a la vez, en una señal analógica, y reconstruir de esta manera, la señal original. La combinación de digitalización y reconstrucción es el proceso aplicado en los sistemas de audio.

La recuperación de la señal original requiere de un decodificador PCM. Básicamente, consiste de un conversor D/A el cual, a partir de un valor representado en código digital, lo convierte en un voltaje o corriente proporcional al valor digital; el voltaje o corriente toma diferentes valores de cierto intervalo mediante un circuito que convierte las palabras del código en un pulso que mantiene la amplitud hasta el siguiente pulso.

La salida del conversor D/A no es una cantidad analógica ya que sólo puede tomar valores específicos. De este modo, la salida en realidad es digital (señal escalera) como se indica en la figura 7. Sin embargo, se puede reducir la diferencia entre dos valores consecutivos al aumentar el número de salidas, mediante el incremento del número de bits de entrada.

Figura 7. Recuperación de la señal original mediante conversión D/A



Fuente: COIMBRA, Edison. Digitalización de señal analógica PCM. Santa Cruz de la Sierra, Bolivia, 2006.

El incremento del número de bits de entrada permite producir una salida cada vez más similar a la cantidad analógica, que cambia de manera continua sobre un rango de valores. En otras palabras, la salida del conversor D/A es una cantidad “pseudoanalógica”.

Por tal razón, a la salida del conversor D/A se dispone de un filtro, con la misma frecuencia de corte que la señal original en el emisor, para lograr suavizar la señal escalera y obtener finalmente una señal analógica; el valor máximo y mínimo de la señal original se obtiene mediante amplificación.

3.4.3.1 Características de los conversores D/A.¹⁹ Para una aplicación efectiva de los conversores digital-analógicos es preciso conocer y saber interpretar las especificaciones de los mismos, ya que ponen de manifiesto las limitaciones, así como las verdaderas prestaciones, que en muchos casos difieren de la idealidad considerablemente. Los principales parámetros que condicionan la selección de un convertidor son, entre otros, el número de bits y el tiempo de establecimiento. El primero determina la resolución de la interfaz y el segundo la velocidad de conversión.²⁰

Estos parámetros permiten la elección del conversor D/A más adecuado a las necesidades, teniendo en cuenta que su comportamiento no es ideal en absoluto. El tipo de información que facilitan los fabricantes sobre los conversores D/A es el tipo de salida que suministra el sistema que puede ser tensión o intensidad; el código digital de entrada que puede ser gray, binario natural, código bipolar, entre otros. También se especifica la tensión de referencia (V_{ref}), en cuyo caso se determina las características de la misma y su variación respecto a la temperatura.

3.4.4 Estándar SD.²¹ En secciones anteriores se hizo referencia de la técnica PCM para codificar audio digital para su posterior almacenamiento. Ahora bien, las siguientes secciones abarcan una descripción de la arquitectura interna y modo de funcionamiento de uno de los dispositivos de almacenamiento más extendido hoy en día, las tarjetas de memoria SD.

SD (Security Digital) es un estándar de las tarjetas de memoria extraíble para el almacenamiento de datos, diseñado y autorizado por la SD Card Association. El estándar SD es en gran medida un esfuerzo de colaboración de tres fabricantes, Toshiba, SanDisk, y MEI, y surgió del estándar MMC (Multi Media Card).

El estándar SD no sólo se limita a los dispositivos extraíbles de almacenamiento sino que ha sido adaptado a muchas clases de dispositivos, incluyendo tarjetas 802.11, dispositivos Bluetooth y módems.

En la figura 8 se muestra la arquitectura interna de una tarjeta de memoria SD. En esta se observa como todas las unidades de la tarjeta están sincronizadas por un generador de reloj interno. La unidad de control de la interfaz sincroniza las señales DAT y CMD del CLK externo utilizado, a la señal del reloj interno. La organización interna de la tarjeta está constituida por un área de almacenamiento y una zona de registros de configuración e información. A su vez, el área de almacenamiento se subdivide en 2 áreas independientes, el área protegida, a la cual no se puede acceder a menos que el dispositivo se identifique, y el área de

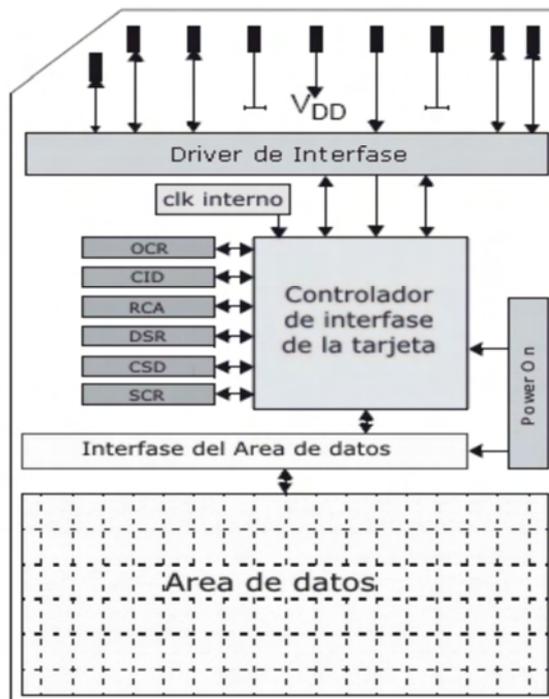
¹⁹ DOMÍNGUEZ, Fernando. Lógica digital y microprogramable. España: Mc Graw Hill, 1999.

²⁰ GONZÁLES, Juan José. Instrumentación electrónica: Conversión Digital/Analógica [en línea]. <http://www.uca.es/grupnvest/instrument_electro/T10_CDA.pdf> [Citado el 27 de Mayo de 2011]

²¹ KINGMAX DIGITAL INC. Secure digital memory card specification [en línea]. <<http://downloads.amilda.org/MODs/SDCard/SD.pdf>> [Citado el 20 de mayo de 2011]

almacenamiento común o de usuario, que ofrece la posibilidad de guardar cualquier tipo de información.

Figura 8. Arquitectura interna de la tarjeta SD



Fuente: KINGMAX DIGITAL INC. Secure digital memory card specification.

La zona de registros esta conformada por 6 registros a saber OCR, CID, RCA, DSR, CSD y SCR. Así por ejemplo, para la identificación de la tarjeta SD en un arreglo de tarjetas SD, se requiere del registro de identificación de la tarjeta (CID) y del registro de dirección (RCA). La tarjeta se controla mediante seis líneas que contienen las señales: CMD, CLK, DAT0 - DAT3. La comunicación utilizando las líneas anteriores, para acceder a un campo de tarjeta o a un registro cualquiera, se define por el estándar SD en el modo cuatro-bit SD. Sin embargo, las tarjetas SD soportan un segundo modo de operación mediante una interfaz SPI (Serial Peripheral Interface); éste modo se activa si la señal CS se configura durante la recepción del comando reset.

3.4.4.1 Interfaz. Existen 3 modos de transferencia soportados por SD: modo SPI (entrada serial y salida serial), modo un-bit SD (separa comandos, canales de datos y un formato propietario de transferencia) y modo cuatro-bit SD (utiliza terminales extra más algunas terminales reasignadas) éste último soporta transferencias paralelas de cuatro bits.

Figura 9. Diagrama de pines, vista posterior y frontal de una tarjeta SD



Fuente: SANDISK Co.

El modo SD permite un alto rendimiento en la transferencia de datos, en tanto que el modo SPI, aunque representa una pérdida de rendimiento relativo al modo SD, permite una interfaz sencilla y común por un canal estándar SPI.

En la tabla 6 se describe la función de los pines para cada uno de los dos modos principales, el modo SD y el modo SPI.²² En ésta se evidencia que muchos de los pines de la tarjeta SD son de doble propósito.

Tabla 6. Asignación de pines en una tarjeta SD para los modos SD y SPI

Pin	Nombre	Modo SD	Modo SPI
1	DAT3/CS	Línea de datos (Bit 3)	Selección de dispositivo (SS)
2	CMD/DI	Línea de comandos	Master Out Slave In (MOSI)
3	VSS1	Tierra	Tierra
4	VDD	Fuente de voltaje	Fuente de voltaje
5	CLK	Reloj	Reloj (SCK)
6	VSS2	Tierra	Tierra
7	DAT0/D0	Línea de datos (Bit 0)	Master In Slave Out (MISO)
8	DAT1/IRQ	Línea de datos (Bit 1)	N.C o IRQ
9	DAT2/NC	Línea de datos (Bit 2)	N.C

Fuente: Autor.

A continuación se describe las principales características de los tres modos compatibles con el estándar SD. El protocolo SD de 1 bit es un protocolo síncrono serie con una línea de datos utilizada para la transferencia de datos, una línea de reloj para la sincronización, y una línea de comandos, que se utiliza para el envío de comandos. Este protocolo permite compartir el bus, admitiendo múltiples tarjetas SD con una única señal de reloj y una única línea DAT0.

²² APACER TECHNOLOGY. Application note secure digital card interface. Taipei, Taiwan, 2005. 100p.

El protocolo SD de 4 bits es casi idéntico al protocolo SD de 1 bit. La principal diferencia radica en la transferencia de datos del bus, la cual se produce en paralelo a 4 bits en lugar de un bit. Con un diseño apropiado, este modo tiene el potencial de cuadruplicar el rendimiento de la transferencia de datos.

Tanto el protocolo SD de 1 bit como el de 4 bits, requieren protección o comprobación de redundancia cíclica (CRC), para detectar la presencia de errores de inversión de bits en un bloque de transmisión de datos. Por ejemplo, en el modo SD de 4 bits, los datos de entrada son multiplexados con la línea DAT y el CRC, de 16 bits, debe ser calculado de forma independiente para cada una de las cuatro líneas. Esto implica que, en una aplicación software, el cálculo de la CRC llegaría a ser tan complejo, que la sobrecarga computacional lograría mitigar los beneficios del bus de 4 bits.²³

El tercer protocolo compatible al protocolo SD es el modo SPI. Este es distinto de los protocolos de 1 bit y de 4 bits, ya que se basa en la interfaz de bus genérica y conocida, SPI. Éste es un protocolo serie síncrono muy popular para interconectar dispositivos periféricos con microcontroladores pero sólo admite un subconjunto de la totalidad de los comandos del protocolo SD. Sin embargo, la mayoría de los comandos no compatibles simplemente no son necesarios en el modo SPI y se puede implementar una aplicación completamente funcional sólo con SPI.

3.4.4.2 Bus en el modo SPI.²⁴ El modo SPI consiste en un protocolo de comunicación secundaria ofrecido por las tarjetas de memoria SD. Este modo está diseñado para comunicarse por un canal SPI, comúnmente encontrado en los microcontroladores. La interfaz es seleccionada durante la primera orden después de ser conectada la tarjeta de memoria y no puede ser cambiada una vez se enciende.

La figura 10 muestra la interfaz eléctrica para el modo SPI. Las resistencias de pull-up externas son requeridas por el protocolo y deben estar presentes, incluso para los pines de datos no utilizados. La alimentación y las señales deben encontrarse en el rango de los 2,6V a 3,7V.

La interconexión serie SPI consta de cuatro señales:

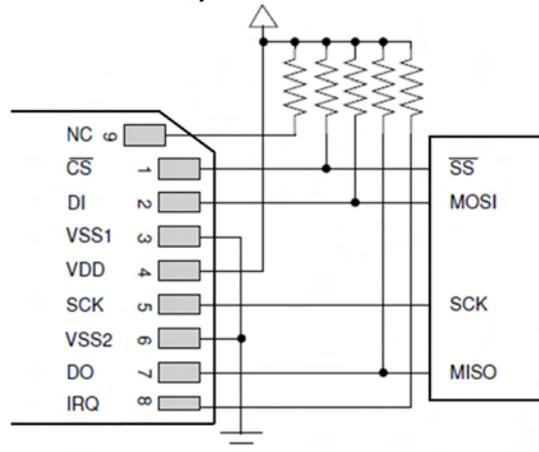
- SCLK (Serial Clock). Es la señal de reloj generada por el controlador y es la que marca el ritmo de transferencia de la información serie entre ambos, así los datos se capturan o transmiten por la tarjeta al ritmo marcado por esta señal.
- MISO (Master In Slave Out data): Une la entrada de datos del microcontrolador con la salida de datos serie de la tarjeta SD.

²³ *Ibid.*, p.15.

²⁴ SANDISK CORPORATION. SanDisk secure digital card. Estados Unidos, 2003. 113p.

- MOSI (Master Out Slave In data): Conecta la salida de datos del maestro a la entrada de datos serie del dispositivo esclavo.
- SS (Slave Selector): Permite seleccionar distintos dispositivos conectados a un mismo bus. Cada dispositivo conectado al bus requiere de una señal SS.

Figura 10. Diagrama de conexión de la tarjeta de memoria SD en el modo SPI



Fuente: SD CARD ASSOCIATION. SD memory card specifications. Versión 1.01. San Ramon, CA.2001.

3.4.4.3 Comandos. Todos los comandos son iniciados por el maestro. La tarjeta SD responde al comando con un margen de respuesta y, a continuación, en función de la orden, puede ser seguido por un token de datos que indica el comienzo de una transferencia masiva de datos o una condición de error.

Los comandos SD son emitidos en una trama de 6 bytes enviada a través del puerto SPI. La cabecera siempre contiene "01" seguido del número de identificación del comando codificado en binario natural. Así el primer byte correspondiente al comando 39 (CMD39) es 01100111. Los siguientes 4 bytes corresponden al argumento del comando y son enviados primero los bits más significativos (MSB). Si el comando no tiene argumento entonces el valor enviado es cero.

Finalmente es enviado el byte CRC cuyo LSB correspondiente a "1" marca la parada. Éste último byte no se utiliza en el protocolo SPI, a no ser que en el registro de configuración se especifique que se desea utilizar. La Tabla 7 muestra el formato de un comando. Se debe tener en cuenta que CRC es opcional en el modo SPI y por defecto la comprobación CRC está desactivada.

Tabla 7. Estructura de un comando del estándar SD

Primer byte			Bytes 2-5	Último byte	
7	6	5-0	31-0	7-1	0
0	1	Comando	Argumento (primero los MSB)	CRC	1

Fuente: Autor.

Toda la comunicación entre la tarjeta y el controlador se realiza de izquierda a derecha, es decir que primero se transmite el bit de más peso (bit 7) del primer byte, y por último el bit de menos peso (bit 0) del sexto byte, lo que indica que es una transferencia MSB.

Los comandos SD son dispuestos en la forma CMDXX o ACMDXX, donde las etiquetas CMD y ACMD hacen referencia a comandos generales y comandos de aplicación específicos, respectivamente, y XX es el número de identificación del comando. La tabla 8 muestra los principales comandos.

Tabla 8. Principales comandos del protocolo de acceso mediante SPI

Comando	Argumento	Respuesta	Descripción
CMD0	No	R1	Resetea la tarjeta
CMD1	No	R1	Inicializa la tarjeta
CMD9	No	R1	Pide a la tarjeta su información CSD
CMD10	No	R1	Pide a la tarjeta su identificación CID
CMD13	No	R2	Consulta el estado de la tarjeta
CMD16	[31...0] Longitud del bloque.	R1	Establece la longitud (en bytes) del bloque para los datos en las operaciones de lectura y escritura.
CMD17	[31...0] Dirección de datos.	R1	Lee un bloque del tamaño indicado por el comando 16.
CMD24	[31..0] Dirección de datos	3 x R1	Escribe un bloque del tamaño indicado por el comando 16.

Fuente: SD CARD ASSOCIATION. SD memory card specifications. Versión 1.01. San Ramon, CA.2001.

Secuencia de reset. Por defecto, al arrancar la tarjeta, ésta se encuentra en modo SD. Para que entre en modo SPI, se envía el comando 0 (CMD0) mientras se mantiene activa en bajo la señal CS (CS=0); ésta debe estar continuamente activa durante la duración de la trasmisión, la única excepción es el momento de la programación.

Inicialización de la tarjeta. Una vez reseteada y en modo SPI, se realiza la inicialización de la tarjeta mediante el envío del comando 1 (CMD1), hasta que el bit "idle" en la respuesta R1 sea 0. Esto indica que la tarjeta ha completado su inicialización y se encuentra lista para recibir nuevos comandos.

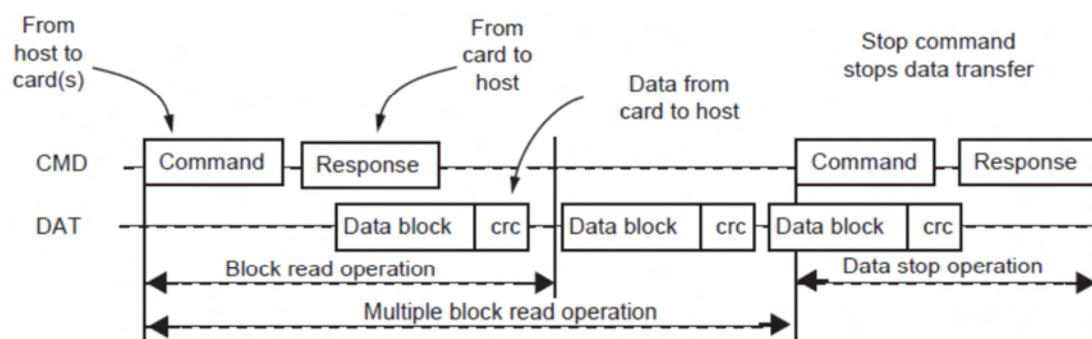
Bloqueo y desbloqueo de la tarjeta. Se puede comprobar mediante el comando SEND_STATUS (CMD13).

Protección del bus. Cada comando transferido por el bus está protegido por bits de CRC, siempre que esté en el modo CRC ON. En el modo CRC OFF, los bits de CRC son ignorados tanto por el transmisor como por el receptor. En la interfaz SPI, aparece por defecto el modo CRC OFF, pero el usuario puede modificarlo mediante el comando CRC_ON_OFF (CMD59). Para ello, antes se debe activar la verificación de CRC mediante ACMD41. Si se detecta un error CRC, la tarjeta devuelve el error en la respuesta R1.

Lectura de datos. El controlador (maestro) envía el comando de petición de lectura a la tarjeta y esta le envía la respuesta de confirmación seguida del bloque de datos con la información contenida a partir de la dirección solicitada.

El modo SPI, soporta lectura de bloques simples (CMD17) y lectura de bloques múltiple (CMD18). En las tarjetas de capacidad estándar, se puede determinar el tamaño de los bloques con el comando SET_BLOCKLEN (CMD16). En las de alta capacidad, el tamaño está fijado en 512 bytes. El comando *stop transmission* (CMD12), detiene la operación de transferencia de datos.

Figura 11. Lectura de múltiples bloques



Fuente: SD CARD ASSOCIATION. SD memory card specifications. Versión 1.01. San Ramon, CA.2001.

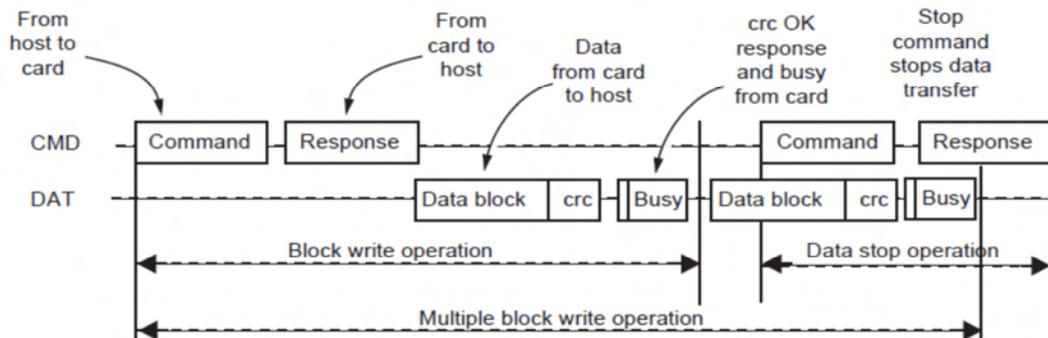
Para leer un bloque de la tarjeta se envía el comando 17 con la dirección de inicio de lectura en los bytes de argumento. La dirección puede tomar cualquier valor comprendido dentro del rango de direcciones válidas de la tarjeta, teniendo en cuenta que todo el bloque leído debe estar en un mismo sector físico. A

continuación la tarjeta envía un byte de respuesta R1 seguido del bloque de datos, que comienza por 0xFE, continúa con los bytes de datos y finaliza con los 2 bytes de CRC que no son empleados. El número de bytes de datos depende del tamaño del bloque que se haya programado mediante el comando 16, y en la lectura puede ir de 1 a 512.

Escritura de datos. El controlador indica a la tarjeta mediante el comando de escritura, que se procederá con la escritura desde una dirección determinada, ésta le responde indicando que está lista y, a continuación, el controlador envía el bloque de datos a escribir. Las operaciones que no requieren intercambio de datos funcionan de igual forma pero sin usar bloques de datos.

El modo SPI también soporta escritura de bloques simples (CMD24) y de bloques múltiples (CMD25). Después de la recepción de un comando de escritura válido, la tarjeta responde con un *token* y espera el bloque de datos enviado por el usuario. Si el bloque es recibido sin errores, es programado. Una vez terminadas las operaciones de programación, el usuario puede comprobar los resultados de ésta mediante el comando *Send_status* (CMD13).

Figura 12. Escritura de múltiples bloques



Fuente: SD CARD ASSOCIATION. SD memory card specifications. Versión 1.01. San Ramon, CA.2001.

Para escribir un bloque de la tarjeta, se envía el comando 24 con la dirección de inicio a partir de la cual van a ser guardados los datos. Si la comunicación es correcta, la tarjeta envía tres respuestas repetidas R1 informando al controlador que ya puede enviar el bloque de datos, el cual debe tener una longitud de 512 bytes (en la escritura sólo son permitidos 512 bytes) más el byte de inicio de bloque de datos y los dos bytes de CRC.

3.4.4.4 Formato de los bloques de datos. Los bloques de datos comienzan siempre con el byte 0xFE, seguido por los bytes de datos y por último los 2 bytes de CRC. El número de bytes de datos depende de la longitud del bloque definida

mediante el comando 16 (CMD16), y ésta puede ir de 1 hasta 512 bytes (por defecto 512). Por tanto, sumando a los bytes de datos, el byte de inicio y los dos bytes de CRC, la longitud total del bloque de datos puede variar entre 4 y 515 bytes.

Formato de datos. Los bytes enviados desde el microcontrolador deben tener el formato que se muestra en la tabla 9 y se describe en la tabla 10.

Tabla 9. Formato de la trama de datos

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
0 1 Comando	Argumento MSB	Argumento	Argumento	Argumento LSB	CRC 1

Fuente: JORDI, Bartolomé. Las tarjetas multiMedia Card y SD. Madrid, 2004.

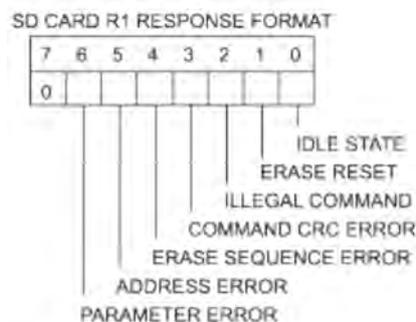
Tabla 10. Descripción de la trama de datos

Posición bit	47	46	[45:40]	[39:8]	[7:1]	0
Cantidad bits	1	1	6	32	7	1
Valor	'0'	'1'	x	x	x	x
Descripción	Bit de inicio	Bit de transmisión	Índice de comando	Argumento	CRC	Bit de parada

Fuente: JORDI, Bartolomé. Las tarjetas multiMedia Card y SD. Madrid, 2004.

3.4.4.5 Respuestas. El intercambio de información entre el dispositivo controlador y la tarjeta de memoria SD, se lleva a cabo mediante el envío de comandos por parte del controlador y de respuestas por parte de la tarjeta. Cuando la SD recibe un comando, esta envía un byte de respuesta en la que cada bit representa una acción.

Figura 13. Bits de estado de respuesta



Fuente: SD CARD ASSOCIATION. SD memory card specifications. Versión 1.01. San Ramon, CA.2001.

Las respuestas de la tarjeta son bloques formados por 1 o 2 bytes, dependiendo del tipo de respuesta, el cual es función del comando, es decir que cada comando tiene asociado un tipo de respuesta. En la figura 13 se muestra el formato de la respuesta R1.

La tabla 11 muestra los bits de estado para los tipos de respuesta; así por ejemplo, *Parameter Error* (PE), *Illegal Command* (IC) e *Idle State* (IS) se encargan de indicar el estado en que se encuentra la transmisión-recepción, así como de indicar los errores.

Tabla 11. Tipos de respuesta en el modo SPI

Bit	R1	R2
0	Idle State	0
1	Erase Reset	WP Erase Skip
2	Illegal Command	Error
3	Com CRC Error	CC Error
4	Erase_Seq_Error	Card ECC Failed
5	Address Error	WP Violation
6	Parameter Error	Erase Parameter
7	0	Out of Range

Fuente: SD CARD ASSOCIATION. SD memory card specifications. Version 1.01. San Ramon, CA.2001.

3.4.5 Sistema de Posicionamiento Global.²⁵ El sistema de posicionamiento global, GPS (Global Positioning System, es un sistema compuesto por una red de satélites, radio-bases terrestres y receptores GPS que permiten casi todas las posibilidades de navegación y posicionamiento en cualquier parte del mundo, a cualquier hora, sin importar las condiciones climatológicas. Este complejo está compuesto por tres sistemas principales a saber, sistema satelital, sistema de control terrestre y sistema de usuario

3.4.5.1 Sistema satelital. Se trata de una constelación de 24 satélites denominada Navstar que gira alrededor de la Tierra en seis planos orbitales a 60° entre sí, con 4 satélites en cada plano. Existen 21 satélites activos y 3 de reserva, por lo cual, en caso de fallo de uno de los satélites, el de reserva ocupa su lugar.

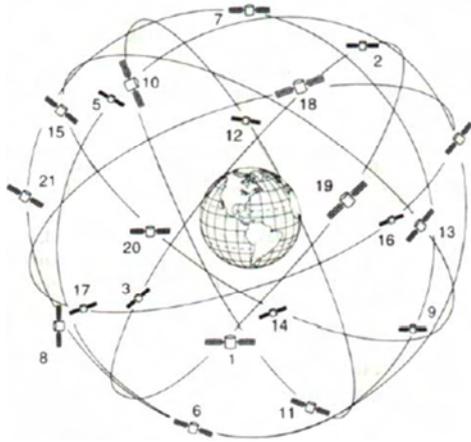
La figura 14 muestra un esquema de las órbitas de los 21 satélites funcionales de la red Navstar. La red fue desarrollada por el Departamento de defensa de los Estados Unidos de América y declarada totalmente operativa desde el 27 de abril de 1995. Al principio, fue creada exclusivamente para fines militares y,

²⁵ BRAIN, Marshall. How GPS receivers works [en línea].
<<http://electronics.howstuffworks.com/gadgets/travel/gps.htm>> [Citado el 15 de Junio de 2011]

posteriormente, dado las aplicaciones tecnológicas, se puso al servicio de la comunidad civil.

Los satélites Navstar no son geosincrónicos, sino que giran en torno a la Tierra en órbitas circulares inclinadas. La elevación promedio de un satélite Navstar es de 20.200 Km sobre la Tierra y tarda aproximadamente 12 horas para completar una revolución.

Figura 14. Constelación de satélites Navstar



Fuente: PALACIOS, Manuel. Sistema de posicionamiento global. Santiago de Chile, 2003.

Los satélites están equipados con un transmisor de señales codificadas de alta frecuencia, un sistema de computación y un reloj atómico de cesio, tan exacto que solamente se atrasa un segundo cada 30 mil años. Las señales de estos satélites proporcionan una posición tridimensional de alta precisión, de forma permanente y en cualquier lugar del mundo, que el receptor GPS decodifica y transforma en latitud, longitud, altitud y rutas marítimas o terrestres y velocidad de vehículos en movimiento como barcos, aviones y automóviles, entre otros.

3.4.5.2 Sistema de control terrestre. También llamado sistema de control de operación, incluye todas las estaciones monitoras terrestres fijas que se encuentran ubicadas en todo el mundo. Las estaciones monitoras no son más que receptores GPS que rastrean los satélites cuando pasan sobre ellas y acumulan datos de telemetría y efemérides de los mismos. Esta información se transmite a una estación de control maestro donde se procesa y determina si la posición real del satélite es igual a la calculada por el GPS. La estación de control maestro recibe datos de las estaciones monitoras en tiempo real, con esa información determina si los satélites sufren cambios de reloj o de efemérides y detecta el incorrecto funcionamiento del equipo.

3.4.5.3 Sistema de usuario. Consiste en todos los receptores GPS y la comunidad de usuarios. Los receptores GPS convierten las señales recibidas de los satélites en estimaciones de posición, velocidad y tiempo. La posición tridimensional (latitud, longitud y altitud), conocida como 3D, requiere cuatro satélites a la vista, mientras que la bidimensional (latitud y longitud), llamada 2D sólo necesita tres. La mayoría de los receptores GPS pueden recibir y procesar información de hasta ocho satélites simultáneamente, aunque la recepción de un mínimo de tres, permite la navegación terrestre o marítima, prácticamente las 24 horas del día en cualquier lugar de la Tierra.

3.4.5.4 Funcionamiento del sistema GPS.²⁶ El sistema GPS se basa, al igual que los sistemas electrónicos antiguos de navegación, en el principio matemático de la triangulación. Por tanto, para calcular la posición de un punto, el receptor GPS determina con exactitud la distancia que lo separa de los satélites.

El sistema GPS funciona determinando cuánto tiempo tarda una señal de radio transmitida de un satélite en llegar al receptor en tierra y, a continuación con ese tiempo, calcula la distancia entre el satélite y el receptor de la estación terrestre. Si un receptor puede determinar con exactitud en qué momento el satélite envía un mensaje de radio y en qué momento lo recibe, puede determinar el tiempo de propagación o retardo.

La distancia existente entre cada satélite y el receptor GPS la calcula el propio receptor realizando operaciones matemáticas. Para hacer este cálculo el receptor GPS multiplica el tiempo de retraso de la señal de control por el valor de la velocidad de la luz. Si la señal ha viajado en línea recta, sin que la haya afectado ninguna interferencia por el camino, el resultado matemático será la distancia exacta que separa al receptor del satélite.

A partir del tiempo de propagación, el receptor puede determinar la distancia entre él y el satélite con la expresión:

$$d = v \times t \tag{3}$$

Donde:

d = distancia entre el satélite y el receptor (metros)

v = velocidad (3×10^8 m/s)

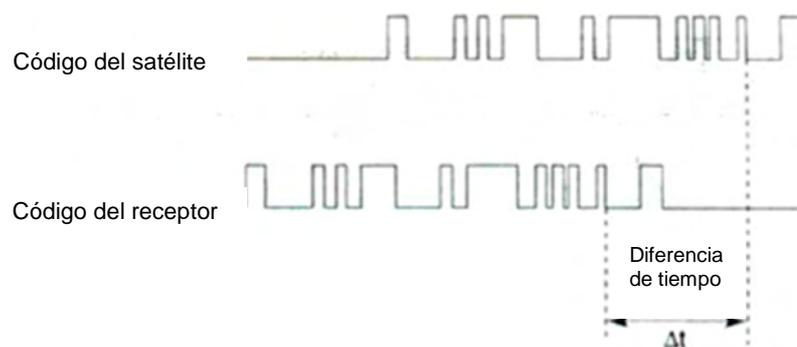
t = tiempo de propagación (segundos)

²⁶ Ibid.,

Para medir el momento a partir del cual el satélite emite la señal y el receptor GPS la recibe, es necesario que tanto el reloj del satélite como el del receptor estén perfectamente sincronizados. El satélite utiliza un reloj atómico de cesio, extremadamente exacto, pero el receptor GPS posee uno de cuarzo, no tan preciso.

Por tal razón, para sincronizar con exactitud el reloj del receptor GPS, el transmisor del satélite y el receptor de la estación terrestre producen códigos idénticos de sincronización pseudoaleatorios exactamente al mismo tiempo, tal como lo muestra la figura 15. Esa señal de control llega al receptor GPS con más retraso que la señal normal de radiofrecuencia, en este caso, el retraso entre ambas señales es igual al tiempo que tarda la señal de radiofrecuencia en viajar del satélite al receptor GPS.

Figura 15. Código pseudoaleatorio de tiempo GPS



Fuente: PALACIOS, Manuel. Sistema de posicionamiento global. Santiago de Chile, 2003.

Debido a que cada satélite transmite en forma continua un código de sincronización preciso, el receptor GPS, después de haber recibido dicho código de sincronización, lo compara con su propio código producido en forma local, para determinar el tiempo de propagación. La diferencia de tiempo multiplicada por la velocidad de la señal de radio determina la distancia al satélite.

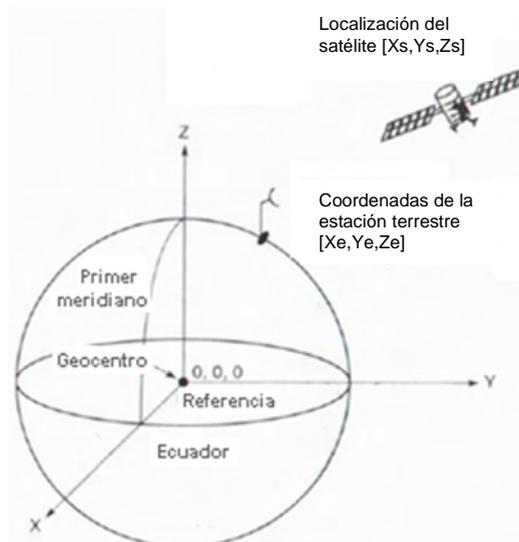
Sin embargo, debido a la inexactitud del reloj de los receptores, se hace necesaria una cuarta medición ya que se produce una discrepancia en todos los rangos a los satélites. De esta manera, el control interno del receptor GPS detecta la discrepancia y atribuye la diferencia a una sincronización imperfecta con la hora universal. Dado que cualquier discrepancia con la hora universal afecta a las cuatro mediciones, el receptor busca un factor de corrección único el cual aplica a las mediciones de tiempo, lo que hace que los rangos coincidan en un solo punto. Dicha corrección permite al reloj del receptor ajustarse nuevamente a la hora universal; una vez que el receptor GPS aplica dicha corrección al resto de sus mediciones, se obtiene un posicionamiento preciso.

Ahora bien, con el código pseudoaleatorio como un pulso confiable para asegurar la medición correcta del tiempo de la señal y la medición adicional como elemento de sincronización con la hora universal, se cuenta con todo lo necesario para medir la distancia de cualquier punto en la tierra a un satélite en el espacio.

Triangulación. El concepto general de triangulación aplicado a los satélites permite ubicar un punto en la superficie de la tierra. Esto se logra mediante una exacta medición de la distancia del receptor respecto al menos tres satélites, lo que permite "triangular" la posición en cualquier parte de la tierra, de esta manera, los satélites en el espacio son puntos de referencia para ubicaciones en la tierra.

Desde el mismo momento que el receptor GPS detecta una señal de radiofrecuencia transmitida por un satélite desde su órbita, se genera una esfera virtual o imaginaria que envuelve al satélite. El propio satélite actúa como centro de la esfera cuya superficie se extiende hasta el punto o lugar donde se encuentra situada la antena del receptor; por tanto, el radio de la esfera es igual a la distancia que separa al satélite del receptor. A partir de ese instante el receptor GPS mide las distancias que lo separan como mínimo de dos satélites más. Para ello calcula el tiempo que tarda cada señal en viajar desde los satélites hasta el punto donde éste se encuentra situado y realiza los correspondientes cálculos.

Figura 16. Sistema de coordenadas GPS para el satélite y la estación receptora

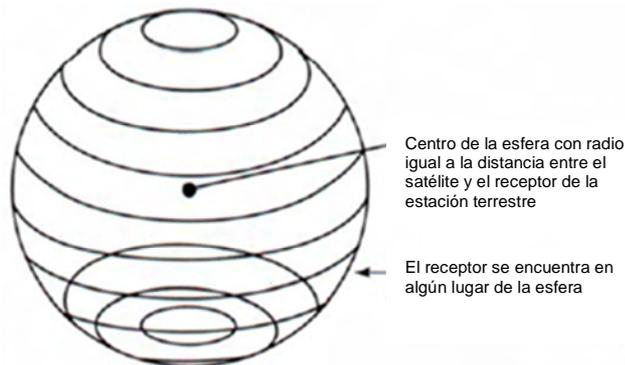


Fuente: PALACIOS, Manuel. Sistema de posicionamiento global. Santiago de Chile, 2003.

Por tal razón, para que un receptor en tierra determine su longitud y latitud, debe recibir señales de tres o más satélites, debido a que se tienen tres incógnitas: X, Y y Z que son las coordenadas que determinan la posición del receptor.

Cada satélite envía al receptor su propio código pseudoaleatorio de sincronización (PNR) y su localización. De acuerdo a la figura anterior, la ubicación de un satélite se describe con un sistema tridimensional de coordenadas en relación con el centro de la Tierra. El centro de la Tierra es el punto de referencia y sus coordenadas son (0,0,0). Así, cada satélite tiene las coordenadas X_s , Y_s , Z_s que determinan su lugar con respecto al geocentro. Sin embargo, las coordenadas de los satélites deben ser actualizadas en forma continua, ya que varían cuando los satélites recorren sus órbitas. El lugar de una estación terrestre tiene como coordenadas tridimensionales X_e , Y_e , Z_e referidas al centro de la Tierra como se muestra en la figura 16. Si una estación receptora conoce la ubicación de un solo satélite y la distancia entre el satélite y esta, se sabe que debe estar en algún lugar de una esfera imaginaria centrada en el satélite y con un radio igual a la distancia del satélite al receptor; esto se muestra en la figura 17.

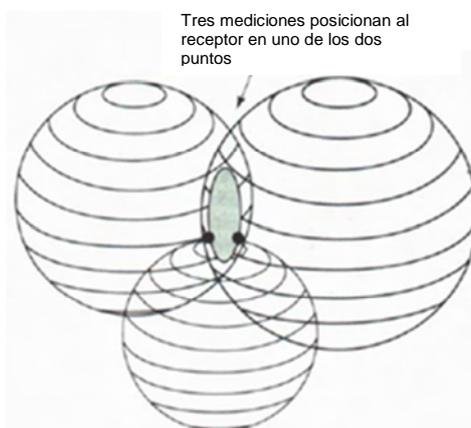
Figura 17. Ubicación de la estación receptora respecto a un satélite



Fuente: PALACIOS, Manuel. Sistema de posicionamiento global. Santiago de Chile, 2003.

Para el caso en el que se conoce la ubicación y la distancia de un segundo y tercer satélite, el receptor puede ubicar su posición limitándola a los dos puntos en los cuales la tercera esfera intercepta la circunferencia que resulta de la intersección de las dos primeras esferas. Es decir, que midiendo la distancia a tres satélites se limita el posicionamiento a sólo dos puntos posibles en el espacio, tal como se muestra en la figura 18.

Figura 18. Ubicación de la estación receptora respecto a tres satélites



Fuente: PALACIOS, Manuel. Sistema de posicionamiento global. Santiago de Chile, 2003.

Para calcular cual de ellos es la posición correcta, se podría efectuar la medición de un cuarto satélite; pero normalmente uno de los dos puntos posibles resulta ser muy improbable por la ubicación demasiado lejana de la superficie terrestre y es descartado sin necesidad de mediciones posteriores.

En conclusión, los receptores GPS pueden determinar un punto correcto de ubicación gracias a la red que forman los satélites y que envuelve la esfera terrestre, de forma que desde cualquier punto teórico de su superficie se puede sintonizar al menos cuatro satélites de manera simultánea. En la práctica, casi todos los GPS del mercado actualmente, acceden a más de 6, y hasta a 12, satélites simultáneamente.

Los receptores GPS tienen una gran variedad de aplicaciones, entre las más importantes se encuentran: el posicionamiento, navegación, diseminación de la hora, cartografía, topografía, seguridad pública entre otras que aún están en vías de desarrollo. La aplicación más frecuente de los receptores GPS es la determinación de la posición o localización. El GPS es el primer sistema de posicionamiento que ofrece una localización altamente precisa en cualquier punto.

3.4.5.5 Estándar NMEA 0183.²⁷ La información que entrega el receptor GPS corresponde a códigos NMEA a través de cadenas de datos denominadas "sentencias", las cuales contienen paquetes de datos entre las que se encuentran posición, velocidad y tiempo.

²⁷ NATIONAL MARINE ELECTRONICS ASSOCIATION. NMEA Protocol [en línea]. <<http://www.nmea.org>> [Citado el 16 de Junio de 2011]

Éste estándar fue desarrollado por el National Marine Electronics Association (NMEA) que corresponde también al nombre del estándar. El estándar NMEA 0183 define la interfaz eléctrica y el protocolo de datos para el intercambio digital de información entre diferentes equipos electrónicos marítimos. La información se entrega por el puerto serie del dispositivo a una frecuencia conocida, que por lo regular es de 1 Hz, lo que significa que se puede obtener la ubicación geográfica actualizada cada 1 segundo.

Características estándar NMEA. Bajo la norma NMEA 0183, todos los caracteres usados son texto ASCII imprimible; emplea una interfaz serial asíncrona donde los datos se envían a 4800 baudios, 8 bits de datos, 1 bit de parada y sin paridad. El estándar permite un único "talker", y varios "listeners" en un circuito.

El cableado recomendado para las interconexiones es un par trenzado y cubierto, con la cobertura conectada a tierra sólo en el "talker". Las normas no especifican el uso de ningún conector en particular. El estándar especifica que la salida del "talker" puede ser RS-232, o de un buffer TTL, capaz de entregar 10 mA. a 4 V.

Formato general de las sentencias.²⁸ En la mayoría de los módulos receptores GPS, la cadena que se transmite es de un mensaje por segundo (1PPS) compuesto por cinco sentencias NMEA las cuales son descritas en el Anexo A. Éstas son:

- *GGA - Global Positioning System Fix Data*
- *GSA - GPS DOP and Active Satellites*
- *GSV - GPS Satellites in View*
- *VTG - Course Over Ground and Ground Speed*
- *RMC - Recommended Minimum Specific GPS Data*

La comunicación entre el receptor GPS y el PIC se realiza bajo el protocolo RS232. Así, lo que se recibe a través de la línea RX del PIC corresponde a una cadena de mensajes en formato NMEA del receptor GPS. Todos los datos se transmiten en forma de sentencias en caracteres ASCII; cada sentencia inicia con el símbolo "\$", seguido por dos caracteres "talker ID", que identifican el equipo, que en el caso de dispositivos GPS se conforma de las letras GP, y finalmente tres caracteres "ID sentencia", que identifican el tipo de sentencia que se está enviando.

En cuanto a los datos, se presentan de campos separados por comas, terminando con un checksum optativo, y un retorno o *line feed*. El campo optativo *checksum* consiste en el símbolo "*" y dos dígitos en formato hexadecimal que representan la

²⁸ BADDELEY, Glenn. GPS NMEA sentence information [en línea]. <<http://home.pacific.net.au/~gnb/gps/nmea.html>> [Citado el 20 de Junio de 2011]

OR exclusiva de todos los caracteres que hay en medio, pero sin incluir el "\$" y el "*".²⁹ En la tabla 12 se muestra un ejemplo de sentencia GGA.

Tabla 12. Formato de la sentencia GGA

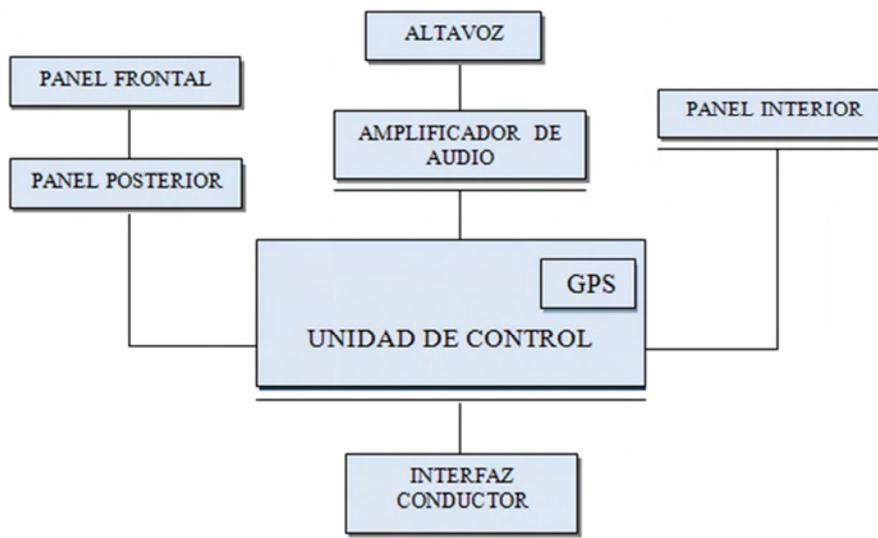
Sentencia GGA (<i>Global Positioning System Fix Data</i>)	
Formato:	\$GPGGA,hhmmss.ss,lll.ll,a,mmmm.mm,b,x,yy,z.z,c.c,M,v.v,M,d,d,wwww*hh
hhmmss.ss	Hora UTM a la que tuvo lugar la toma de datos
lll.ll	Latitud en grados, minutos y segundos
a	Tipo de latitud. Norte (N) ó Sur (S)
mmmm.mm	Longitud en grados, minutos y segundos
b	Tipo de longitud. Este (E) ó Oeste (W)
x	Calidad de la señal GPS. (0 = no válida; 1= corrección GPS; 2 = DGPS)
yy	Número de satélites en uso
z.z	HDOP
c.c	Altitud sobre el nivel del mar
M	Unidad de medida de la altitud. Metros (m) por defecto
v.v	Distancia entre el elipsoide WGS-84 y el nivel del mar
M	Unidad de medida de la distancia. Metros (m) por defecto
d,d	Tiempo desde la última actualización DGPS (segundos)
wwww	Identificación de la estación DGPS
*hh	Palabra de comprobación o checksum

Fuente: Autor.

²⁹ MAGELLAN CORPORATION. Data transmission protocol specification [en línea]. <http://www.magellangps.com> [Citado el 16 de Julio de 2011]

4. DESARROLLO

Figura 19. Diagrama de bloques del sistema de información al usuario



Fuente: Autor.

4.1 DESCRIPCIÓN GENERAL

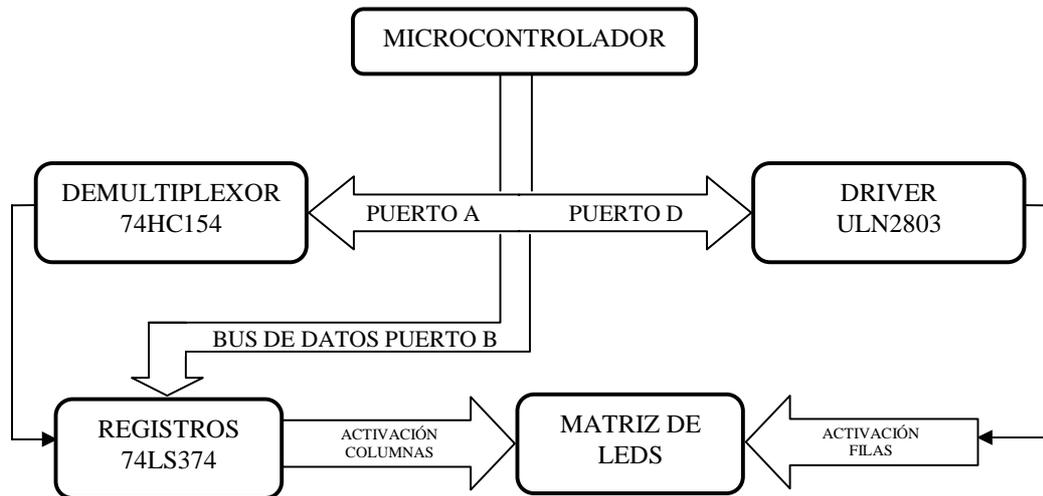
El sistema de información al usuario consta de tres partes funcionales: el módulo de señalización visual, el módulo de señalización acústica y la interfaz del conductor. En este sentido, las siguientes secciones están organizadas de acuerdo con esta división. El diagrama de la figura 19, indica la estructura general del sistema. Se dispone de una unidad de control desarrollada en torno al microcontrolador PIC18F2550, el cual gestiona las operaciones de voz y datos del sistema; igualmente, de tres paneles electrónicos de matriz de leds que indican de manera visual la información concerniente a próxima parada e itinerarios de ruta y por último la interfaz de usuario que corresponde a la unidad de mando del conductor.

4.1.1 Señalización visual.

4.1.1.1 Panel electrónico de matriz de leds. El diagrama de bloques de la figura 20 representa el circuito de control de un panel electrónico de matriz de leds, el cual se constituye por el microcontrolador PIC16F887 que incorpora una memoria de programa tipo flash de 8KB, cinco puertos de entrada/salida de 8 bits, 3 de los

cuales (A, B y D), son configurados como salidas; este microcontrolador cuenta con recursos de comunicación tales como SPI e I²C.

Figura 20. Diagrama de bloques del circuito de control de la matriz de leds



Fuente: Autor.

La técnica de multiplexación se lleva a cabo gracias a las propiedades de carga y retención de los registros de memoria empleados. La tabla 13 corresponde a la tabla de verdad del registro 74374 que corresponde a un flip flop tipo D. En esta se puede observar que los buffers de salida son habilitados cuando la entrada de habilitación de salida OE tiene un nivel bajo, lo que permite la transferencia de las salidas externas Q₀ a Q₇. Si cualquiera de las entradas que habilitan las salidas tiene un nivel alto, los buffers son deshabilitados y las salidas se encuentran entonces en el estado de alta impedancia. La entrada OE no tiene ningún efecto sobre la operación de carga de datos; ésta solo se utiliza para controlar el hecho de si las salidas de los registros aparecerán o no en las salidas externas.

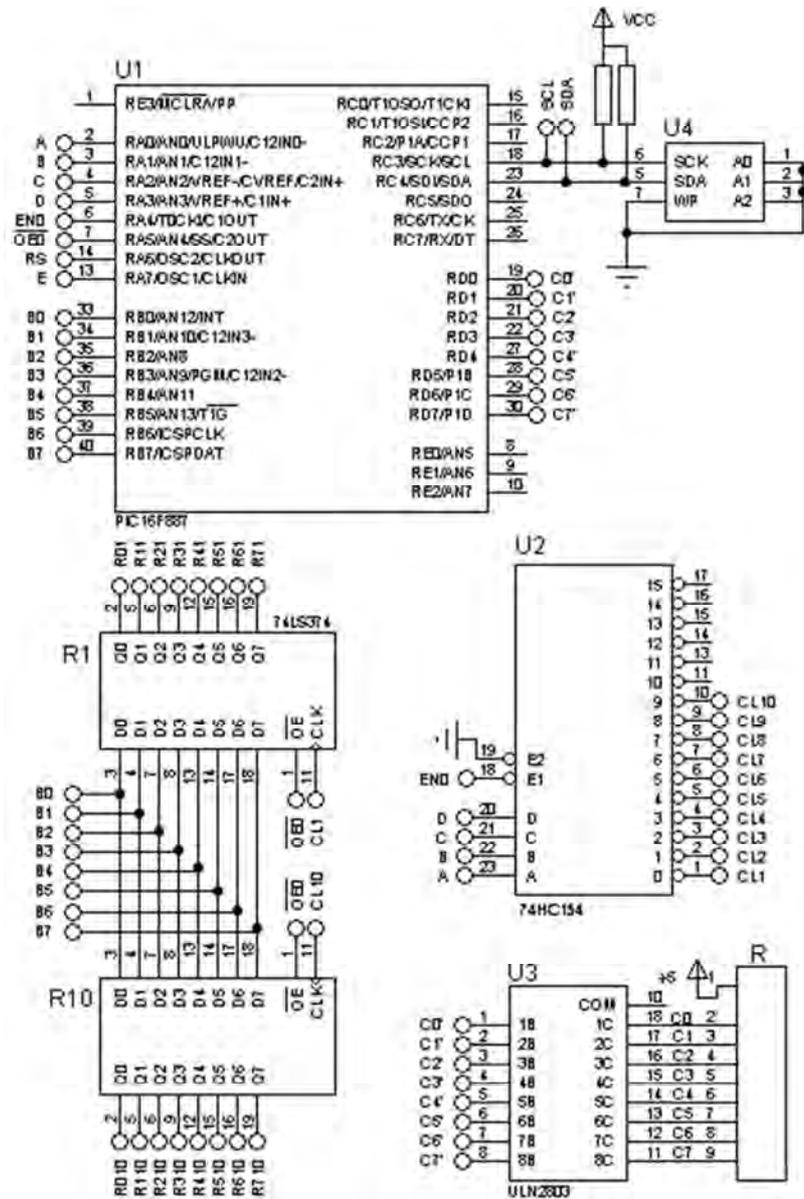
Tabla 13. Tabla de verdad del registro CMOS 74374

OPERATING MODES	INPUTS			INTERNAL FLIP-FLOPS	OUTPUTS Q ₀ to Q ₇
	\overline{OE}	CP	D _n		
load and read register	L	↑	l	L	L
	L	↑	h	H	H
load register and disable outputs	H	↑	l	L	Z
	H	↑	h	H	Z

Fuente: PHILIPS SEMICONDUCTORS.

Esta lógica admite dos modos de operación, la primera, de carga, donde los datos en las entradas desde D₀ a D₇ son transferidos a los flip flops internos cuando ocurre un disparo por flanco positivo (TPP) y la segunda, de retención, donde los datos del registro no cambian aun cuando ocurre la TPP de CP.

Figura 21. Diagrama esquemático del circuito de control matriz de leds



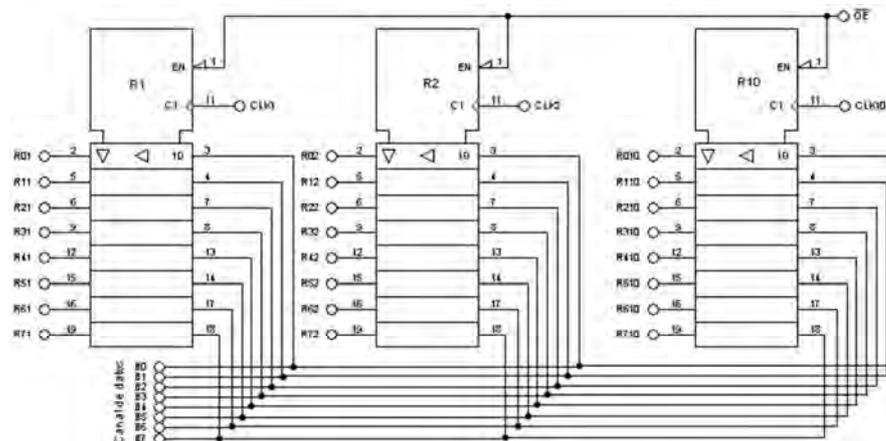
Fuente: Autor.

Las columnas de la matriz de leds son activadas por flip-flops 74374, los cuales son habilitados selectivamente mediante el demultiplexor 74154, cuyo control se realiza por el puerto A. Las entradas de los flip-flops comparten los 8 bits del bus de datos del puerto B, el cual envía la codificación de los caracteres. La activación de las filas se realiza mediante el puerto D, en cuyas salidas se dispone del driver ULN2803, que realiza el control de corriente de paso a tierra.

La figura 21 muestra el diagrama esquemático del circuito de control, en éste se indica la conexión del microcontrolador con los circuitos integrados mencionados anteriormente y la memoria eeprom serie 24LC256, utilizada para el almacenamiento de los itinerarios de ruta.

4.1.1.2 Canal de datos. En esta configuración, el canal de datos consta de ocho líneas etiquetadas B₀- B₇. La figura 22 muestra el canal de datos empleado para la transferencia de los datos a los flip-flops tri-estado. Cada flip-flop cuenta con una entrada de habilitación OE (también EN) conectada a una sola entrada OE activa en bajo. En este caso, para un nivel lógico bajo, (OE=0) todos los registros se encuentran habilitados.

Figura 22. Canal para la transferencia de datos

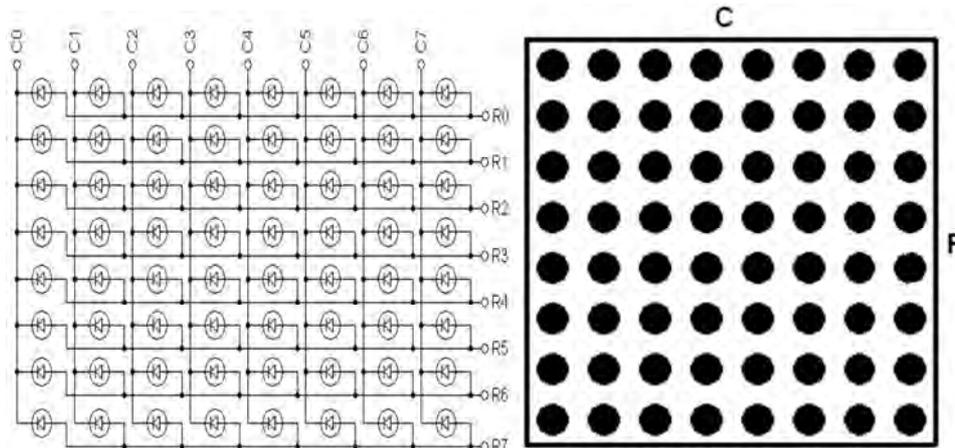


Fuente: Autor

Las entradas de cada registro están conectadas a la misma línea del canal, por lo tanto, los niveles en la línea pueden ser transferidos a cada uno de los registros según sea la señal de reloj generada por la unidad de control. La operación de transferencia de datos se realiza transfiriendo el contenido, en paralelo, sobre el canal de datos a uno de los registros a la vez, a través de la aplicación adecuada de niveles lógicos a las entradas de la señal de reloj de cada registro.

4.1.1.3 Técnica de multiplexación. Con el objetivo de brindar claridad en la forma como se implementó la técnica de multiplexación, se toma como referencia la matriz individual de 8 filas por 8 columnas de la figura 23.

Figura 23. Diagrama circuital y esquemático de una matriz individual 8x8



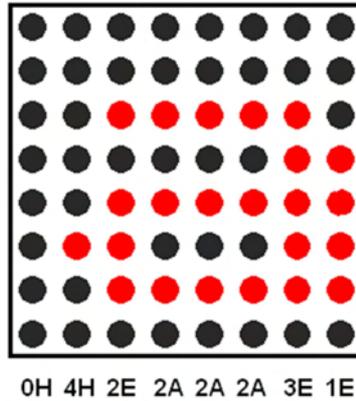
Fuente: Autor.

Las filas están en disposición ánodo común y las columnas en cátodo común, lo que significa que para la activación de cualquier punto (led) de la matriz, se requiere de un nivel alto en la fila y un nivel bajo en la columna correspondiente.

La multiplexación de columnas se realiza mediante un barrido en lógica directa, mientras se carga en el canal de datos los bits correspondientes a la columna que va a ser activada. El direccionamiento de la señal de reloj se realiza a través del demultiplexor 74HC154 el cual genera también un pulso de activación del CLK del registro respectivo. El pulso hace que los datos se posicionen en la salida del registro seleccionado.

Tomando como ejemplo la letra “a”, representada en la matriz individual de 8x8, de la figura 24, se puede inferir que corresponde a la representación de la cadena de valores de cada columna en formato hexadecimal: 0H, 4H, 2E, 2A, 2A, 3E y 1E. El barrido se realiza, en este caso, columna a columna después de la activación de las filas. De esta forma, para visualizar sobre la matriz el punto correspondiente a la coordenada (R₅,C₁), se posiciona sobre las filas, el dato 4H (en este caso, [R₇,R₆, R₅,R₄,R₃,R₂,R₁,R₀] es igual a [00000100]) y posteriormente se activa la columna C₁. Así, repitiendo el proceso varias veces en una unidad de tiempo (del orden 10⁻³s), se crea la impresión de una imagen estática y fija.

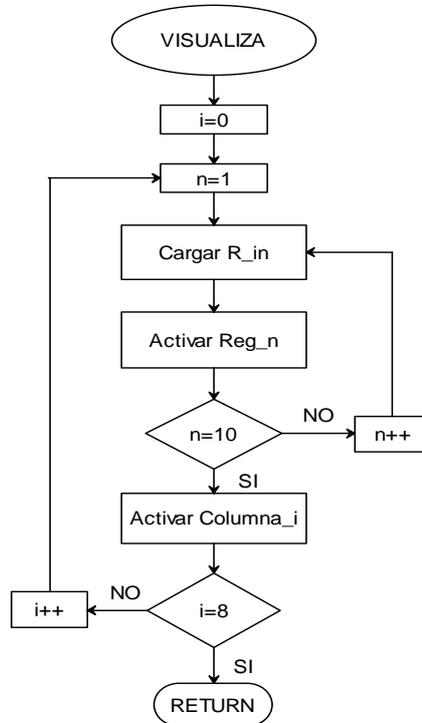
Figura 24. Visualización de un caracter en una matriz individual 8x8



Fuente: Autor.

4.1.1.4 Rutina de visualización. El diagrama de flujo de la figura 25 representa el algoritmo seguido para cargar los datos a visualizarse en una matriz de 8x80. Se inicializa los contadores i y n en 0 y 1 respectivamente; n corresponde al contador del número de columna y i representa el número de fila de una matriz individual 8x8.

Figura 25. Diagrama de flujo utilizado para la visualización

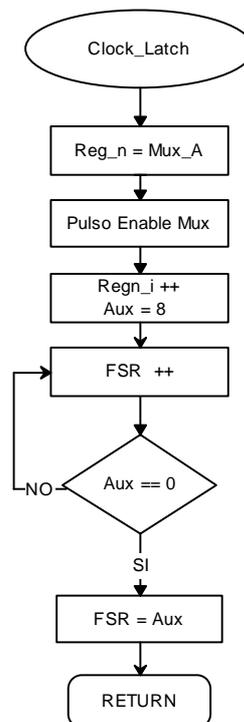


Fuente: Autor.

Se introduce la posición del registro direccionado, comenzando con la primera posición que aloja el valor del registro R_{in} guardado en la memoria eeprom serial. El envío de datos por el canal se realiza mediante direccionamiento indirecto, donde FSR es el registro de selección en el que se introduce la dirección del registro direccionado. Éste registro actúa como puntero ya que su valor corresponde a la dirección de una posición de memoria. Por otro lado, mediante el registro INDF se accede a los datos apuntados por el registro FSR y se los saca por el puerto B hacia el canal de datos.

La subrutina *Clock_Latch* corresponde a un lazo limitado por el contador Reg_n , para activar mediante el demux, los diez flip-flops tri-estado. En ésta se envía el valor de Reg_n a las entradas A0, A1, A2 y A3 del demux y se genera un pulso de reloj para habilitar un flip-flop a la vez.

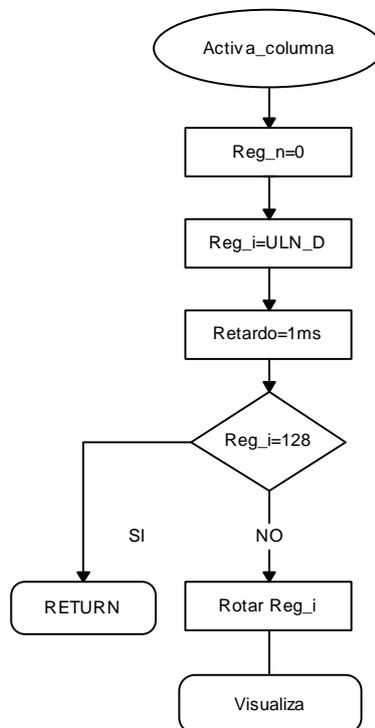
Figura 26. Diagrama de flujo de la subrutina *Clock_Latch*



Fuente: Autor.

Terminada la activación de los 10 flip-flops, se tiene 8 bits de datos en la salida de cada uno. En este punto se reinicia el contador Reg_n y se procede a la activación de las columnas realizando un retorno para cargar los próximos datos hasta que todas las columnas han sido multiplexadas. La multiplexación se realiza por el puerto D por el cual se activa las columnas secuencialmente. Cuando la multiplexación termina se inicia un nuevo ciclo de visualización.

Figura 27. Diagrama de flujo de la subrutina Activa_columna



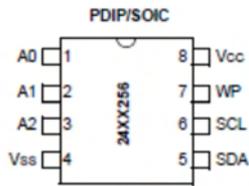
Fuente: Autor.

Se evidencia en la práctica que utilizando tiempos superiores a 5ms para el encendido, repercute negativamente en la correcta visualización de los mensajes, ya que se tiene un incremento significativo del parpadeo. Por lo tanto, el tiempo de retardo, inferior a los 5 ms, logra una visualización adecuada, esto es, sin parpadeo y con los leds correctamente iluminados.

4.1.1.5 Rutina memoria eeprom. La memoria utilizada para almacenar los itinerarios de ruta corresponde a la memoria 24LC256 la cual permite almacenar hasta 32 Kbyte. Se trata de una memoria eeprom serie con interfaz I2C fabricada por Microchip, en un encapsulado de 8 pines mostrado en la figura 28.

La paginación de la memoria se constituye de dos palabras de 8 bits, lo que significa que cada posición de memoria se identifica por 2 bytes de dirección. Las 32 k posiciones de memoria están divididas en 128 bloques o páginas de 256 posiciones cada una. Cada posición de memoria está identificada por el byte de dirección alta que identifica el número de página y el byte de dirección baja que identifica la posición de esa palabra dentro de la página.

Figura 28. Patillaje de la memoria serie 24LC256



Pin	Descripción
Vcc	Fuente alimentación
Vss	Masa
SDA	Línea de datos
SCL	Línea de reloj
A ₀₋₂	Líneas de dirección
WP	Protección escritura

Fuente: MICROCHIP TECHNOLOGY INC.

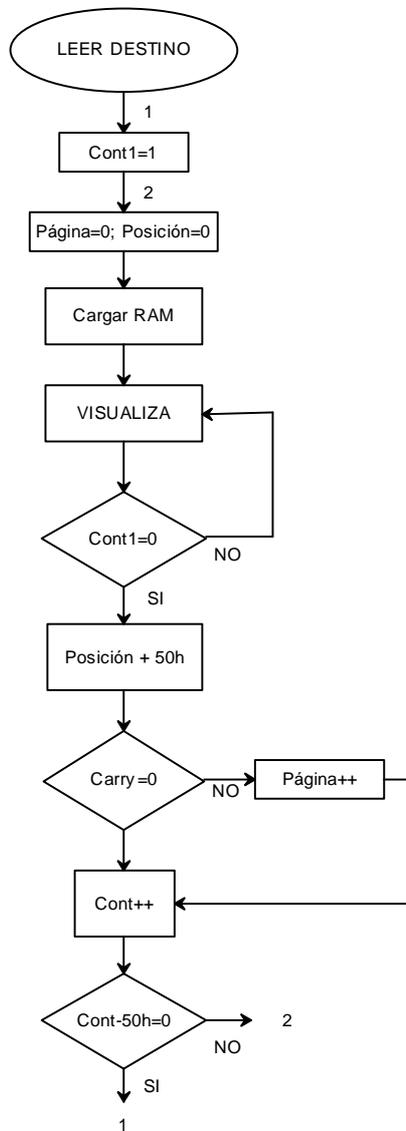
En la memoria eeprom están almacenados los 80 bytes correspondientes a una imagen completa de la matriz, para el caso de la matriz 8x80. El cálculo de la capacidad de la memoria se realizó con base en los lineamientos del componente operativo del SETP de Pasto. En este sentido, el SETP se compone de 7 rutas estratégicas y 16 complementarias; sobre cada ruta se visualiza 7 destinos de ida y vuelta, para un total de 14 destinos por ruta. Puesto que cada destino corresponde a una imagen completa de la matriz, la capacidad de la memoria eeprom es la adecuada para satisfacer dicho requerimiento.

El diagrama de flujo de la figura 29 muestra el algoritmo seguido para acceder a cada destino de ruta. El proceso inicia cargando la RAM del microcontrolador con los 80 bytes de la imagen a visualizarse en la matriz de leds. Paso seguido se procede a la visualización del destino de ruta, la cual se realiza mediante la subrutina *Visualiza* descrita anteriormente. El programa lee dentro de una misma página hasta alcanzar la posición límite que corresponde a 256 bytes luego de sucesivas sumatorias del contador y 80; para pasar a otra página se incrementa el contador de pagina.

La subrutina *Cargar_Ram* accede a la memoria eeprom externa y guarda en la memoria RAM interna del microcontrolador, los 80 bytes de una imagen completa de la matriz que corresponde a un destino específico de ruta. El primer byte se guarda a partir de la dirección de propósito general 20h del banco cero del microcontrolador y los bytes siguientes, en posiciones consecutivas hasta la dirección 6F donde finaliza la lectura de la eeprom; los datos son guardados en la RAM mediante direccionamiento indirecto a través de los registros FSR e INDF.

Las subrutinas siguientes permiten realizar las tareas de manejo de la memoria eeprom, que transmite y recibe la información vía serie a través de un bus I2C. Se inicia la lectura a partir de la posición de memoria fijada por los registros *Página* y *Posición*; el primer registro indica el número del bloque o página de memoria a escribir, mientras que el segundo indica la posición a escribir dentro del bloque

Figura 29. Diagrama de flujo para la lectura de itinerarios de ruta

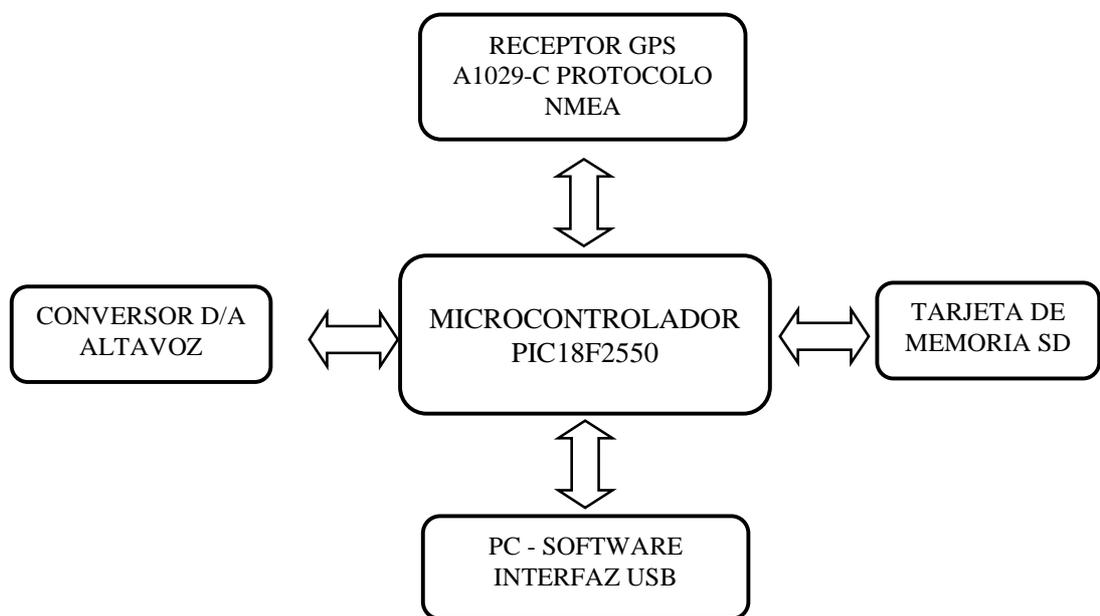


Fuente: Autor.

En la inicialización, no se lee dato alguno, sólo se envía la condición de start y la dirección del esclavo contenida en *Dir_escritura*, a partir de la dirección apuntada por los registros *Página* y *Posición*. Mediante la dirección de lectura *Dir_lectura*, se indica a la memoria que se procede a la lectura. En el registro de trabajo W se carga el valor del dato leído y se realiza el retorno de la subrutina con dicho valor, el cual se introduce en la RAM del microcontrolador.

4.1.2 Señalización acústica. El módulo de señalización acústica se representa mediante el diagrama de bloques de la figura 30. En éste se detalla que el control del módulo está desarrollado en torno al microcontrolador PIC18F2550, el cual recibe del receptor GPS A1029-C las coordenadas en formato NMEA, y luego de un proceso de conversión, procede al cálculo de la distancia entre las coordenadas actuales y las almacenadas en la tarjeta de memoria SD. Si las coordenadas a comparar se encuentran a una distancia no superior a doscientos metros, se procede al anuncio de la próxima parada de forma audible.

Figura 30. Diagrama de bloques del componente de señalización acústica



Fuente: Autor.

El sistema cuenta con dos modos de operación, a saber: modo programación y modo de operación independiente.

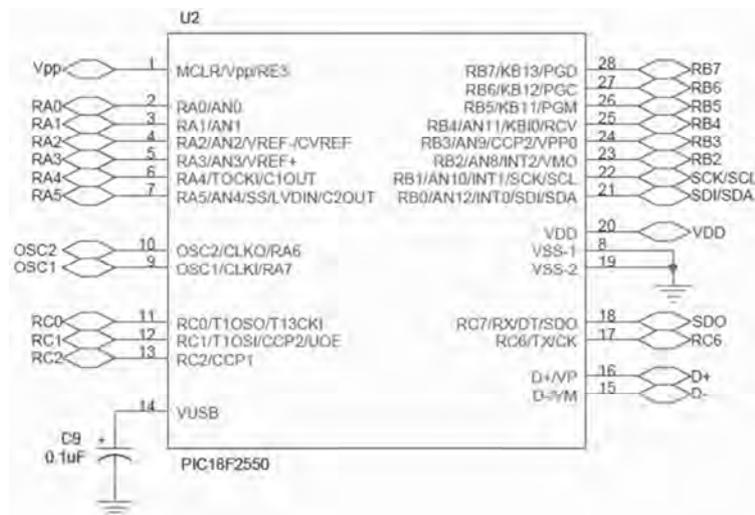
El modo programación se implementa con el fin de guardar previamente las coordenadas correspondientes a cada paradero. En este caso, el software dispuesto para tal fin, permite la visualización de las coordenadas recibidas por el receptor GPS. Una vez el receptor GPS se enlaza con el servicio GPS, las coordenadas de latitud y longitud pueden ser guardadas para ser posteriormente asociadas, por el usuario, al archivo de audio de extensión wav correspondiente. Finalmente, se realiza la programación del sistema, mediante el envío vía USB, de los datos obtenidos a la unidad de control del módulo. La tarjeta de memoria SD permite guardar como datos binarios los archivos de audio para la reproducción de los mensajes de voz y las coordenadas a comparar, los cuales son enviados a la

unidad de control en bloques de 64 bits desde el PC a través del puerto USB. El microcontrolador tiene acceso a la memoria mediante las funciones de inicialización y lectura/escritura de sectores.

En el modo operación independiente, la unidad de control toma lectura, vía serie, de los datos del receptor GPS y luego de interpretar la trama NMEA extrae las sentencias correspondientes a la GPGGA, la cual contiene las coordenadas de posición: latitud y longitud. Cuando el sistema se encuentra en funcionamiento, en este caso el automotor se encuentra prestando el servicio, la unidad de control del módulo realiza la comparación de las coordenadas y comprueba que la trama recepcionada tiene correspondencia con la esperada. Para realizar esta tarea, el microcontrolador lee de la tarjeta de memoria SD las coordenadas almacenadas con anterioridad, si la coordenada recibida se encuentra en el margen de distancia dispuesto, entonces ordena la reproducción del mensaje de voz que contiene la próxima parada.

La figura 31 muestra la designación de entradas y salidas del PIC18F2550 sobre el cual se desarrolla el control del sistema del módulo.

Figura 31. Diagrama de entradas y salidas del PIC18F2550 como unidad de control

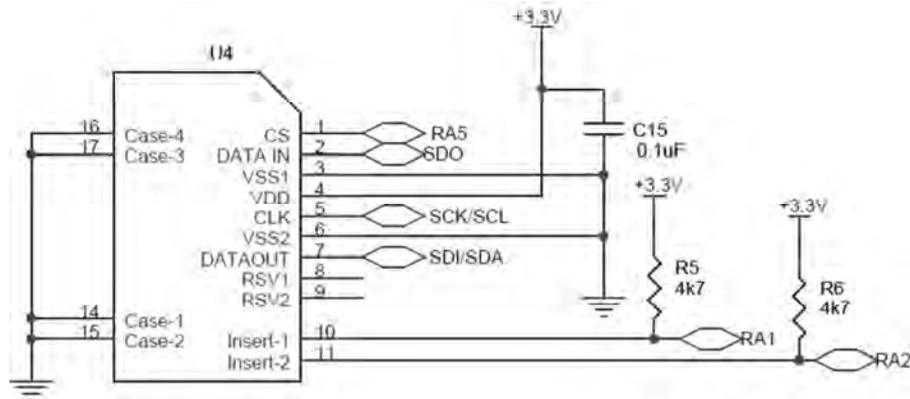


Fuente: Autor.

En secciones siguientes se expone en detalle los bloques funcionales implicados en la reproducción del audio, la cual se lleva a cabo en tres etapas, la primera corresponde a la lectura del archivo de audio grabado en la tarjeta de memoria SD, la segunda es la conversión digital-analógica y la última etapa es la adecuación de la señal de salida donde se amplifica.

4.1.2.1 Bloque funcional tarjeta de memoria SD. La tarjeta de memoria utilizada corresponde a un tipo SD fabricada por SanDisk de 2Gb de capacidad. El diagrama de pines del conector y la conexión con el microcontrolador se muestra en la figura 32.

Figura 32. Diagrama de pines del conector de la tarjeta de memoria SD



Fuente: Autor.

La información audible se almacena en la tarjeta de memoria SD a partir del sector cero, donde se guarda el total de los mensajes de voz grabados, el resto del sector cero es reservado. Desde el sector 1 hasta el sector 566 se encuentra almacenada la información concerniente a rutas, destinos y paradas. A partir del sector 567 se almacena el índice de los mensajes de voz, cada uno con 32 bytes de información, éste contiene información de la dirección del sector donde se encuentra grabado el archivo wav, la cantidad de bytes que ocupa y las coordenadas en longitud y latitud asociadas al archivo de audio.

Tabla 14. Detalles de los 32 bytes de información de un archivo de audio

Byte	Descripción
0-3	4 bytes de dirección del sector donde inicia el archivo de voz
4-7	4 bytes de longitud de cada archivo wav
8-15	8 bytes de latitud
16	1 byte de latitud Norte o Sur
17-24	8 bytes de longitud
25	1 byte de longitud Oeste o Este
26	1 byte frecuencia de muestreo
27-32	5 bytes reservados

Fuente: Autor.

Las operaciones de lectura y escritura se realizan enviando el comando correspondiente junto a la dirección del primer byte del bloque. El largo del bloque fijado en el comando CMD16 es 512, en este caso no está permitido realizar operaciones en dos sectores a la vez.

A continuación se describe las características de los archivos de audio, las cuales son determinantes para el programa (*firmware*) en la decodificación de los mensajes de voz. El formato utilizado en los archivos de audio corresponde al formato wav, con información mono, a una frecuencia de muestreo de 8Khz y a 8 bits de resolución.

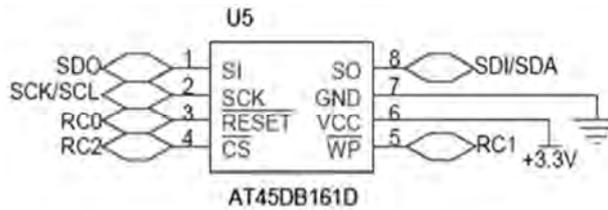
La elección de este formato está motivada por la eficiencia en la decodificación en sistemas desarrollados en torno a microcontroladores. En cuanto a la elección de la frecuencia de muestreo se tuvo en cuenta el teorema de Nyquist, en este caso la frecuencia de muestreo de 8KHz resulta de la premisa que la frecuencia máxima de la señal de voz con calidad telefónica es de 4KHz. Lógicamente a frecuencias de muestreo superiores tales como 16KHz e incluso 40KHz (calidad HI-FI) se logra alcanzar mayor fidelidad. Sin embargo, pruebas realizadas dieron como resultado que a frecuencias superiores, los continuos pasos de lectura necesarios, a nivel firmware, para la reproducción del audio, prolongan el proceso de lectura repercutiendo en la velocidad de reproducción, la cual se torna lenta.

Por otro lado, con el fin de delimitar las características de los archivos de audio aceptados por el sistema, se limita a aquellos archivos wav cuya resolución corresponde a una representación de 8 bits por muestra (256 niveles de cuantificación). En este caso, esta característica se define en el grabador de voz empleado, que para efectos de las pruebas realizadas corresponde al grabador y editor de audio digital *Gold Wave* de la firma Goldwave Inc.

En la operación de lectura de la tarjeta de memoria SD, es necesario imponer cierto tiempo de espera definido por el protocolo de acceso SPI. Esto hace que el proceso de lectura se lleve a cabo de forma lenta y no permite una transferencia directa de datos al convertidor D/A. Por esta razón, se introdujo en el diseño la memoria RAM auxiliar de la figura 33.

De esta manera, cuando un mensaje de voz va a ser reproducido, se realiza la lectura de la tarjeta de memoria SD y se guarda la información obtenida en el buffer de la RAM auxiliar para posteriormente convertir los datos binarios a una señal analógica mediante el convertidor D/A y finalmente proceder con la reproducción a 8 Kb/s.

Figura 33. Diagrama de pines de la memoria RAM



Fuente: ATMEL CORPORATION.

A continuación se describe las principales rutinas empleadas en la lectura y escritura de bloques de la tarjeta de memoria SD en el modo SPI cuyo diagrama de conexiones se muestra en la figura 32. El módulo MSSP, empleado para realizar la transferencia de datos con otros dispositivos según los protocolos SPI e I2C, del PIC18F2550 tiene tres registros de control asociados: el registro de estado (SSPSTAT) y dos registros de control (SSPCON1 y SSPCON2). El uso de estos registros y sus bits de configuración individuales difieren significativamente en función de si el módulo MSSP se opera en el modo SPI o I2C.

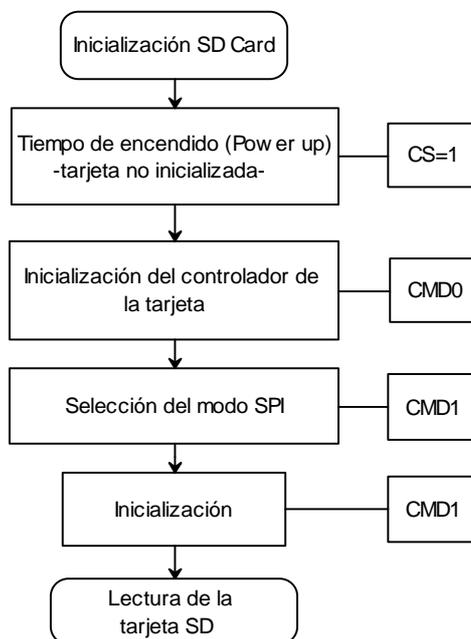
La rutina de inicialización de la tarjeta de memoria SD comienza con la configuración de los pines de I/O del microcontrolador, empleados para el modo de operación SPI.

En particular, en el modo SPI, el registro de SSPCON1 del PIC18F2550 debe ser configurado en modo maestro, con polaridad del reloj baja y el punto de muestreo para la entrada SDI debe estar centrado. El pre escale seleccionado es de 1:64, de esta forma, la frecuencia de operación es controlada por la preescala, que divide el ciclo de reloj del procesador principal para generar la señal de reloj SPI.

Antes de utilizar la tarjeta en aplicaciones de almacenamiento masivo, se debe completar la secuencia que se resume en el diagrama de la figura 34.

Los pasos de la secuencia de inicialización comprenden insertar la tarjeta en el conector (el circuito debe estar alimentado), mantener en alto el pin CS (CS=1 tarjeta no seleccionada), inicialización del controlador de la tarjeta para lo cual se debe proporcionar los pulsos de reloj necesarios antes que la tarjeta sea capaz de recibir comandos (como mínimo 74 pulsos), seleccionar el modo de operación que en este caso corresponde al SPI, enviar la secuencia de reset y finalmente enviar el comando de inicialización.

Figura 34. Secuencia de inicialización de la tarjeta de memoria SD

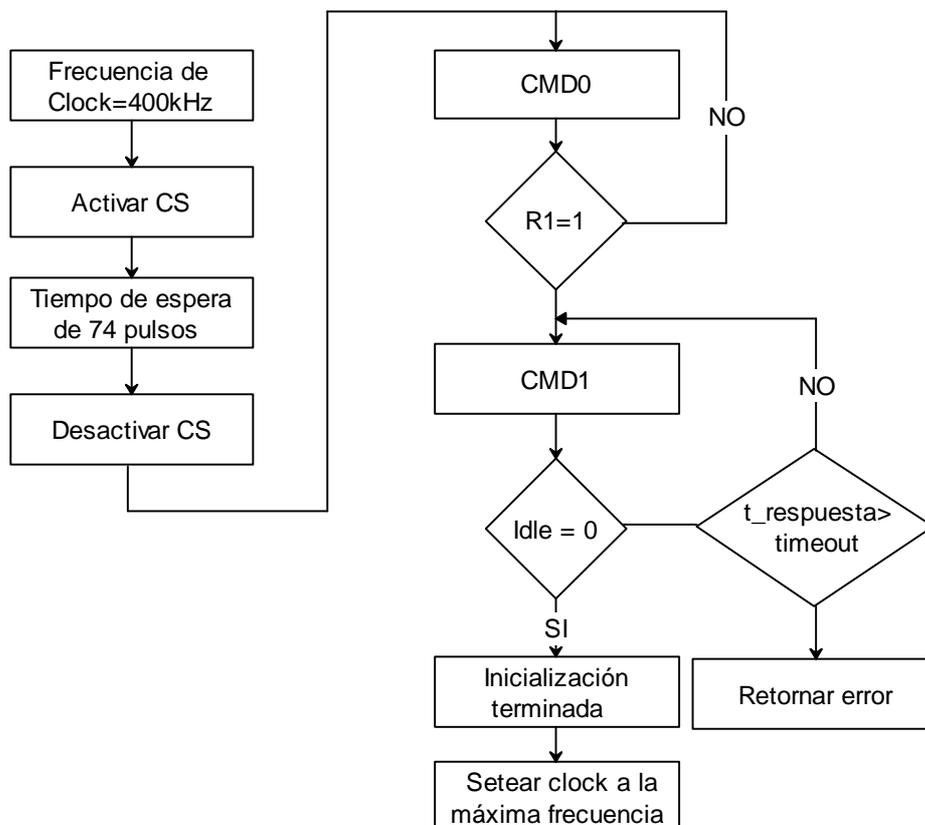


Fuente: Autor.

La secuencia de inicialización está implementada en la función *Start_up_SD*, véase el Anexo B, la cual permite ejecutar, posteriormente a la inicialización, las operaciones de lectura y escritura pertinentes. Básicamente en ésta se envía primero el comando de reset *Comando_SD(0x40)* hasta que la respuesta R1 es 1, que indica que la memoria se encuentra en modo *idle*. Una vez reseteada se envía el comando CMD1 *Comando_SD(0x41)* hasta que el bit idle de la respuesta R1 es cero lo que indica que la tarjeta ha completado su inicialización y se encuentra lista para recibir nuevos comandos. El diagrama de flujo del proceso se muestra en la figura 35.

Según el fabricante, es preciso reducir la velocidad del reloj por debajo de 400KHz como medida de protección, después del encendido y mientras la tarjeta SD se inicializa correctamente; esto es sólo un arreglo temporal, ya que después de enviar los primeros comandos es posible acelerar considerablemente la comunicación. La secuencia de inicialización requiere de suficiente tiempo (para una frecuencia de 8MHz es de aproximadamente de 2 segundos) y depende además, del tamaño de memoria y tipo de tarjeta.

Figura 35. Secuencia de inicialización completa



Fuente: Autor.

Envío de comandos en el modo SPI. En el modo de SPI, los comandos son enviados a la tarjeta de memoria SD como paquetes de seis bytes, y todas las respuestas de la tarjeta SD son proporcionadas en varios bloques de datos de longitud variable. La tarjeta identifica los comandos porque el primer byte de estos (byte 1) siempre comienza por 01, el resto de bits del primer byte contiene el número de comando codificado en binario natural, éste es seguido por una dirección de memoria de 32-bit, que corresponde a un entero largo sin signo cuyos bits más significativos deben ser enviados en primer lugar. La estructura del paquete de datos se muestra en la tabla 15.

Tabla 15. Formato de un paquete de datos

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
01	COMANDO	DIRECCIÓN			CRC

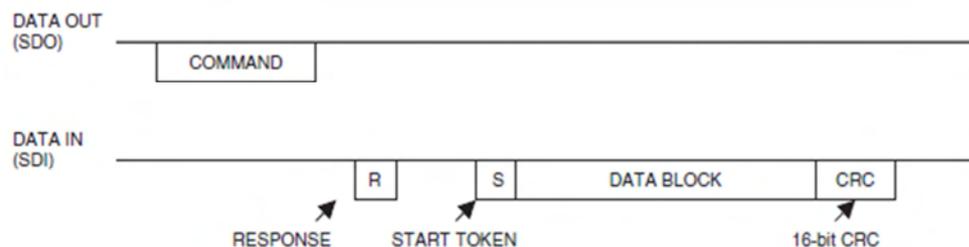
Fuente: Autor.

Finalmente, el comando se completa con un byte CRC, también llamado byte de comprobación de redundancia cíclica (CRC). Sin embargo, aunque ésta característica se utiliza siempre en el modo de bus SD, para asegurarse que todos los comandos y bloques de datos transmitidos estén libres de errores, en el modo SPI, la protección CRC se desactiva automáticamente. De esta forma, se puede simplificar considerablemente el código, en este caso, la comprobación se ignora.

Mediante la función *Comando_SD*, véase el Anexo B, se envía un comando y se espera la respuesta respectiva. Se comienza seleccionando la tarjeta (CS bajo) y paso seguido se envía el paquete de datos de 6 bytes a través del puerto SPI. En este caso, la respuesta es 0xFF (básicamente la línea SDI se mantiene alta) hasta que la tarjeta se encuentre lista para dar con un código de respuesta adecuado. Las especificaciones indican que son requeridos, hasta 74 pulsos de reloj antes de recibir una respuesta. En caso de que supere este límite, se asume un fallo en la tarjeta y se aborta la comunicación.

Lectura de datos. Una vez comprobado que el código de respuesta no contiene errores, se espera que la tarjeta envíe la señal de inicio, la cual identifica el principio del bloque de datos, en este momento se puede leer, en una secuencia rápida, los 512 bytes que componen el bloque de datos solicitado, el cual está seguido por un valor de 16-bit CRC que se debe leer pero que no tiene uso alguno en este caso. En este punto se desactiva la tarjeta de memoria y se da por terminado el proceso de lectura.

Figura 36. Transferencia de datos durante la lectura de bloques



Fuente: SD CARD ASSOCIATION. SD memory card specifications. Versión 1.01. San Ramon, CA.2001.

Como se mencionó anteriormente, todas las operaciones están definidas en bloques de tamaño fijo de 512 bytes. La función *leer_sd_sector*, véase el Anexo B, se utiliza para iniciar la lectura de un sector determinado de la memoria. Para leer un bloque de la tarjeta se envía el comando 17 con la dirección de inicio de lectura. La dirección puede tomar cualquier valor dentro del rango de direcciones válidas de la tarjeta de memoria, pero todo el bloque leído debe estar dentro de un mismo sector físico. Por lo tanto, se declara la variable *adress* como entero de 32 bits, la cual indica desde que dirección se va a dar lectura de los datos. Mediante la función *SD_insertada(void)* se cerciora que la tarjeta se encuentra insertada, en

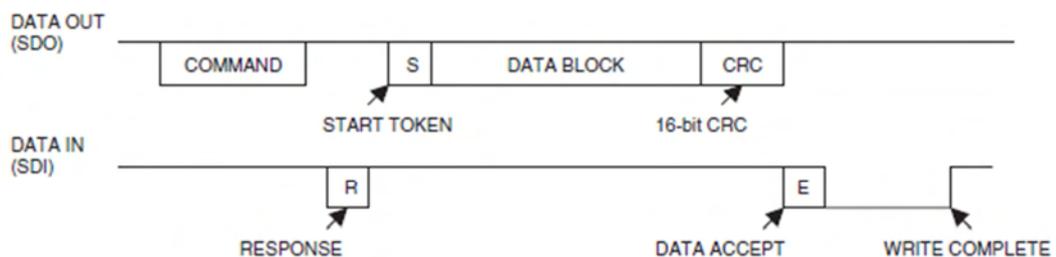
este caso, que el pin A1 es cero. Si es así, se guarda en la variable *adress* la dirección actual, la cual se obtiene multiplicando el valor LBA por 512, y ya que *dir_sector* es una variable de 32 bit se convierte 512 en un dato de 32 bit mediante $(int32)512$.

La lectura de una dirección dentro del sector, se realiza a través de la función *leer_sd_adress*. Si los pasos se concretan correctamente, entonces se ha terminado de leer el sector. Una vez más, como ocurre durante las fases de inicialización, es importante imponer un tiempo de espera. Puesto que sólo el llamado a la rutina *Leer_SPI()* se realiza repetidamente (envío / recepción de un solo byte a la vez) a la espera de la señal de datos, el proceso de lectura se lleva a cabo de forma lenta.

La función *leer_sd_adress* lee una dirección y retorna un 1 si se realiza la operación o un cero si no se cumple. Se declara una variable sin signo tipo long de 16 bits para realizar el incremento en la dirección que va a ser leída dentro del sector, una variable para el control del tiempo de espera del ciclo de lectura y una variable tipo entero para guardar la respuesta de la SD. Mientras el valor de la respuesta sea distinto de cero, se asigna a la respuesta el valor del comando en la dirección designada en *Comando_LEE*. Se establece un tiempo de espera de 100us para acceder al puerto SPI, para que en caso de producirse un tiempo mayor, abortar el proceso y retornar un *false*. Cuando el puerto SPI está listo para dar lectura, se incrementa la variable *i* y se guarda en el arreglo *sd_sector [i]* los datos del sector. Finalmente se da lectura de los 16 bits CRC.

Escritura de datos. Sobre la base de la consideración hecha para las funciones de lectura, se desarrolla la función de escritura, la cual también está limitada a operar en sectores, es decir, bloques de 512 bytes. Para realizar la escritura de un único bloque se envía el comando 24 indicando la dirección del bloque en el argumento de la función. La función *escribir_sd_sector*, véase el Anexo B, realiza la escritura de datos por bloque, en este caso, la transferencia de datos está en dirección opuesta a la lectura.

Figura 37. Transferencia de datos durante la escritura



Fuente: SD CARD ASSOCIATION. SD memory card specifications. Versión 1.01. San Ramon, CA.2001.

Cuando la tarjeta de memoria da respuesta al comando de escritura, se realiza el envío de los 512 bytes de datos, seguido de dos bytes CRC. En este punto se comprueba la respuesta dada por la tarjeta y se confirma que todo el bloque de datos ha sido recibido. Mientras la tarjeta está ocupada escribiendo, la línea SDO se mantiene baja.

Como sucede en el proceso de lectura, se constata que la tarjeta de memoria esté insertada y adicionalmente, se verifica que no tenga protección contra escritura. Paso seguido se envía el comando de escritura y la dirección de escritura; la dirección de escritura se obtiene de multiplicar el número del sector por 512.

En la función *escribir_sd_adress* se recibe la respuesta R1, la cual indica si se produjo algún error durante la recepción del comando de escritura. Si todo es correcto, la tarjeta envía tres respuestas R1 repetidas informando al microcontrolador que ya puede enviar el bloque de datos, que ha de tener una longitud de 512 bytes, ya que en la escritura solo se permite 512 bytes, omitiendo el byte de inicio del bloque de datos y los dos bytes de CRC.

Si todo es correcto el PIC envía un token (0xFE) y luego los 512 datos del bloque más 2 bytes de CRC; luego se mantiene a la espera de una respuesta de la memoria indicando si los datos han sido recibidos correctamente o ha ocurrido un error.

La finalización de la operación de escritura se lleva a cabo cuando los bits de estado de R1 son '010' lo que indica que los datos han sido aceptados. Una vez más, se impone un tiempo de espera para permitir que la tarjeta complete la operación. Puesto que todas las memorias SD/MMC se basan en tecnología Flash, se puede inferir que el tiempo normalmente requerido para una operación de escritura es considerablemente mayor al requerido para una operación de lectura. Para este caso, el límite de 0,3 s es más que suficiente para satisfacer incluso la tarjeta de memoria más lenta del mercado.

Por otro lado, para proporcionar una indicación visual de la actividad de la tarjeta de memoria, similar a la utilizada en las unidades USB Flash, se ha asignado un LED de la placa de control como indicador, con el fin de evitar que un usuario retire la tarjeta mientras está en uso. El LED se enciende tan pronto como la tarjeta se inicializa, intermite cuando se está leyendo o escribiendo y se apaga para indicar al usuario que la tarjeta se puede retirar.

4.1.2.2 Bloque funcional GPS. El GPS empleado corresponde a un receptor GPS de la empresa Tyco Electronics de la línea A1029, capaz de recibir señales de hasta 12 satélites que indican el posicionamiento global del dispositivo. El GPS recibe información precisa de posición, velocidad y tiempo (PVT) actualizando los datos anteriores con los que se recibe en tiempo real. La característica principal de

este módulo radica en que los datos recibidos desde el satélite son transmitidos vía serial a través del estándar RS232.

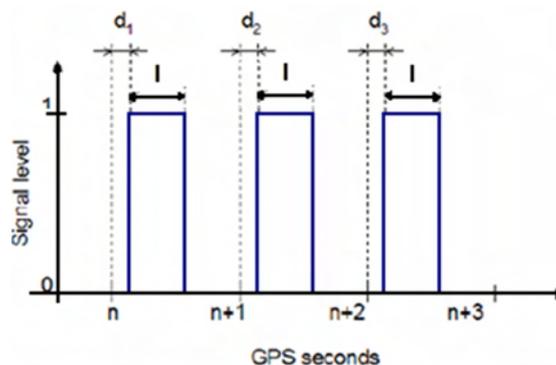
Figura 38. Módulo receptor GPS A1029-C de Tyco Electronics



Fuente: TYCO ELECTRONICS.

Además de un posicionamiento preciso, el receptor GPS permite una sincronización exacta debido a los relojes que son sincronizados por los satélites GPS; el módulo receptor A1029-C de la figura 38, adquirido para el desarrollo del proyecto, cuenta con un pulso por segundo (1PPS), sincronizado con la hora universal (UTC) y utiliza un oscilador de cristal compensado térmicamente (TCXO) lo que mejora su rendimiento alrededor de un 25% comparado con un cristal estándar. El pin 1PPS es un pin de salida y proporciona una señal de tiempo exacto y preciso. Se debe tener en cuenta que la señal 1PPS sólo es válida, siempre y cuando el módulo GPS proporcione una posición válida.

Figura 39. Forma de onda de la señal 1PPS



Fuente: TYCO ELECTRONICS. GPS Receivers A1029 user's manual. California, 2005.

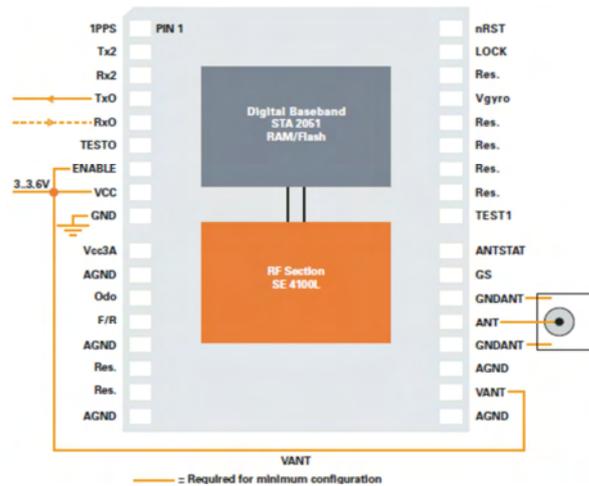
Tabla 16. Características generales del módulo GPS

Canales		12, seguimiento paralelo
Frecuencia de operación		L1 = 1575MHz
Sensibilidad	Seguimiento	-156 dBm
	Adquisición	-138 dBm
Interfaz de comunicación		SPI, I2C
Sentencias NMEA		GGA, GSA, GSV, VTG, RMC
Baudrate		4800 a 57600 baudios

Fuente: TYCO ELECTRONICS. GPS Receivers A1029 user's manual. California, 2005.

La configuración mínima recomendada, la cual se muestra en la figura 40, para salida NMEA a través de una interfaz RS232 del modulo receptor GPS A1029-C, se emplea los pines descritos en la tabla 17.

Figura 40. Descripción pines entrada-salida del módulo receptor GPS A1029-C



Fuente: TYCO ELECTRONICS. GPS Receivers A1029 user's manual. California, 2005.

Tabla 17. Descripción de pines utilizados en la configuración mínima del GPS A1029-C

Pin	Símbolo	Descripción
4	Tx0	Salida Serial, NMEA
5	Rx0	Entrada Serial, NMEA
7	Enable	+3.3V
8	Vcc	+3.3V
9	Gnd	Tierra general
21	Gndant	Tierra antena
22	Ant	Señal Antena 50ohm
23	Gndant	Tierra antena

El fabricante recomienda usar una antena GPS activa con tensión de alimentación de 3 a 5 Vdc y con un consumo de corriente de máximo 50 mA. La calidad de la antena GPS elegida es de vital importancia para la sensibilidad del sistema GPS. Una antena activa debe tener una ganancia mayor o igual a 20 dB y una susceptibilidad al ruido menor a 1.5dB. La antena activa puede estar montada directamente sobre el módulo GPS, o separada del circuito de control y encapsulada para protegerla del clima.

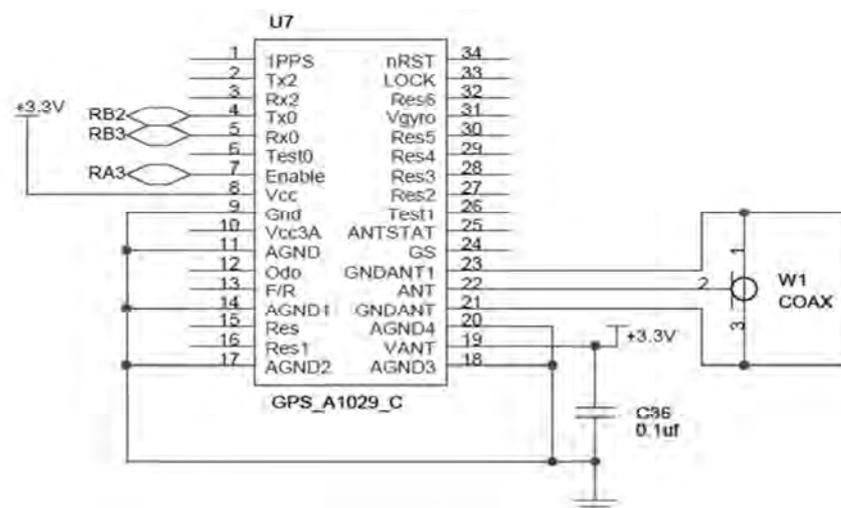
Figura 41. Antena activa del receptor GPS



Fuente: TYCO ELECTRONICS. GPS Receivers A1029 user's manual. California, 2005.

A pesar que el microcontrolador PIC18F2550 dispone de un módulo EUSART (versión mejorada del USART, en inglés Enhanced Universal Synchronous Asynchronous Receiver Transmitter), la comunicación con el GPS se realizó mediante emulación por software debido a que éste módulo se comparte con el SPI, el cual fue empleado para la adquisición y almacenamiento de datos de la tarjeta de memoria SD y para la conversión digital analógica. La figura 42 muestra la conexión del receptor GPS al microcontrolador.

Figura 42. Diagrama de pines y conexión del módulo GPS al PIC18F2550



Fuente: Autor.

El receptor GPS A1029 efectúa el envío de datos a través del protocolo NMEA a una velocidad de 4800 baudios. Esta información es enviada al microcontrolador el cual escoge la sentencia GPGGA, que es la que contiene los datos de relevancia para el programa. Los datos de la sentencia GPGGA corresponden a:

Tabla 18. Valores reales de una trama GPGGA

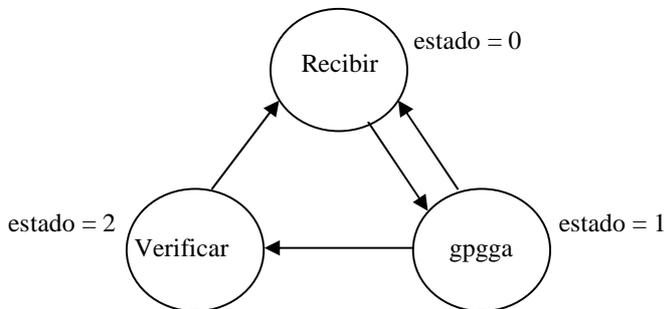
\$GPGGA,161038.000,0112.2241,N,07717.5960,W,1,06,2.5,2631,M,47.6,M,,*66	
Formato:	\$GPGGA,hhmmss.ss,lll.ll,a,mmmm.mm,b,x,yy,z.z,c.c,M,v.v,M,d.d, www*hh
161038	Datos tomados a las 16:10:38 UTC
0112.2241	Latitud: 01° 12' 22.41"
N	Hemisferio de latitud: Norte
07717.5960	Longitud: 077° 17' 59.60"
W	Hemisferio de longitud: Occidente
1	Calidad de la señal GPS válida
06	Número de satélites en uso
2.5	Dilución horizontal de precisión: 2.5
2631	Altitud sobre el nivel del mar: 2.631
M	Unidad de medida de la altitud: Metros (M)
47.6	Separación geoidal: 47.6
M	Unidad de medida de la separación: Metros (m)
	(Campo vacío) Tiempo desde la última actualización DGPS (segundos)
	(Campo vacío) Identificación de la estación DGPS
*66	Palabra de comprobación o checksum

Fuente: Autor.

Para extraer los datos específicos de la cadena de caracteres, enviada por el módulo GPS, a través del microcontrolador, se implementa una máquina de estados finitos, mediante la cual se almacena la sentencia GPGGA.

La recepción y discriminación de la trama GPGGA se realiza mediante la máquina de estados finitos de la figura 43. La función *Task_Rx_gps*, véase el Anexo B, utiliza la sentencia switch para estimar el valor del estado y se efectúa una tarea dependiendo de su valor, pudiendo crear una trama nueva, comprobar si la trama corresponde a la esperada y verificar si está completa. Esta función, tipo short, retorna un true cuando la trama recepcionada GPGGA está completa.

Figura 43. Máquina de estado para discriminación de la sentencia GPGGA



Fuente: Autor.

En el estado cero o case “recibir”, los datos del buffer del puerto RS232 se sacan y almacenan de forma dinámica, en el arreglo *Trama_IN*, el cual contiene las tramas recepcionadas. Si existen datos, el estado toma el valor de uno.

En el estado uno o case “es_gpgga”, mediante la rutina *trama_gpgga*, se verifica que la trama inicie con \$GPGGA. Para lograrlo, se define el arreglo de caracteres *trama_comp[]* y se le asigna la constante {"\$GPGGA"}. La verificación se realiza comparando los seis primeros caracteres de *Trama_in* con la cadena *trama_comp*. Si la trama recibida no corresponde a una sentencia GPGGA, se retorna al estado cero para crear una nueva trama. Si el resultado es verdadero, continúa al siguiente estado; de esta manera, las demás tramas entregadas por el receptor GPS no son tenidas en cuenta.

En el estado dos o case “verificar”, la rutina *Trama_Completa* verifica que la trama GPGGA entregada por el GPS está completa, es decir, que inicia con el caracter \$ y termina con el caracter *. Si está completa, se trunca hasta los datos necesarios para el programa, esto es hasta el elemento 47, que corresponde al número de satélites en uso y se salva el contenido de la trama de entrada, en el arreglo *Trama_final*.

Tabla 19. Valores ASCII extraídos de una trama GPGGA

\$GPGGA,161038.000,0112.2241,N,07717.5960,W,1,06			
Latitud		Longitud	
Trama final [18]=0	Trama final [23]=2	Trama final [30]=0	Trama final [36]=5
Trama final [19]=1	Trama final [24]=2	Trama final [31]=7	Trama final [37]=9
Trama final [20]=1	Trama final [25]=4	Trama final [32]=7	Trama final [38]=6
Trama final [21]=2	Trama final [26]=1	Trama final [33]=1	Trama final [39]=0
Trama final [22]=.	Trama final [28]=N	Trama final [34]=7	Trama final [41]=W

Fuente: Autor.

De la trama GPGBA almacenada, se realiza la selección de los datos de latitud y longitud, los cuales se expresan bajo el sistema de coordenadas geográficas y son caracteres ASCII. Por este motivo, la función *extraer_coordenadas* los convierte a datos tipo float para poder operar con ellos. Los datos necesarios para la comparación de la distancia se guardan en el arreglo *Trama_final* cuyas posiciones se indican en la tabla 19.

Con esto sólo resta detallar cómo se obtiene la base de comparación con el GPS, es decir, el proceso que se lleva a cabo para comparar las coordenadas actuales con las asociadas al archivo de voz almacenado en la tarjeta de memoria SD.

Cálculo de la distancia. Las coordenadas tomadas con el receptor GPS, que corresponden a coordenadas geográficas, están referidas al elipsoide nativo del sistema, el WGS-84 y se expresan en grados, minutos y segundos sexagesimales. Puesto que se pretende calcular la distancia entre dos puntos, es necesario cambiar la representación dada por el modelo geométrico (elipsoide) a una representación plana; esto se logra mediante la conversión de coordenadas geográficas a un sistema de coordenadas UTM.

Lo primero que se hace es expresar los grados sexagesimales en notación decimal para poder realizar operaciones; esto se realiza mediante la función *conv_ascii_coordenadas*, la cual convierte los caracteres ASCII guardados en el arreglo *Trama_final* a grados sexagesimales expresados en notación decimal, mediante la expresión:

$$\text{grados decimales} = \text{grados} + \text{minutos}/60 + \text{segundos}/3600 \quad (4)$$

Las coordenadas se guardan en las variables de tipo float *latitud_dec* y *longitud_dec* mediante la función *extraer_coordenadas*. Con las coordenadas expresadas en grados decimales, se procede convertirlas a metros. Esto se logra aplicando las ecuaciones de Cotichia-Surace, cuyas expresiones se detallan en el Anexo C.

Una vez se tiene las coordenadas en metros, se realiza el cálculo de la distancia que existe entre las coordenadas en cuestión; éste se efectúa mediante la función *coordenada_cerca*. En esta función se vuelca el valor de las coordenadas guardadas en la tarjeta SD, en dos variables, para compararlas con las coordenadas actuales leídas por el receptor GPS.

En ese orden, primero se accede al sector cero de la memoria en el cual se encuentra almacenado el número de paradas, esto se hace con el fin de recorrer todas las posiciones de memoria que contienen los datos de las coordenadas obtenidas mediante el software GPS. Se utiliza el contador *i*, para acceder a los 10 sectores donde se encuentra almacenada la tabla de índices y, el contador *j* para

acceder a los primeros 32 bytes de información de cada sector de la tarjeta de memoria SD. Por ejemplo, cuando i es cero, el microcontrolador lee el índice del primer sector iniciando desde el byte cero. Los bytes 0 a 3 se convierten en una dirección de 32 bits que se guarda en la variable puntero *index*, igualmente sucede con los bytes 4 a 7 de la variable puntero **n_sector*. La asignación de variables del índice de los archivos de voz almacenados en la tarjeta de memoria SD se muestra en la tabla 20.

Tabla 20. Variables del índice de un archivo de audio

Byte	Variable	Descripción
0-3	*index	Dirección de inicio del archivo de voz
4-7	*n_sector	Tamaño del archivo wav
8-15	lat_comp	Latitud
16	lat_c_comp	Norte o Sur
17-24	lon_comp	Longitud
25	lon_c_comp	Oeste o Este
26	*freq_muestreo	Frecuencia de muestreo
27-32	---	Bytes reservados

Fuente: Autor.

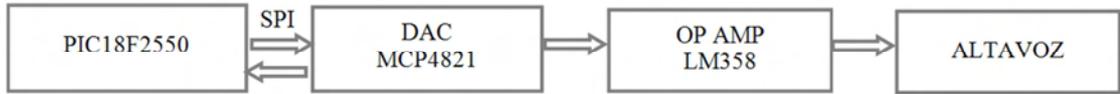
Una vez se tiene direccionado el primer archivo de voz, se procede al cálculo de la distancia entre las coordenadas; puesto que se cuenta con 4 coordenadas planas, la distancia queda determinada por la expresión:

$$distancia = \sqrt{(lat_{comp} - lat)^2 + (lon_{comp} - lon)^2} \quad (5)$$

En esta función se determina el margen de distancia para la reproducción del archivo de voz que anuncia la próxima parada, el cual corresponde a 200 metros. El contador j permite acceder a cada índice mediante un incremento de 32 bytes, que es el tamaño de un índice; el proceso de comparación continúa validando los demás índices hasta determinar un resultado positivo en la comparación o hasta completar los 10 sectores que ocupa la tabla de índices.

4.1.2.3 Bloque funcional conversor D/A. Cuando el resultado de la comparación de coordenadas es verdadero, se ordena la reproducción del archivo de voz. Para esto es necesario realizar una conversión digital-analógica, la cual se implementa con el conversor D/A MCP4821 de Microchip. Se trata de un conversor digital-analógico de baja potencia de 12-bit de resolución, que cuenta con voltaje de referencia interno. Este dispositivo utiliza una interfaz de comunicación serie SPI y opera en un rango tensión de 2.7V a 5.5V.

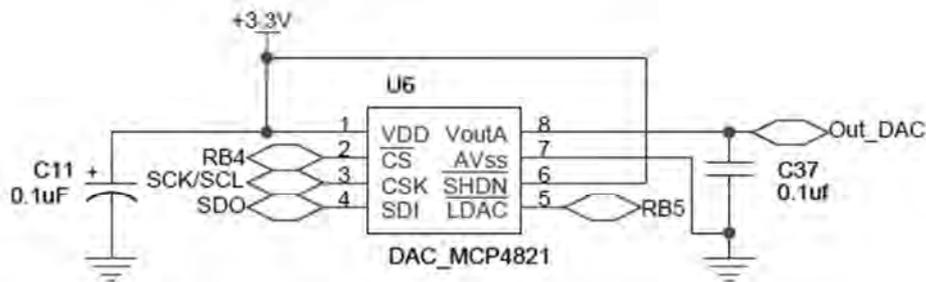
Figura 44. Bloque funcional etapa de conversión y amplificación



Fuente: Autor.

Además de las tres conexiones CS, SCK y SDI, el dispositivo MCP4821 cuenta con la señal LDAC la cual sincroniza la salida del DAC con el estado disponible en el pin LDAC. Cuando el estado del pin LDAC es bajo, los datos de entrada y de configuración del registro de entrada son transferidos al registro de salida, y la salida *Out_DAC* se actualiza con el valor analógico de voltaje V_{OUT} . Cuando el pin LDAC esta en alto, la salida se actualiza inmediatamente una vez se completa una transmisión válida. La figura 45 muestra las conexiones del DAC con el microcontrolador, en ésta se observa que el pin LDAC se conecta con el PIC mediante el pin RB5.

Figura 45. Diagrama de pines y conexiones del DAC MCP4821



Fuente: Autor.

La expresión 4 muestra el cálculo de la tensión de salida analógica. La codificación del valor digital de entrada se encuentra en binario, mientras que la salida analógica está expresada en voltios.

(6)

Dónde:

- Voltaje interno de referencia
= 2.048V
- Valor digital de entrada
= (0 - 4095)
- Selección de ganancia
= 2 para GA bit = 0

$n = 1$ para GA bit = 1bit
 bits de resolución
 $= 12$ bit

Reemplazando en 6:

$$V_{OUT} = \left(\frac{2.048V \times DN}{4096} \right) \times (1 \text{ o } 2) \quad (7)$$

En este caso, la salida del amplificador tiene dos opciones de ganancia configurables: un rango de salida ideal de 0.0V a $4095/4096 * 2.048V$ cuando la selección de ganancia es 1 y un rango de 0.0V a $4095/4096 * 4.096V$ cuando la selección de ganancia es 2.

Interfaz SPI. El DAC MCP4821 soporta comunicación SPI en el modo 0.0 y modo 1.1. Los comandos y datos son enviados al dispositivo a través del pin SDI en el flanco ascendente de la señal SCK. La comunicación es unidireccional y, por tanto, los datos no pueden ser leídos una vez son transmitidos al dispositivo. El pin CS debe mantenerse bajo durante la duración de un comando de escritura; el comando de escritura se compone de 16 bits y permite la configuración del DAC. La Tabla 21 detalla la estructura del registro de entrada usado para configurar y cargar el DAC.

Tabla 21. Estructura registro de entrada del MCP4821

W-x	W-x	W-x	W-0	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
0	--	GA	SHDN	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit15								Bit0							

Fuente: MICROCHIP TECHNOLOGY.

Dónde:

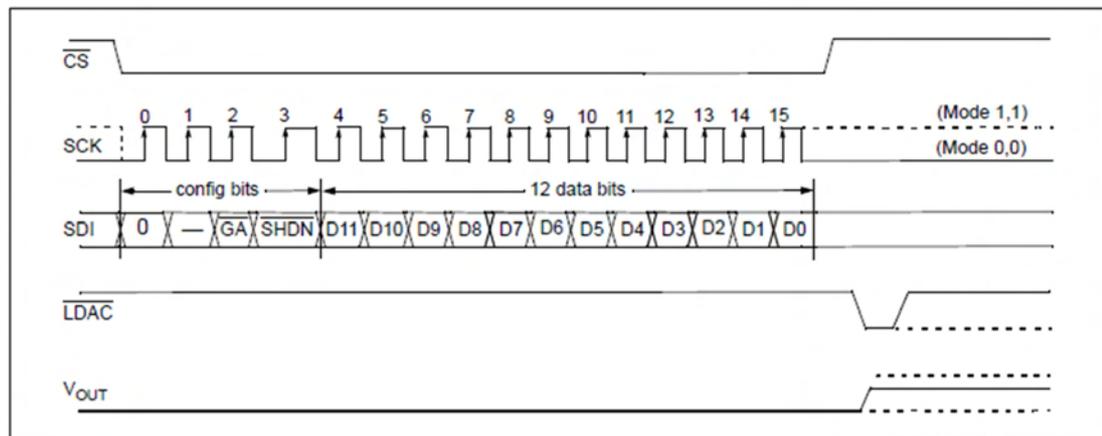
- bit 15 0 = Escribe en el registro del DAC
 1 = Ignora el comando
- bit 14 No importa
- bit 13 GA: bit de selección de la ganancia de salida
 1 = $1x(V_{OUT} = 1 * V_{REF} * D/4096)$
 0 = $2x(V_{OUT} = 2 * V_{REF} * D/4096); V_{REF} = 2.048V$
- bit 12 SHDN: bit de control de desconexión de la salida
 1 = Active el modo de operación shut down. V_{OUT} disponible
 0 = La salida analógica no está disponible
- bit 11-0 D11:D0: Datos de entrada (bits)

Proceso de escritura. Se inicia poniendo el pin CS en bajo, seguido por la

transferencia de cuatro bits de configuración y 12 bits de datos en el pin SDI mediante el flanco ascendente de la señal de reloj. Terminada la transferencia se pone el pin CS en alto, lo que bloquea el ingreso de datos en el registro de entrada del DAC.

El MCP4821 utiliza una estructura de doble buffer lo que permite sincronizar la salida analógica con el pin LDAC, de esta forma, al colocar el pin LDAC en estado bajo, el contenido almacenado en el registro de entrada se transfiere al registro de salida, actualizando V_{OUT} . La figura 46 muestra el diagrama de tiempos de las señales en el proceso de escritura.

Figura 46. Comandos de escritura del convertor D/A



Fuente: MICROCHIP TECHNOLOGY.

Todas las operaciones de escritura del DAC se realizan con palabras de 16 bits. Cualquier dato después de la 16^a señal de reloj se ignora; los bits más significativos corresponden a 4 bits de configuración, los 12 bits restantes corresponden a bits de datos. De acuerdo a la figura 46, la transferencia de datos sólo se produce si han sido transferidos 16 pulsos de reloj; si el flanco de subida de CS se produce antes, el cambio de datos en el registro de entrada se aborta.

La reproducción de los mensajes de voz se ordena mediante la función *activar_voz*. Esta función toma como argumento las variables puntero *índice*, *n_sector* y *freq_muestreo*. En *índice* se carga la dirección del archivo a reproducir; la variable *n_sector* contiene la cantidad de sectores que ocupa el archivo, esto para que evitar la reproducción de partes de otro mensaje y, *freq_muestreo* que indica la velocidad de muestreo.

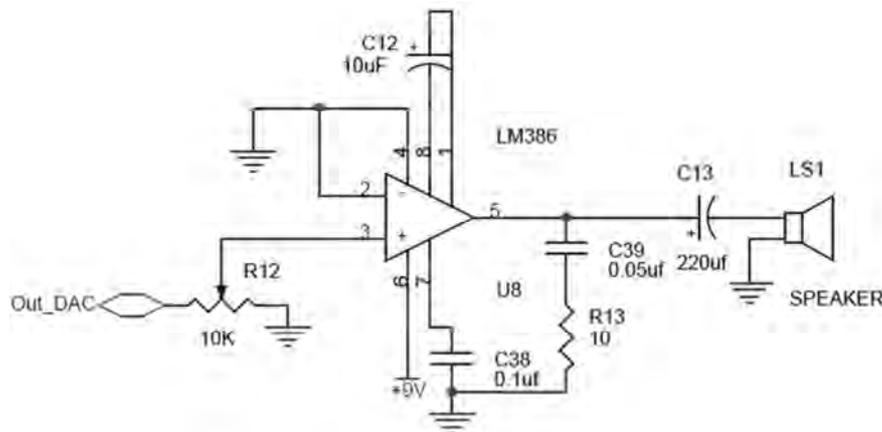
Todas las interrupciones son desactivadas para evitar que se produzca efecto de 'voz cortada' en la reproducción. Se verifica que la función *transferir_SD_RAM* se ha llevado a cabo de manera exitosa; en esta función se habilita y lee la tarjeta de

memoria SD y se transfieren los datos a la RAM auxiliar. Si la función devuelve un 1, se procede a transferir los datos binarios al convertidor D/A para su conversión al mundo analógico. El proceso de reproducción del archivo de voz se completa con la función *transferir_RAM_DAC* en la cual se escribe en el DAC.

La salida proporcionada por el DAC, es amplificada mediante el circuito desarrollado en torno al amplificador operacional LM386 constituido por un circuito de entrada diferencial, que puede ser alimentado con un voltaje mínimo de 5V.

Los pines 1 y 8 del amplificador operacional LM386 proporcionan un control de ganancia ajustable en el rango de 26db a 46db. En el circuito de la figura 47, el condensador de 10uf proporciona una ganancia fija de 200 (26db). El control de ganancia también se puede hacer por acoplamiento capacitivo de una resistencia (o FET) del pin 1 a tierra.

Figura 47. Circuito de amplificación con ganancia de 46db



Fuente: Autor.

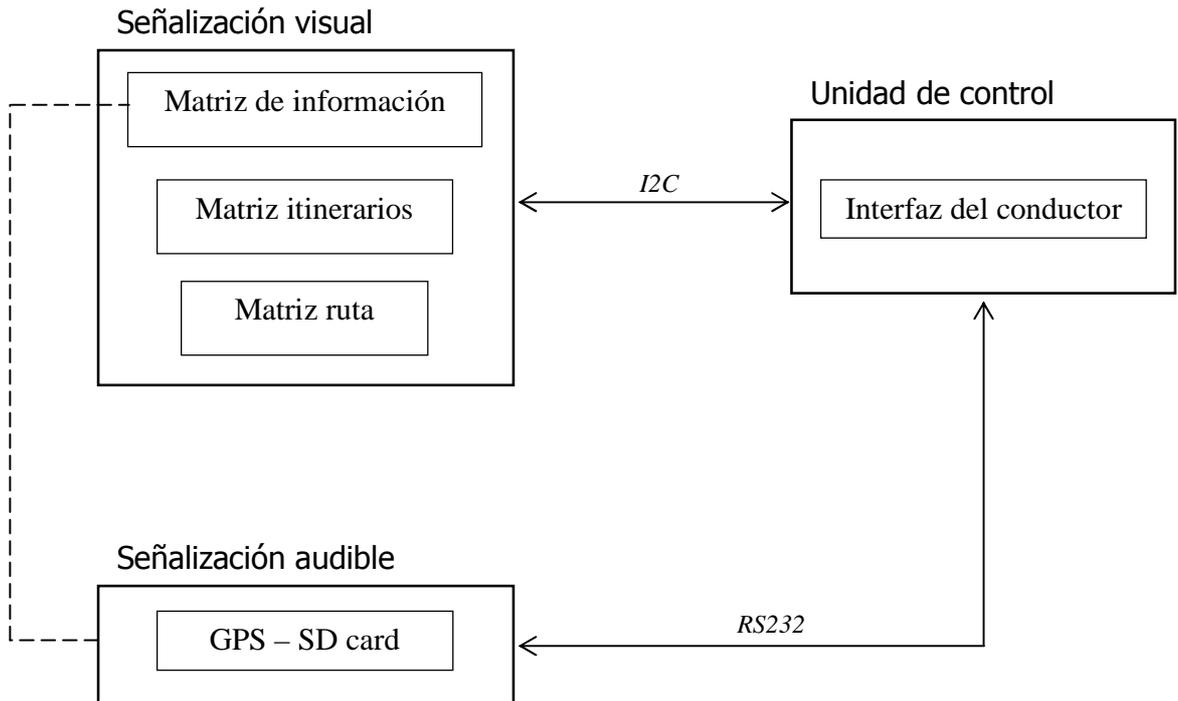
4.1.3 Interfaz del conductor. Además de los archivos de voz y la información de coordenadas, en la tarjeta de memoria SD se almacena las rutas e itinerarios de ruta, obtenidos mediante el programa de edición, que van a ser visualizados en los paneles electrónicos de matriz de leds.

El acceso a la memoria SD se realiza a través del módulo de señalización audible y es éste el que transfieren los datos al módulo de control del sistema, a través del puerto RS232 a una tasa de 9600 baudios. Al igual que con la comunicación con el GPS, la transferencia de datos se realiza por software utilizando la librería use RS232 que dispone CCS.

Por otra parte, mediante el mando del conductor, se realiza la selección de la ruta que va a ser operada por el conductor mediante teclado y pantalla LCD.

El envío de datos a los paneles electrónicos de matriz de leds se realiza desde la unidad de control del sistema a través de un bus I2C. En este caso, se graba la información concerniente a ruta, itinerarios y próxima parada, en la memoria eeprom respectiva de cada matriz. La figura 48 muestra los protocolos utilizados y el sentido de comunicación entre los módulos.

Figura 48. Comunicación entre los módulos del sistema

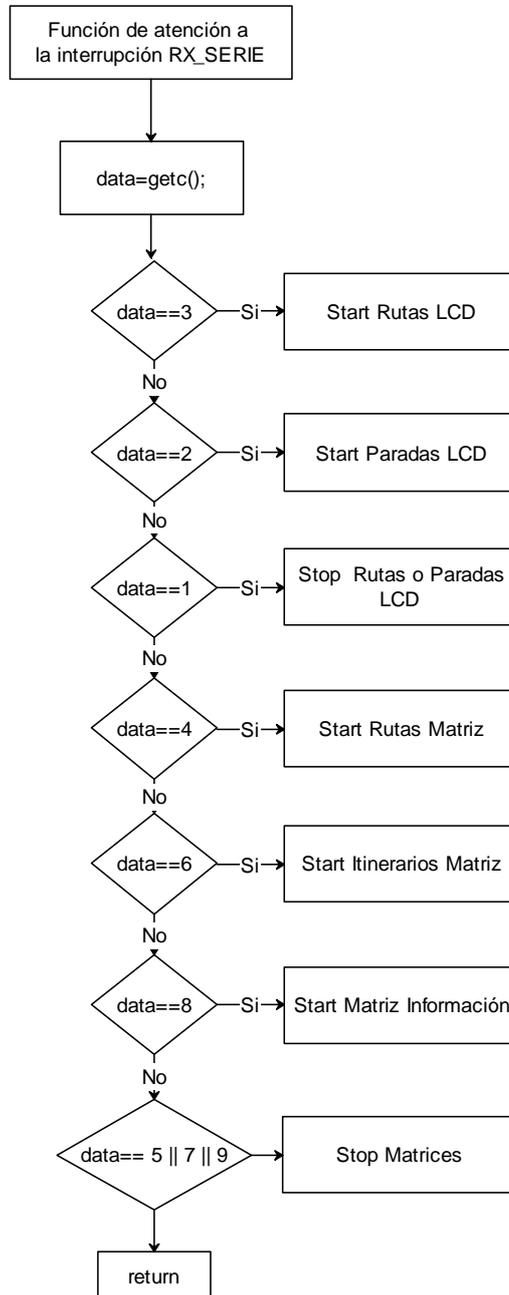


Fuente: Autor.

Para inicializar el sistema, el conductor introduce la contraseña asignada y paso seguido elige la ruta de operación. En ese momento, la unidad de control inicia la comunicación con el módulo de señalización audible mediante el envío de un byte de inicio, en solicitud a los datos pertenecientes a la ruta seleccionada.

El módulo de señalización audible responde a la petición con un byte de confirmación seguido de la trama de datos solicitada y finaliza con un byte de parada. Este procedimiento se lleva a cabo a través de la función, *RX_SERIE*, de atención a la interrupción por puerto serie *#INT_RDA*. La figura 49 muestra el diagrama de flujo implementado para la atención del puerto serie en la comunicación con el módulo de control del sistema.

Figura 49. Función de atención del puerto serie en la comunicación entre módulos



Fuente: Autor.

La tabla 22 muestra los bytes empleados en el proceso de solicitud de datos desde la unidad de control hacia el módulo de señalización audible, el cual controla la unidad de almacenamiento SD.

Tabla 22.Bytes utilizados en la comunicación entre la unidad de control y el módulo audible

Unidad de control	Módulo de señalización audible			
<i>Byte petición</i>	<i>Byte start</i>	<i>Trama</i>	<i># bytes</i>	<i>Byte stop</i>
10	3	Rutas LCD	[23]	1
9	2	Paradas LCD	[10]	1
8	4	Rutas Matriz	[30]	5
7	6	Itinerarios Matriz	[70]	7
6	8	Información Matriz	[80]	9

Fuente: Autor.

Puesto que los bytes de control son datos enteros que pueden estar incluidos en la trama de datos, se hace necesario diferenciar las palabras de control con ésta; los bytes de control son enviados a través del puerto serie mediante la función *putc*, mientras que los bytes de datos son convertidos a caracteres ASCII antes de ser enviados, a través del puerto serie RS232, al módulo de control del sistema.

La conversión se realiza mediante la función *Conv_INT_ASC*; en ésta cada byte de datos, que corresponde a un entero, se convierte en ASCII y se almacena en una cadena tipo string. La función *sprintf* envía el *valor* de la conversión alojado en la cadena *ip_ga_ma[]*. En este punto cada byte se compone de 3 caracteres. Para la conversión se tiene en cuenta tres casos posibles:

- Si el dato es menor que 9, las unidades son almacenadas en la posición LSB de la cadena y se completa con dos caracteres nulos "00". Ejemplo: dato = 2; *ip_ga_ma[0]* = 50, *ip_ga_ma[1]* = 48, *ip_ga_ma[2]* = 48. (valor ASCII de 0 es 48)
- Si el dato está entre 10 y 99, se almacena las unidades en la posición LSB y las decenas y se completa con un carácter nulo "0" en la posición MSB. Ejemplo: dato = 24; *ip_ga_ma[0]* = 52, *ip_ga_ma[1]* = 50, *ip_ga_ma[2]* = 48.
- Si el dato es mayor que 99 y menor que 255, se almacena las unidades, las decenas y las centenas que corresponden al valor ASCII del dato. Ejemplo: valor = 200; *ip_ga_ma[0]* = 48, *ip_ga_ma[1]* = 48, *ip_ga_ma[2]* = 50.

De esta forma, para enviar por ejemplo la trama de rutas, después de haber recibido el byte de petición de envío (8), se inicia la transmisión con el byte de start mediante *putc* (3), y paso seguido se procede a la llamada a la función que convierte los datos enteros, de la trama de datos, en caracteres ASCII. Este proceso se realiza hasta convertir los 30 bytes correspondientes a la matriz ruta y finaliza con el byte de stop mediante *putc* (1).

Entre tanto, el módulo de control al recibir el byte de start, envía a la función *Conv_ASCII_INT* los caracteres recepcionados, allí son convertidos en datos enteros y después almacenados en el arreglo *data_RUTA[i]*; posteriormente son enviados a la matriz correspondiente.

El envío de los datos a las matrices, para ser escritos en la memoria eeprom, se realiza a través del bus I2C. En el caso de la matriz ruta, la escritura de la eeprom se hace mediante la función *MATRIZ_RUTA*.

La relación de rutas y paraderos autorizados hasta la fecha por la unidad administrativa Avante, se encuentra detallado en el ANEXO D.

5. RESULTADOS

5.1 HARDWARE.

El sistema de información al usuario, figura 50, está diseñado con el fin de proporcionar la señalización visual y acústica requerida por un usuario de transporte público, antes y después del ingreso al autobús, permitiendo dar lectura del número de ruta y destino, en cualquier hora del día o de la noche durante la operación del automotor. Para poder cumplir con este requerimiento, el sistema se compone a nivel hardware, de tres módulos a saber, el módulo de señalización visual, el módulo de señalización acústica y la interfaz del conductor.

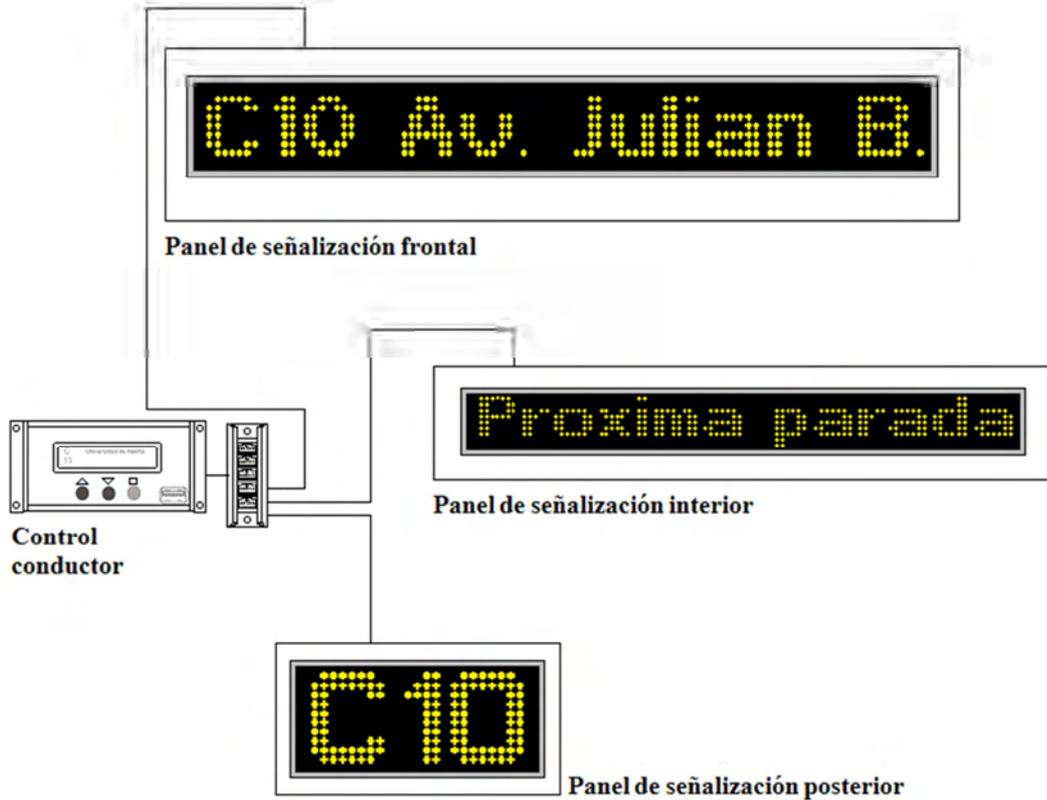
Figura 50. Sistema de Información al usuario



Fuente: Autor.

5.1.1 Módulo de señalización visual. En la figura 51 se muestra, como referencia, el diagrama esquemático del módulo de señalización visual del sistema.

Figura 51. Diagrama esquemático del módulo de señalización visual

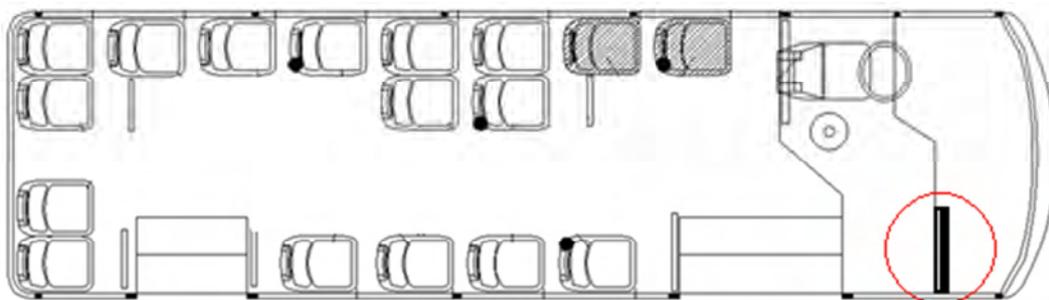


Fuente: Autor.

La señalización visual se realiza mediante tres paneles electrónicos unilínea de matriz de leds, construidos con leds de alta luminosidad. Se ubican al interior del autobús y cuentan con las siguientes características (para ver las especificaciones técnicas refiérase al Anexo E):

- **Panel frontal:** Suministra al usuario, antes de su ingreso al autobús, la información necesaria que le permite identificar ruta, origen y destino. Se ubica en el frente del autobús, en la parte superior del parabrisas, como se indica en la figura 52.

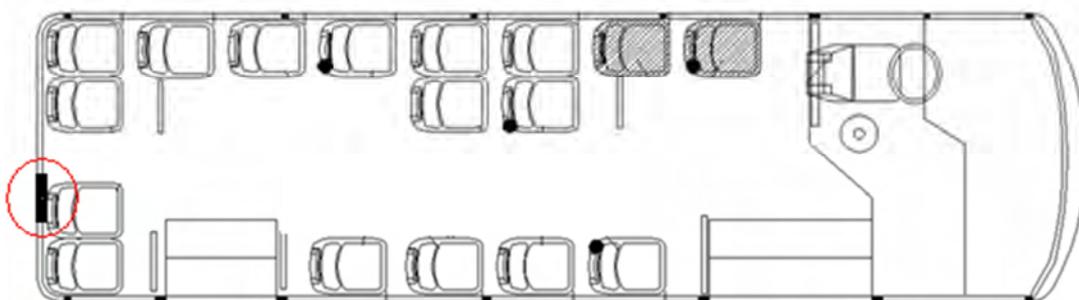
Figura 52. Instalación panel frontal



Fuente: ALCALDÍA DE PASTO. Decreto 735 de 2009. Pasto, 2009.

- Panel posterior: Permite la visualización del número de ruta, se ubica en la parte superior del vidrio panorámico trasero.

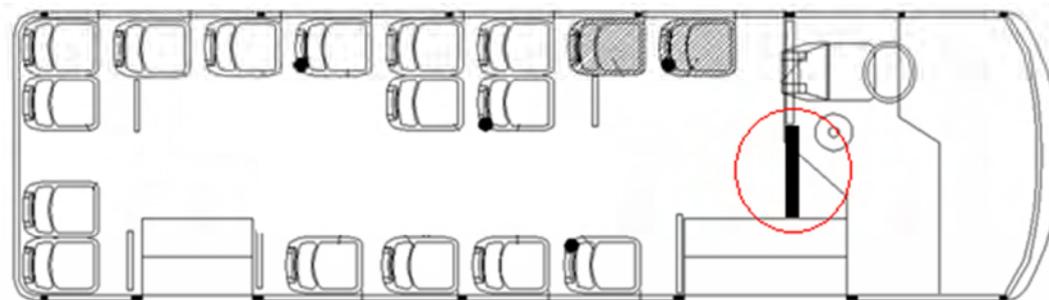
Figura 52a. Instalación panel posterior



Fuente: ALCALDÍA DE PASTO. Decreto 735 de 2009. Pasto, 2009.

- Panel interior: Proporciona a los usuarios dentro del bus, la información audible de próxima parada así como también mensajes informativos, se ubica en la parte superior transversal a la puerta de acceso.

Figura 52b. Instalación panel interior

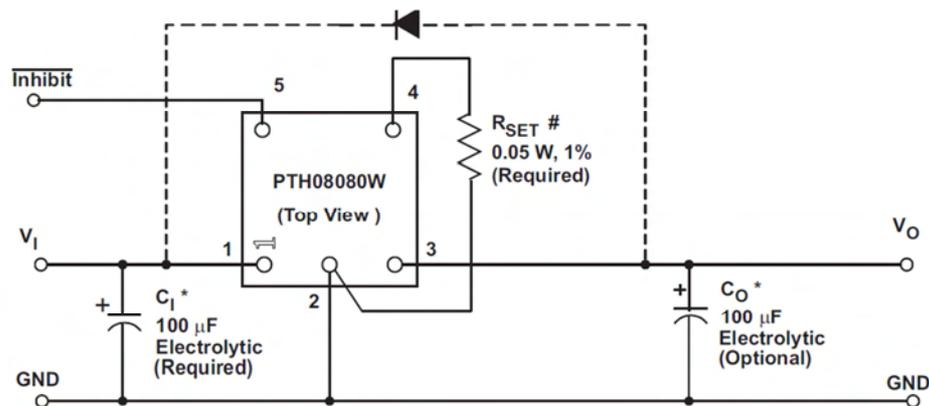


Fuente: ALCALDÍA DE PASTO. Decreto 735 de 2009. Pasto, 2009.

Los paneles de información al usuario disponen de gestión de datos y comunicaciones con el objeto de reprogramar la asignación de códigos-textos en la memoria interna del sistema. Los textos que se muestran en los paneles electrónicos son creados mediante el programa de edición, compatible con Windows, cuya descripción se encuentra más adelante. De esta manera, se dispone de una potente herramienta software, así como también de una interfaz entre la unidad de control y el puerto USB del PC donde se realiza las tareas de edición de texto y transferencia de los mensajes a visualizarse en los paneles electrónicos.

Se emplea un regulador tipo *switching* de alta eficiencia que adapta la tensión de la batería del automotor (12 Volts) a la tensión necesaria para la alimentación de los circuitos y provee una corriente de salida de 2.25A. Este tipo de regulador disminuye notablemente la disipación térmica en comparación con los reguladores lineales, eliminando la necesidad de disipadores, lo que se traduce en ahorro en el consumo. La fuente junto con la electrónica de control es desmontable para fácil mantenimiento y actualización.

Figura 53. Circuito conmutado empleado en la alimentación de los circuitos del sistema

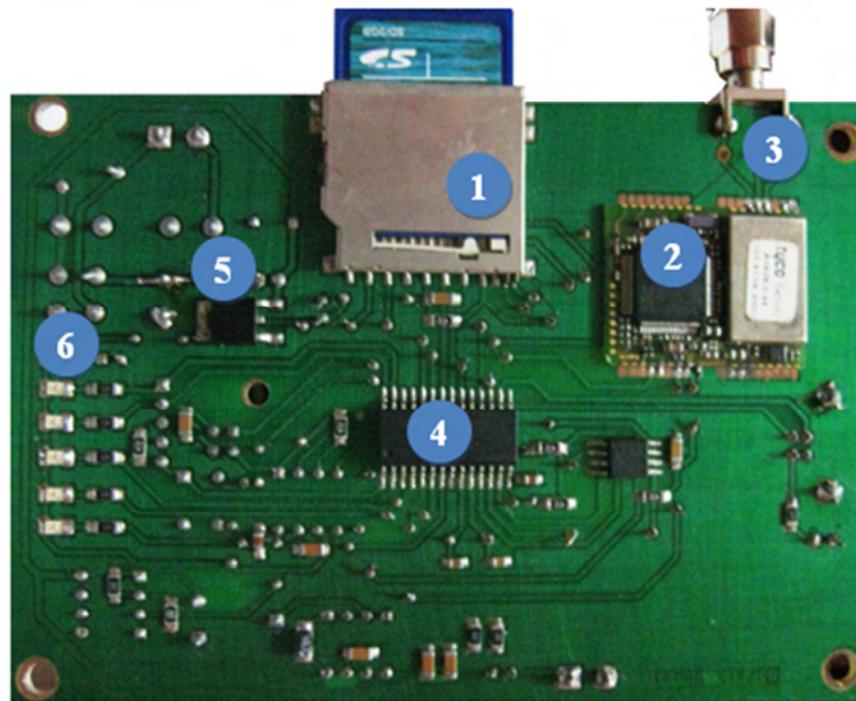


Fuente: TEXAS INSTRUMENTS. Adjustable switching regulator. 2009.

5.1.2 Módulo de señalización acústica. A nivel hardware, el módulo de señalización acústica se basa en un circuito de reducidas dimensiones y bajo costo diseñado para su programación directa a través de un conector ICSP (8), lo que permite realizar cambios y/o actualizar el programa principal de la unidad de control. Los dos modos de operación, programación y operación independiente, pueden ser seleccionados mediante el pulsador (9). Las figuras 54 y 55 muestran fotografías reales de la tarjeta electrónica que indican las secciones más relevantes del circuito.

1. Slot para inserción de la tarjeta de memoria SD.
2. Receptor GPS A1029-C. Proporciona una salida serie compatible con el protocolo NMEA.
3. Conector de la antena activa del receptor GPS. Proporciona la alimentación de la antena activa de 3.3V y una corriente de 50 mA.
4. Microcontrolador PIC18F2550 en empaquetado SMD. Este microcontrolador se encuentra conectado a un cristal de cuarzo de 20MHz y los pines para descarga de firmware están conectados al conector ICSP para la programación in-circuit.
5. Regulador LM117. Este regulador entrega 3.3V, soporta como máximo 15V de entrada y proporciona una corriente máxima de 800mA.
6. Leds de monitoreo.

Figura 54. Secciones de la tarjeta electrónica del módulo de señalización acústica P.1



Fuente: Autor.

7. Regulación. La placa cuenta con regulación de tensión y protección contra sobre corriente incorporados, para esto se dispone del regulador LM117 que reduce el voltaje 3.3V para la alimentación de la tarjeta SD y de un fusible de 500mA.
8. Conector ICSP de 5 vías. Header de conexión de los pines de programación del microcontrolador para la descarga de las actualizaciones del firmware. Los pines están distribuidos como se muestra en el esquema:

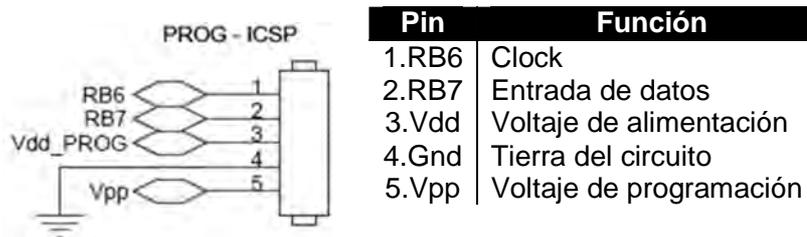
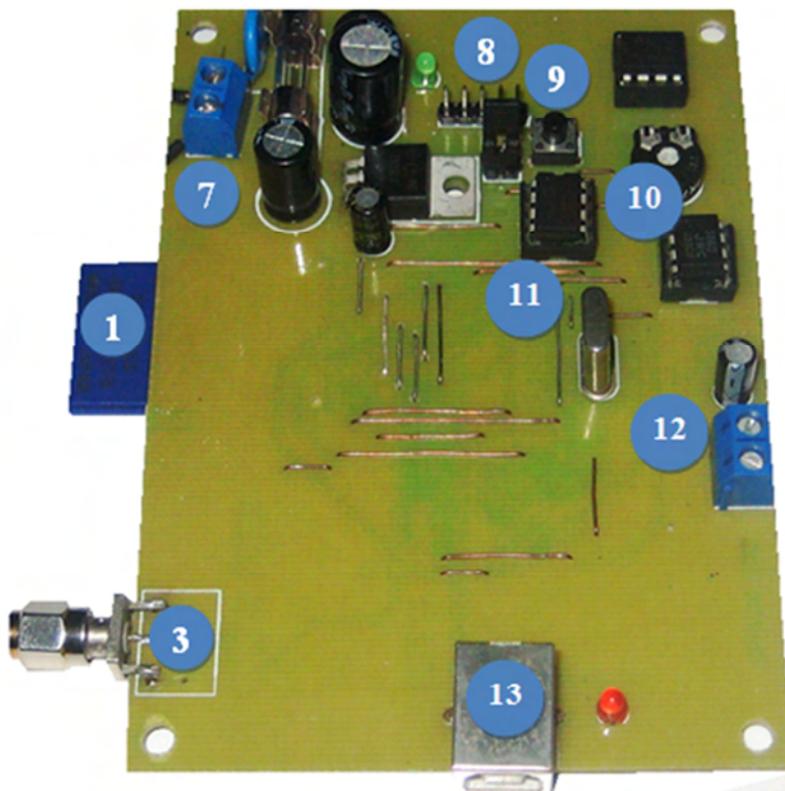


Figura 55. Secciones de la tarjeta electrónica del módulo de señalización acústica P.2



Fuente: Autor.

9. Pulsador modos de operación. La figura 56 muestra el diagrama de flujo implementado para seleccionar uno de los modos de operación del sistema: modo de operación independiente o modo programación.

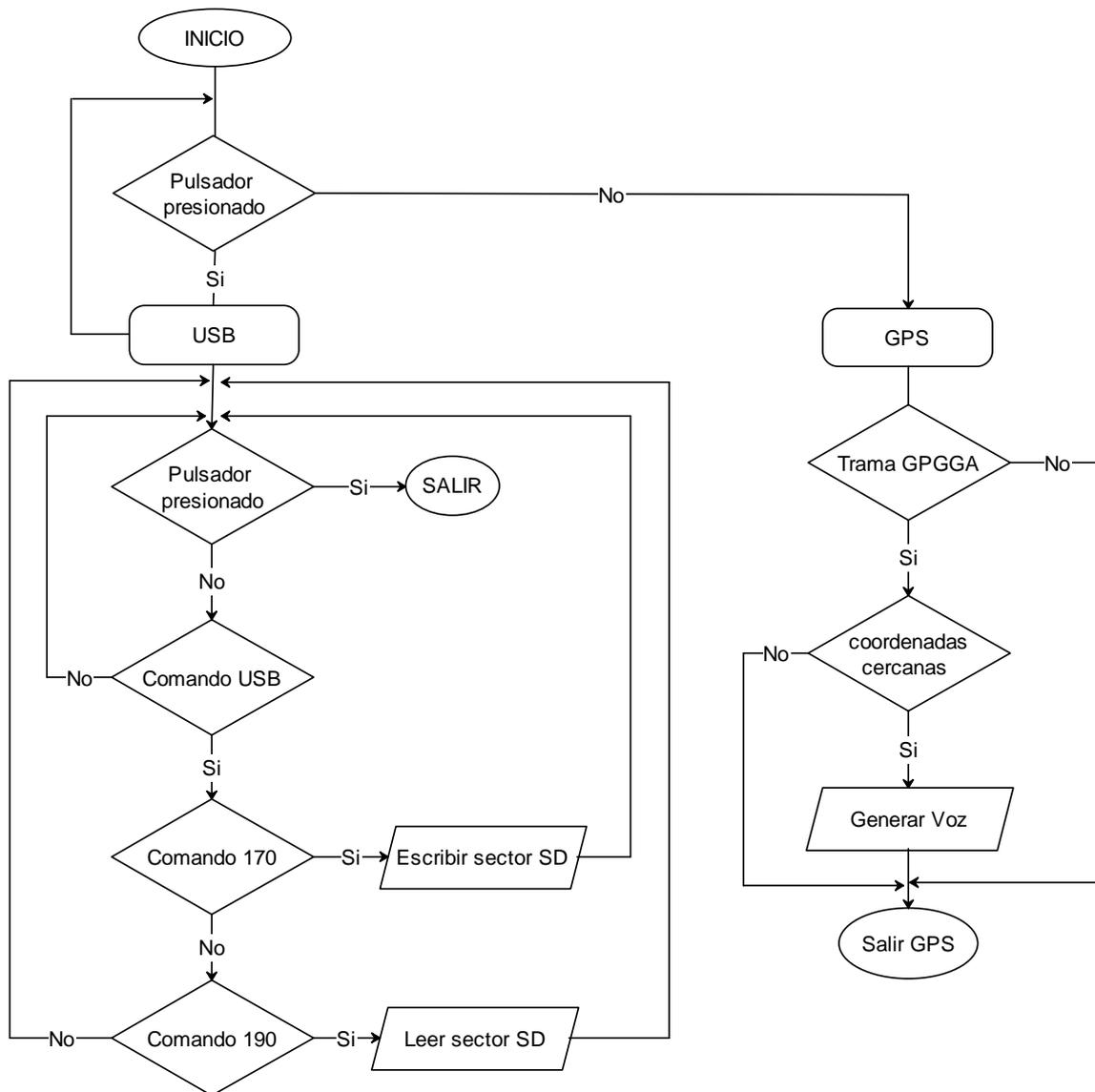
10. Circuito amplificador. Desarrollado entorno al amplificador LM386 proporciona una ganancia de 46dB.

11. Conversor D/A. Se trata del circuito integrado MCP4821 de 12 bits de resolución.

12. Salida amplificador.

13. Conector USB tipo B. Permite la programación del sistema, mediante el envío vía USB, de los datos obtenidos a la unidad de control. Tiene un LED indicador de conexión al puerto USB del computador.

Figura 56. Modos de operación del sistema



Fuente: Autor.

5.1.3 Interfaz del conductor. Se desarrolla entorno a una pantalla LCD y teclado alfanumérico, los cuales permiten al conductor del autobús identificarse como personal autorizado para operar el vehículo. La clave de ingreso se solicita para poner en funcionamiento el sistema, y se ingresa mediante el teclado. Una vez el sistema se inicializa, el conductor realiza la elección de la ruta y el sistema se pone en funcionamiento; el itinerario de la ruta se visualiza en el teclado como medida de orientación.

Las operaciones ejecutadas por el conductor mediante el teclado a bordo son desplegadas jerárquicamente, de forma clara sobre la pantalla LCD. Cada operación permite confirmar lo solicitado, de manera que el conductor no maneja información excesiva al tiempo.

Figura 57. Interfaz conductor



Fuente: Autor.

5.2 CONEXIÓN AL PC

Para el enlace del Sistema de información al usuario con el PC, se implementó una comunicación bidireccional masiva, bulk transfer, entre el PIC y el PC vía USB 2.0 a full speed. Se utiliza las funciones USB incorporadas en el lenguaje C del programa CCS, las cuales permiten que el microcontrolador sea reconocido como un dispositivo personalizado usando los descriptores que incluye este lenguaje.

En la comunicación USB se requiere por una parte, del firmware para el PIC en CCS y de otra el software del PC en Visual Basic 6.0. Como intermediarios se utiliza dos recursos que provee Microchip: el driver mchpubs.sys para Windows y el archivo mpubapi.dll.

En CCS, para activar el método de transferencia masiva mediante el USB se debe configurar los endpoint de transmisión y recepción, *USB_EP1_TX_ENABLE* y *USB_EP1_RX_ENABLE*, indicándolo con la constante *USB_ENABLE_BULK*. Es imprescindible deshabilitar el método HID (Human Interface Device). El tamaño del buffer de transferencia es ajustable y se estableció en 64 bytes de envío y recepción.

Las librerías específicas que dispone CCS para comunicación USB son:

usb.c: Maneja las peticiones USB estándar y las interrupciones USB.

usb.h: Librería con el prototipo de funciones, definiciones y variables globales USB, utilizadas en el driver USB.

Las funciones `usb_init()`, `usb_task()`, `usb_wait_for_enumeration()`, `usb_enumerated()`, `usb_kbhit()`, `usb_get_packet()` y `usb_put_packet()` están desarrolladas en el driver que proporciona CCS C para el manejo del USB 2.0 y vienen definidas e implementadas en `pic18_usb.h`, `usb.c` y `usb.h`, las cuales se encuentran en el directorio `..\Drivers` de la carpeta de instalación de CCS.

`usb_init()`, `usb_task()` y `usb_wait_for_enumeration()` se utilizan para establecer la comunicación y se ejecutan únicamente tras un reset del micro. La función `usb_enumerated()` devuelve un true cuando se ha efectuado la conexión y reconocimiento por parte del PC.

A partir de este punto solo resta esperar recibir un comando proveniente del PC; esto se detecta con la función `usb_kbhit()` que devuelve un true para indicar que existen datos pendientes de recibir. Los datos recibidos se guardan mediante `usb_get_packet()`.

Por otra parte, para la comunicación del dispositivo USB con la aplicación software desarrollada en Visual Basic, se requiere de la librería `mpusbapi.dll` que proporciona Microchip y que contiene el conjunto de funciones de acceso al puerto USB con el microcontrolador PIC18F2550.

Los pasos llevados a cabo son los siguientes:

- a. Definir el VID y PID (identificador de vendedor y producto) para poder utilizar las funciones de comunicación sabiendo hacia quién van dirigidas. Instrucción en VB :
`String vid_pid_norm="vid_04d8&pid_0011"`
- b. Definir el endpoint de transmisión y recepción por el que van a pasar los datos. Instrucción en V.B :
`Public Const out_pipe = "MCHP_EP1"`
`Public Const in_pipe = "MCHP_EP1"`
- c. Importar las funciones necesarias para el proyecto desde el archivo `mpusbapi.dll`.

Las funciones utilizadas para llamar a la librería `mpusbapi` desde Visual Basic son:

```
Public Declare Function MPUSBGetDeviceCount Lib "mpusbapi.dll" (ByVal pVID_PID As String) As Long
```

Identifica si hay un dispositivo con el nombre `pVID_PID` conectado a la PC; la variable `pVID_PID`, es una entrada de cadena de caracteres que da como

resultado el número de dispositivos conectados al Host, que tienen asignado el mismo pVID_PID.

Public Declare Function MPUSBOpen Lib "mpusbapi.dll" (ByVal instance As Long, ByVal pVID_PID As String, ByVal pEP As String, ByVal dwDir As Long, ByVal dwReserved As Long) As Long

Permite establecer una tubería pasándole como parámetros el identificador del vendedor de producto, el punto final con el que se requiere comunicar y la dirección de los datos.

Public Declare Function MPUSBRead Lib "mpusbapi.dll" (ByVal handle As Long, ByVal pData As Long, ByVal dwLen As Long, ByVal pLength As Long, ByVal dwMilliseconds As Long) As Long

Public Declare Function MPUSBClose Lib "mpusbapi.dll" (ByVal handle As Long) As Long

Las funciones anteriores permiten leer los datos y cerrar la tubería respectivamente. El parámetro *pVID_PID* es un string que contiene el PID&VID del dispositivo objetivo; el VID es un código de 16 bits que identifica al fabricante del hardware a conectar. En este caso se ha utilizado el número 04D8h que identifica a Microchip. El PID es un código de 16 bits que identifica al dispositivo hardware a conectar, que para los propósitos del proyecto corresponde a 0011h.

El formato corresponde a "vid_xxxx&pid_yyyy"; donde xxxx es el valor del VID y el yyyy el del PID, los dos en hexadecimal. Ejemplo: Si el dispositivo tiene un VID=0x04d8 y un PID=0x0011, el string de entrada es: "vid_04d8&pid_0011"

De las aplicaciones Matriz de Leds y Soft GPS, solo cabe anotar que es fundamental que el VID y PID definidos en éstas coincidan exactamente con el driver de Windows y con el firmware del PIC.

5.3 SOFTWARE

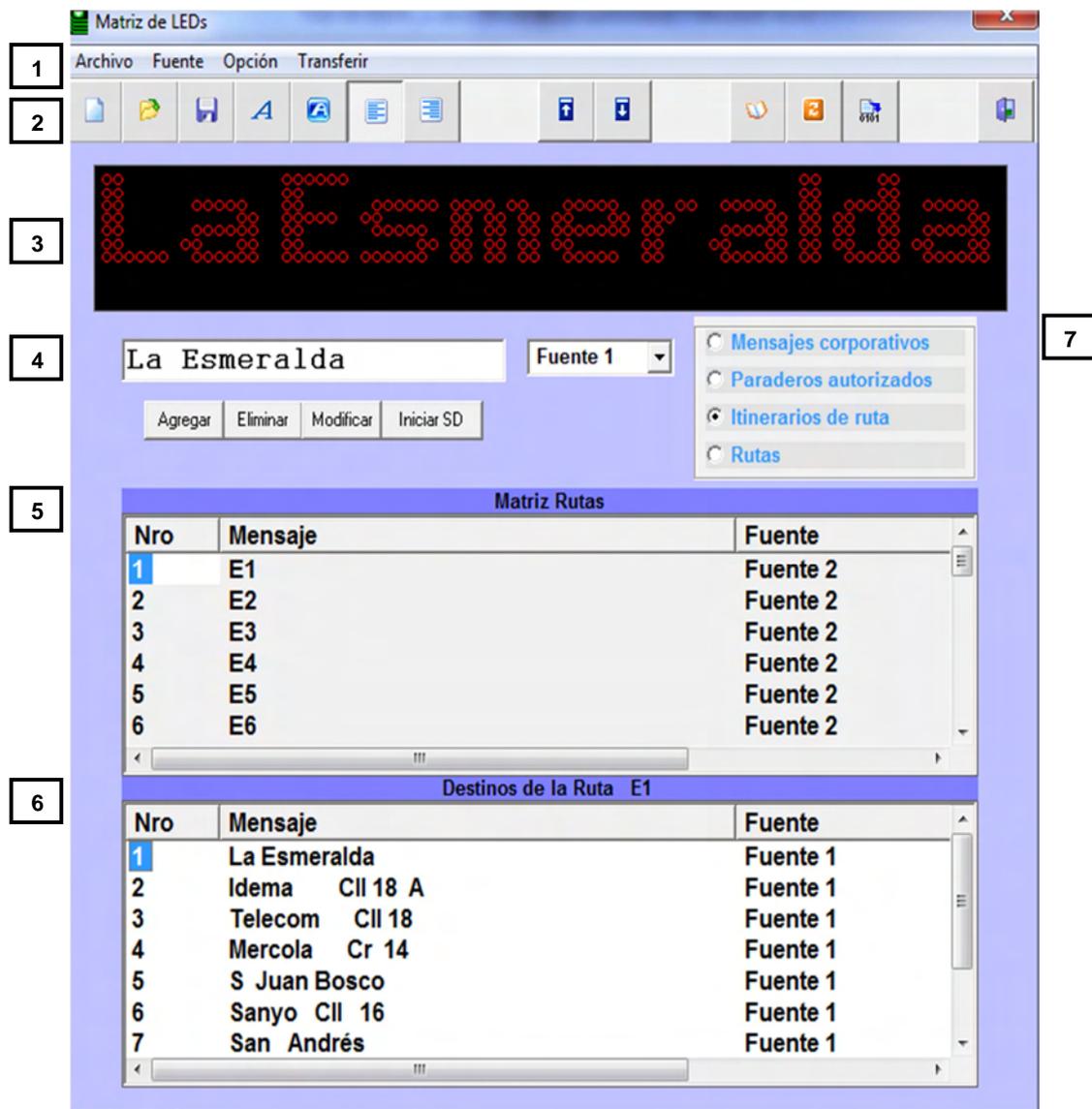
5.3.1 Software editor Matriz de leds. El sistema de información al usuario cuenta con la aplicación software *Matriz de Leds* para realizar la edición de texto de la ruta, itinerarios de ruta y paraderos. El texto es creado en un PC con el programa de edición y los mensajes son enviados por USB a la unidad de control del módulo de señalización audible, en donde se controla la tarjeta de memoria SD. Cuando se termina la edición del texto, se guarda el archivo para realizar posteriormente la transferencia, en este caso, la memoria auxiliar externa de cada matriz se borra eléctricamente para recibir la nueva información.

Para correr el sistema, es necesario de un computador que tenga como mínimo procesador 486 (o superior), 8 Mb RAM, 40 Mb libres en disco, Windows 95, NT o

XP. La programación de mensajes soporta el modo en línea, el cual consiste en el envío en tiempo real de mensajes vía USB.

La figura 58 muestra la ventana al abrir el programa, el cual permite, mediante menús simples, alinear el texto, cambiar las fuentes o tipos de letras y diseñar nuevos caracteres, entre otras opciones, lo que facilita un trabajo rápido.

Figura 58. Ventana principal editor Matriz de led



Fuente: Autor.

La ventana de trabajo del editor cuenta con las siguientes características:

La barra de menús (1), contiene las opciones para trabajar por medio de menús desplegables. Los menús proporcionan acceso a todas las operaciones del software. Estos son: *Archivo, Fuente, Opción y Transferir*. Éstos permiten crear y guardar un archivo de texto, crear y modificar el tipo de caracter, alinear el texto y transferir los itinerarios hacia el circuito de control mediante conexión USB.

La barra de herramientas (2), contiene botones que permiten tener un rápido acceso a algunas de las opciones y funciones. Éstos son: *crear, abrir y guardar archivo; fuente, alineación derecha, izquierda (por defecto), enviar itinerarios y salir del programa*.

La visualización del mensaje se hace a través de la pantalla (3), la cual es una representación de la matriz de leds. La casilla de texto (4) presenta en pantalla el cursor como punto de inserción de texto, el cual se introduce desde el teclado; la lista desplegable de fuentes permite elegir el tipo de fuente. Los botones *agregar, modificar y eliminar* permiten respectivamente, almacenar el texto creado o editado en el archivo que se está creando, modificarlo para realizar algún cambio y eliminarlo de la lista (5).

La lista (5) muestra las rutas creadas y en la lista (6) se encuentran los itinerarios asociados a cada ruta. Mediante las opciones (7) se elige sobre qué matriz se va a realizar la edición del contenido.

Figura 59. Ventana crear-modificar fuentes de texto



Fuente: Autor.

En la barra de herramientas el botón fuente permite abrir una ventana auxiliar, la cual se muestra en la figura 59, para crear, modificar y eliminar caracteres. El tipo de fuente se cambia seleccionando una nueva fuente a partir de la lista desplegable ubicada en la parte superior de la ventana. Es posible crear y modificar un carácter pudiendo elegir el tamaño de éste y asignándole su respectivo valor ASCII.

5.3.2 Software Interfaz Sistema GPS. El dispositivo cuenta con conexión USB por medio de la cual se realiza la interfaz con el computador; el puerto USB es el medio de comunicación por el cual son enviados a la tarjeta de memoria SD los archivos de audio y las coordenadas a comparar.

La aplicación software tiene como función principal acceder a la memoria SD a través del microcontrolador para enviar las coordenadas que el GPS registra en una determinada parada. La figura 60 muestra la ventana principal del software *Sistema GPS*, se trata de una aplicación muy intuitiva y sencilla que cuenta esencialmente con 3 comandos los cuales son desplegados en 3 ventanas diferentes (Figura 60): *ver formato wav*, *coordenadas actuales* y *enviar wav/coordenadas*.

Figura 60. Ventana principal del software Sistema GPS

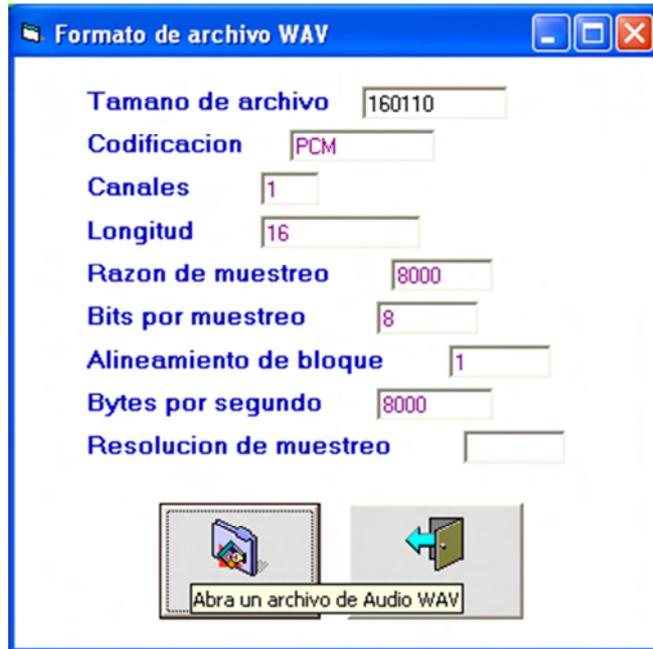


Fuente: Autor.

Ventana *Ver formato wav*. (Figura 61). Permite abrir un archivo de extensión .wav

y ver las propiedades del formato, entre éstas, tamaño del archivo, tipo de codificación, número de canales empleados en la grabación, frecuencia de muestreo, número de bits por muestra, alineación de bloque, bytes por segundo y resolución.

Figura 61. Ventana propiedades del archivo wav



Fuente: Autor.

Ventana *Coordenadas GPS actuales*. (Figura 62). Permite acceder a los datos de las coordenadas del GPS y visualizarlos. En la parte superior de la ventana se muestra la trama interpretada NMEA GPGGA la cual contiene los datos de posición: latitud (N/S) y longitud (W/E). También se muestra el estado del GPS el cual puede ser enlazado o sin satélites. Cuando el GPS se encuentra enlazado, y se da clic sobre el botón *Leer GPS* se visualiza, en la ventana, las coordenadas actuales a las cuales se les debe asignar el nombre de la parada, posteriormente se agregan a la lista para ser asociadas con el archivo de audio correspondiente.

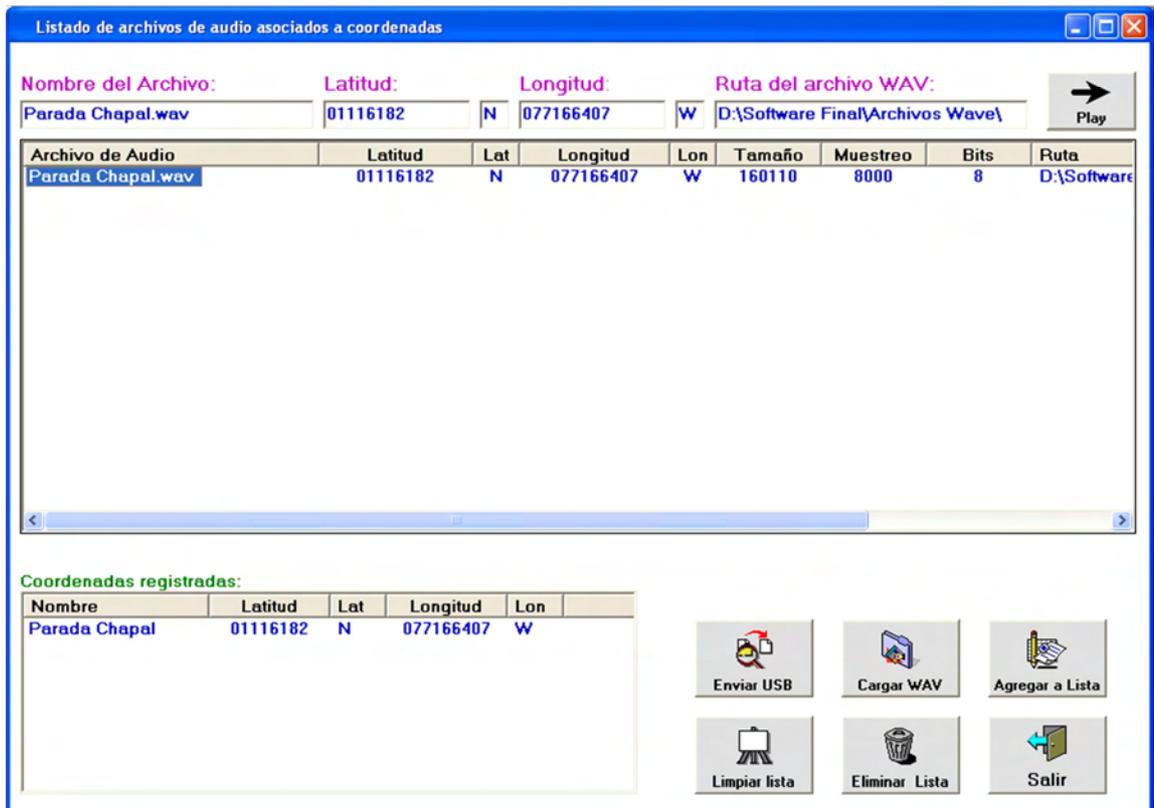
Ventana *Enviar coordenadas/wav*. (Figura 63). Permite asociar el archivo de audio en formato wav con las coordenadas registradas y almacenadas anteriormente. De esta forma, se selecciona la coordenada registrada y mediante el comando *cargar wav* se abre el archivo de audio asociado a dichas coordenadas, se agrega a lista para posteriormente enviar por el puerto USB la información concerniente a la próxima parada.

Figura 62. Ventana Coordenadas actuales



Fuente: Autor.

Figura 63. Ventana enviar Wav/coordenadas



Fuente: Autor.

6. CONCLUSIONES

La modernización del transporte público de pasajeros de la ciudad de Pasto requiere el apoyo de herramientas automatizadas de gestión de la información, para mejorar la calidad de los servicios ofrecidos a los usuarios. En este contexto, se ha diseñado una herramienta que permite gestionar integralmente la señalética del servicio de transporte público de la ciudad.

El sistema de información al usuario está conformado esencialmente por dos componentes que se interconectan en forma coordinada e integral. Por un lado el componente de información visual cuyo objetivo principal es proveer información a usuarios potenciales y pasajeros a bordo, a través de paneles de información variable; y por otro lado el componente de información audible cuyo objetivo principal es entregar información de interés para el pasajero a bordo del bus mediante megafonía. La herramienta cuenta, entre otras características, con una construcción robusta orientada al uso en ambientes con alta vibración, leds de alto brillo para uso exterior y software con editor de textos, gráficos y tipografías.

En este sentido, el sistema de información al usuario permite gestionar la señalética del servicio de transporte y la socialización del SETP en la ciudad, promoviendo que los mismos usuarios sean multiplicadores, en sus comunidades, del cumplimiento de las normas de tránsito, y el desarrollo de procesos de corresponsabilidad entre la ciudadanía y la Administración municipal en relación con la seguridad vial. La señalización audible, por su parte, se convierte en una estrategia de difusión y cultura ciudadana para la movilidad mediante la cual se fomenta no sólo el uso adecuado del sistema, sino que también brinda apoyo a los usuarios con necesidades diferentes, por ejemplo a personas con capacidades reducidas que tienen una mayor necesidad de información y orientación en su recorrido.

Tanto en la etapa de diseño como en la etapa de construcción, se tuvo en cuenta los lineamientos dispuestos por el Gobierno Nacional y Municipal, en los documentos Conpes. El resultado obtenido da claras muestras de un uso eficiente de los recursos que provee la región, logrando implementar una herramienta de señalización embarcada que brinda la información necesaria con un mínimo de elementos hardware en comparación con los dispositivos disponibles en el mercado. Esto se debe básicamente, al método utilizado en el diseño del programa de control, para formar una imagen mediante tecnología multiplexada, utilizando pulsos cortos de corriente en conjunto con un barrido escaneado de toda la matriz, lo que redujo notablemente el firmware y hardware, con la consiguiente ventaja de simplicidad en el armado y mayor confiabilidad.

Finalmente, cabe mencionar que ciertas implementaciones tecnológicas necesarias para la ciudad, pueden ser realizadas por manos y talento regional,

puesto que esto permite adaptar los aportes del progreso y la globalización e introducir y conjugar el diseño con los criterios propios de la cultura. En la realización de este proyecto se ha tenido en cuenta las expectativas que el transporte despierta en los ciudadanos y/o usuarios, como una condición fundamental para tener éxito frente a objetos aparentemente aprobados y comprobados en países desarrollados.

7. RECOMENDACIONES

La constante de conectividad con el pic, VIP&PID debe ser correctamente configurada tanto en Visual Basic, en el módulo que contiene las rutinas de llamada a la librería Mpusbapi.ll, como en CCS, en el archivo cabecera de extensión .h (picusbnut.h) que contiene los identificadores con los que el dispositivo se presenta ante Windows. Igualmente, ésta constante debe coincidir con el driver mchpusb.inf que es el que le permite al sistema operativo reconocer el dispositivo USB. Para esto se debe sustituir el valor resaltado que se comenta en los extractos de código que se anexan en el apartado B de éste documento. Si no se sustituye correctamente las líneas que se comentan en el código del programa, al conectarse el PIC en la PC, éste no reconocerá el dispositivo.

Como medidas de acción del sistema, se recomienda incluir la supervisión, el registro y la transferencia a la unidad de control centralizada, de la información concerniente a detenciones, velocidad y recorrido, con el ánimo de registrar las detenciones en puntos no autorizados, excesos de velocidad y cumplimiento de los recorridos de los buses durante la prestación del servicio.

8. BIBLIOGRAFÍA

ANGÚLO, José María. Microcontroladores Pic: Diseño práctico de aplicaciones. Madrid: McGraw Hill, 2003. p.9.

APACER TECHNOLOGY. Application note secure digital card interface. Taipei, Taiwán, 2005. 100p.

BADDELEY, Glenn. GPS NMEA sentence information [en línea]. <<http://home.pacific.net.au/~gnb/gps/nmea.html>> [Citado el 20 de Junio de 2011]

BORN, Gunter. Wav file format description [en línea]. <http://www.tav.net/audio/wav_format.htm> [Citado el 26 de mayo de 2011]

BRAIN, Marshall. How GPS receivers works [en línea]. <<http://electronics.howstuffworks.com/gadgets/travel/gps.htm>> [Citado el 15 de Junio de 2011]

CALI. APÉNDICE II DEL CONTRATO. Especificaciones técnicas de los autobuses del Sistema Integrado de Transporte Masivo de Occidente MIO. Cali, 2007.

COLOMBIA. DEPARTAMENTO ADMINISTRATIVO NACIONAL DE ESTADÍSTICA. Proyecciones de población municipales por área 2005-2020. Bogotá, 2005.

COLOMBIA. MINISTERIO DE TRANSPORTE. Conpes 3167: Política para mejorar el servicio de transporte público urbano de pasajeros. Bogotá, 2002. 50p.

COLOMBIA. MINISTERIO DE TRANSPORTE. Conpes 3167: Política para mejorar el servicio de transporte público urbano de pasajeros. Bogotá, 2002. 50p.

COLOMBIA. MINISTERIO DE TRANSPORTE. Conpes 3549: Sistema estratégico de transporte público de pasajeros para la ciudad de Pasto. Bogotá, 2008. 45p.

COLOMBIA. MINISTERIO DE TRANSPORTE. Conpes 3549: Sistema estratégico de transporte público de pasajeros para la ciudad de Pasto. Bogotá, 2008. 45p.

DJORJALIAN, Andrés. Características de los letreros electrónicos de matriz de Leds [en línea]. <<http://www.indicart.com.ar/letreros-electronicos.htm>> [Citado el 20 de Mayo de 2011]

DOMÍNGUEZ, Fernando. Lógica digital y microprogramable. Madrid: Mc Graw Hill, 1999.

GONZÁLES, Juan José. Instrumentación electrónica: conversión Digital/Analógica [en línea]. <http://www.uca.es/grupnvest/instrument_electro/T10_CDA.pdf> [Citado el 27 de Mayo de 2011]

GONZÁLEZ, José Adolfo. Introducción a los microcontroladores. Madrid: McGraw Hill, 2002.

KINGMAX DIGITAL INC. Secure digital memory card: SD card specification [en línea]. <<http://downloads.amilda.org/MODs/SDCard/SD.pdf>> [Citado el 20 de mayo de 2011]

MAGELLAN CORPORATION. Data transmission protocol specification [en línea]. <<http://www.magellangps.com>> [Citado el 10 de Julio de 2011]

MARTÍNEZ, Arturo. Nuevas aplicaciones tecnológicas para la mejora de la eficiencia energética en el transporte: Información al usuario de transporte público. Pamplona, 2007.

MIYARA, Federico. Preguntas frecuentes sobre audio digital [en línea]. <<http://www.cetear.com/FAQad.pdf>> [Citado el 22 de Mayo de 2011]

NATIONAL MARINE ELECTRONICS ASSOCIATION. NMEA Protocol [en línea]. <<http://www.nmea.org>> [Citado el 16 de Junio de 2011]

PASTO. ALCALDÍA MUNICIPAL. Decreto 0735 de 2009: Sistema Estratégico de Transporte Público en el municipio de Pasto. Pasto, 2009. 12p.

PERSISTENCIA RETINIANA [en línea]. <<http://www.elmulticine.com/glosario2.php?orden=130>> [Citado el 15 de mayo de 2011]

PROAKIS, J. G. Tratamiento digital de señales: Principios, algoritmos y aplicaciones. Madrid: Prentice Hall Internacional Ltd., 1998.

SANDISK CORPORATION. Sandisk secure digital card. Estados Unidos, 2003. 113p.

TIMONEDA, José Miguel. Archivos de sonido digital [en línea]. <<http://cursoryc.educa.aragon.es/comunes/modulo3/unidad2.htm>> [Citado el 26 de mayo de 2011]

TRONCHONI, Pepe. Formato de los ficheros de sonido wav [en línea]. <http://www.tav.net/audio/formato_wav.htm> [Citado el 26 de mayo de 2011]

ANEXO A. FORMATOS DE SENTENCIAS NMEA

SENTENCIA GGA (Global Positioning System Fix Data)	
Formato:	\$GPGGA,hhmmss.ss,IIII.II,a,mmmm.mm,b,x,yy,z.z,c.c,M,v.v,M,d.d,wwww*hh
hhmmss.ss	Hora UTM a la que tuvo lugar la toma de datos
IIII.II	Latitud en grados y minutos
a	Tipo de latitud. Norte (N) ó Sur (S)
mmmm.mm	Longitud en grados y minutos
b	Tipo de longitud. Este (E) ú Oeste (W)
x	Calidad de la señal GPS. (0 = no válida; 1= corrección GPS; 2 = DGPS)
yy	Número de satélites en uso
z.z	HDOP
c.c	Altitud sobre el nivel del mar
M	Unidad de medida de la altitud. Metros (m) por defecto
v.v	Distancia entre el elipsoide WGS-84 y el nivel del mar
M	Unidad de medida de la distancia. Metros (m) por defecto
d,d	Tiempo desde la última actualización DGPS (segundos)
wwww	Identificación de la estación DGPS
*hh	Palabra de comprobación o checksum

SENTENCIA GSA (GPS DOP and active satellites)	
Formato:	\$GPGSA,x,y,z1,z2,z3,z4,z5,z6,z7,z8,z9,z10,z11,z12,p.p,h.h,v.v*ff
x	Modo de operación 2D\3D. Automático (A) o Manual (M)
y	Latitud en grados, minutos y segundos
z1 a z12	Identificación PRN de los satélites usados para la corrección, máximo 12.
p.p	PDOP
h.h	HDOP
v.v	VDOP
*ff	Palabra de comprobación o checksum

SENTENCIA GSV (Satellites in view)	
Formato:	\$GPGSV,x,y,zz,aa,ee,rr,ii,s1,s2,s3,s4,t1,t2,t3,t4,c1,c2,c3,c4*ff
x	Número total de mensajes en el ciclo
y	Número de mensaje con respecto al total
zz	Número total de satélites a la vista
aa	Identificación PRN del satélite
ee	Elevación en grados
rr	Azimut en grados
ii	Intensidad de la señal en decibelios
s1	Identificación PRN del satélite 2
s2	Elevación en grados del satélite 2
s3	Azimut en grados del satélite 2
s4	Intensidad en grados del satélite 2
t1	Identificación PRN del satélite 3
t2	Elevación en grados del satélite 3
t3	Azimut en grados del satélite 3
t4	Intensidad en grados del satélite 3
c1	Identificación PRN del satélite 4
c2	Elevación en grados del satélite 4
c3	Azimut en grados del satélite 4
c4	Intensidad en grados del satélite 4
*ff	Palabra de comprobación o checksum

SENTENCIA GLL (Geographic position, latitude and longitude)	
Formato:	\$GPGLL,III.II,a,mmmm.mm,b,hhmmss.ss,A*ff
III.II	Latitud en grados y minutos
a	Tipo de latitud. Norte (N) ó Sur (S)
mmmm.mm	Longitud en grados y minutos
hhmmss.ss	Hora UTM a la que tufo lugar la toma de datos
A	Validez de los datos. (A = válido; V= no válido)
*ff	Palabra de comprobación o checksum

ANEXO B. CÓDIGO FUENTE

```
;*****
;***** PRINCIPALES RUTINAS MATRIZ DE LEDS *****

LIST P=16F887
INCLUDE<P16F887.INC>

;*****
;***** * FUSIBLES *****

__CONFIG    _LVP_OFF & _FCMEN_OFF & _IESO_OFF
__CONFIG    _BOR_OFF & _CPD_OFF & _CP_OFF & _MCLRE_OFF
__CONFIG    _PWRTE_ON & _WDT_OFF & _INTRC_OSC_NOCLKOUT
__CONFIG    _CONFIG2, _WRT_OFF & _BOR21V

CBLOCK      0X20
DCOL1
ENDC

CBLOCK      0X79
REG_N
REG_I
AUX
COLUMNAS
AUX1
CONT
CONT1
ENDC

;*****
;***** ASIGNACIÓN DE PUERTOS *****

#define      MUX_A          PORTA
#define      ENB_MUX       PORTA, 4
#define      DATOS_B       PORTC
#define      ULN_D         PORTD

;*****
;***** MACROS *****

BANCO0
MACRO
BCF          STATUS, RP0
BCF          STATUS, RP1
ENDM
BANCO1
MACRO
BSF          STATUS, RP0
BCF          STATUS, RP1
ENDM
```

```

BANCO2
MACRO
BSF          STATUS, RP0
BSF          STATUS, RP1
ENDM
BANCO3
MACRO
BSF          STATUS, RP0
BSF          STATUS, RP1
ENDM

;*****
;*****

ORG          0X000
CONFIGURACION
BANCO1
CLRF        TRISA
CLRF        TRISB
CLRF        TRISC
CLRF        TRISD
CLRF        TRISE
BANCO3
CLRF        ANSEL
CLRF        ANSELH
BANCO0
CALL        BORRA_REGISTROS

INICIO
CALL        INICIO_LEE_EEPROM
UNO        MOVLW        .100
           MOVWF        CONT1
           CALL        CARGAR_RAM
DOS        CALL        INICIO_VISUALIZA
           CALL        VISUALIZA
           DECFSZ       CONT1,1
           GOTO        DOS
           MOVLW        B'01010000'
           ADDWF        POSICION,1
           BTFSC       STATUS,C
           INCF        PAGINA,1
           INCF        CONT,1
           MOVLW        .15
           SUBWF       CONT,W
           BTFSC       STATUS,Z
           GOTO        COMENZAR_DENUEVO
           GOTO        UNO

COMENZAR_DENUEVO
CLRF        M24LC256_POSICION
CLRF        M24LC256_PAGINA
CLRF        CONT
           GOTO        UNO

```

```

;*****
;***** SUBRUTINAS VISUALIZA *****

```

```

INICIO_VISUALIZA

```

```

    CLRFB    MUX_A
    CLRFB    DATOS_B
    CLRFB    PORTC
    CLRFB    ULN_D
    CLRFB    REG_N
    CLRFB    AUX1
    MOVLW    .1
    MOVWF    REG_I
    RETURN

```

```

VISUALIZA

```

```

    MOVLW    20H
    ADDWF    AUX1,0
    MOVWF    FSR

```

```

NEXT_L

```

```

    CLRFB    ULN_D
    MOVFB    INDF,W
    MOVWF    DATOS_B
    CALL    CLOCK_LATCH
    MOVLW    .10
    SUBWF    REG_N,W
    BTFSC    STATUS,Z
    GOTO    ACTIVA_COLUMNNA
    GOTO    NEXT_L

```

```

ACTIVA_COLUMNNA

```

```

    CLRFB    REG_N
    MOVFB    REG_I,0
    MOVWF    ULN_D
    CALL    RETARDO_1MS
    MOVLW    B'10000000'
    SUBWF    REG_I,W
    BTFSC    STATUS,Z
    RETURN
    RLF     REG_I,1
    INCF    AUX1,1
    GOTO    VISUALIZA

```

```

CLOCK_LATCH

```

```

    MOVFB    REG_N,0
    MOVWF    MUX_A
    BCF     ENB_MUX
    NOP
    BSF     ENB_MUX
    INCF    REG_N,1
    MOVLW    .8
    MOVWF    AUX

```

```

NEXT_I

```

```

    INCF    FSR,1
    DECFSZ  AUX,1
    GOTO    NEXT_I
    MOVFB    FSR,0

```

```

MOVWF      AUX
RETURN

;*****
;***** SUBROUTINAS LECTURA MEMORIA EEPROM *****
;*****

INICIO_LEE_EEPROM
    CLRF      PÁGINA
    CLRF      POSICIÓN
    RETURN

CARGAR_RAM
    MOVLW     70H
    MOVWF     COLUMNAS
    CALL      MENSAJE_A_MATRIZ
    MOVWF     DCOL1
    MOVLW     21H
    MOVWF     FSR
NEXT      CALL      LEE_OTROBYTE
    MOVWF     INDF
    INCF      FSR, 1
    MOVF      FSR, W
    SUBWF     COLUMNAS, W
    BTFSS     STATUS, Z
    GOTO      NEXT
    CALL      FINALIZA_LECTURA
    RETURN

INICIA_LECTURA
    BCF      ULTIMOBYTE
    CALL     RETARDO_5MICROS
    CALL     I2C_BITSTART
    MOVLW   DIR_ESCRITURA
    CALL     I2C_ENVIARDATO
    MOVF    PÁGINA, W
    CALL     I2C_ENVIARDATO
    MOVF    POSICIÓN, W
    CALL     I2C_ENVIARDATO
    CALL     I2C_BITSTOP
    CALL     I2C_BITSTART
    MOVLW   DIR_LECTURA
    CALL     I2C_ENVIARDATO
    RETURN

MENSAJE_A_MATRIZ
    BCF      ULTIMOBYTE
    CALL     INICIA_LECTURA

LEE_OTROBYTE
    CALL     I2C_LEERDATO
    RETURN

FINALIZA_LECTURA
    BSF      ULTIMOBYTE
    CALL     RETARDO_5MICROS
    CALL     I2C_LEERDATO
    CALL     I2C_BITSTOP
    RETURN

```

```

//*****
//***** PRINCIPALES RUTINAS SD-GPS *****
//*****

#include <18F2550.H>

#define DELAY(CLOCK=4800000)
#define FAST_IO(A)
#define FAST_IO(B)
#define FAST_IO(C)
#define FAST_IO(E)

//*****
//***** FUSIBLES *****
//*****

#define HSPLL // Activación del PLL interno
#define USBDIV
#define PLL5 // Div 20Mhz/5 para generar 4Mhz
#define CPUDIV1 // 4Mhz x 24 = 96Mhz /2 = 48Mhz clock CPU
#define VREGEN
#define NOMCLR
#define NOWDT
#define NOPROTECT
#define PUT
#define NOLVP
#define NOCPD

//*****
//***** ASIGNACIÓN DE PUERTOS *****
//*****

/*
RB0 RESERVADO PARA SDI
RB1 RESERVADO PARA SCL
RC7 RESERVADO PARA SDO
RB2 RESERVADO PARA RX
RB3 RESERVADO PARA TX
*/
#define LED_GPS PIN_A4
#define LED_CPU PIN_B7

#define RAM_RESET PIN_C0
#define RAM_WRITE_P PIN_C1
#define RAM_CS PIN_C2

#define SD_INSERT PIN_A1
#define SD_LOCK PIN_A2
#define CS_SD PIN_A5

#define GPS_ENABLE PIN_A3
#define DAC_CS PIN_B4
#define DAC_LDAC PIN_B5
#define PULSADOR PIN_E3 // Función USB

```

```

// Definiciones comunicación USB bulk transfer

#define USB_HID_DEVICE FALSE
//deshabilita el uso de las directivas HID
#define USB_EP1_TX_ENABLE USB_ENABLE_BULK
// Habilita el endpoint1 EP1/entrada Tx
#define USB_EP1_RX_ENABLE USB_ENABLE_BULK
// Habilita el endPoint1 EP1/salida Rx
#define USB_EP1_TX_SIZE 64
// tamaño del buffer de transmisión de 8bytes
#define USB_EP1_RX_SIZE 64
// tamaño del buffer de recepción de 8bytes

//*****
//***** MODOS STÁNDAR SPI *****

VOID SETUP_SPI_MODO_0(VOID)
{
    SSPCON1=0B00100000;
    // activa spi por hardware, maestro en fosc/4, reloj idle en bajo
    // ckp=0, d4=0, polaridad del clock cuando no se este transmitiendo data
    // d3 a d0 elige el modo de trabajo maestro/esclavo y velocidad del clock

    SSPSTAT=0B010000000;
    // se muestrea en el medio del dato (smp=0) d7 = 0
    // la transmisión ocurre en flanco bajada (cke=1) d6=1
    // los bits d1 a d5 solo se usan en i2c
    // d0 responde 1 si el buffer recibio datos
}

//*****
//***** FUNCIÓN STAR_UP_SD *****

INT START_UP_SD()
{
    INT T;
    LONG LTIEMPO_FUERA=0;
    INT RESPUESTA;
    OUTPUT_HIGH(CS_SD);

    DELAY_US(100); // POWER-UP TIMER
    FOR(T=0;T<12;T++)
        LEER_SPI();
    DO
    {
        RESPUESTA = COMANDO_SD(0X40);
        DELAY_US(100);
        IF(LTIEMPO_FUERA++ > LTIMEOUT)
            RETURN(RESPUESTA); // retorna el status command (R1)
    }
    WHILE(RESPUESTA != 1); // 1= estado idle inicialización en progreso

    LTIEMPO_FUERA=0;
    DO
    {

```

```

        RESPUESTA = COMANDO_SD(0X41);
        DELAY_US(100);
        IF(LTIEMPO_FUERA++ > LTIMEOUT)
            RETURN(RESPUESTA); // retorna el status command (R1)
    }
    WHILE(RESPUESTA != 0); // 1= estado idle inicialización en progreso
    OUTPUT_HIGH(CS_SD);
    RETURN(0);
}

//*****
//***** FUNCIÓN COMANDO_SD *****

INT COMANDO_SD(INT COMANDO)
{
    INT IRESPUESTA = 0XFF;
    LONG LTIEMPO_FUERA = 0;
    OUTPUT_LOW(CS_SD);

    LEER_SPI();
    ESCRIBIR_SPI(COMANDO); // ÍNDICE DE COMANDO
    ESCRIBIR_SPI(0); // BYTE MSB DE DIRECCIÓN
    ESCRIBIR_SPI(0);
    ESCRIBIR_SPI(0);
    ESCRIBIR_SPI(0); // BYTE LSB DE DIRECCIÓN
    ESCRIBIR_SPI(0X95); // VALOR CRC
    WHILE(IRESPUESTA == 0XFF)
    {
        IRESPUESTA = LEER_SPI();
        IF(LTIEMPO_FUERA++ > 500)
            BREAK;
    }
    RETURN(IRESPUESTA);
}

//*****
//***** FUNCIÓN LEER_SD_SECTOR *****

SHORT LEER_SD_SECTOR(INT32 DIR_SECTOR)
{
    INT32 ADDRESS;
    IF (SD_INSERTADA() == TRUE)
    {
        ADDRESS = DIR_SECTOR * (INT32)512;
        IF (LEER_SD_ADRESS(ADDRESS) == TRUE)
            RETURN(TRUE);
        ELSE
            RETURN(FALSE);
    }
    ELSE
        RETURN(FALSE);
}

//*****

```

```

//***** FUNCIÓN LEER_SD_ADRESS *****
SHORT LEER_SD_ADRESS(INT32 ADRESS)
{
    UNSIGNED LONG I;
    LONG LTIEMPO_FUERA=0;
    INT RESPUESTA; // guarda respuesta

    DO
    {
        RESPUESTA = COMANDO_LEE(ADRESS);
        DELAY_US(100);
        IF(LTIEMPO_FUERA++ > LTIMEOUT)
        {
            OUTPUT_HIGH(CS_SD);
            RETURN(FALSE);
        }
    }
    WHILE(RESPUESTA != 0 );

    WHILE(LEER_SPI()!=0XFE);
    FOR(I = 0 ; I <= 511 ; I++)
    SD_SECTOR[I]= LEER_SPI();
    LEER_SPI(); // lee dos bytes CRC
    LEER_SPI();
    OUTPUT_HIGH(CS_SD);
    LEER_SPI();
    RETURN(TRUE);
}

//***** FUNCIÓN LEER_SPI *****

INT LEER_SPI()
{
    SSPBUF=0XFF;
    WHILE(!BIT_TEST(PIR1,3));
    // mientras sspif=0 la transmisión continúa
    // si sspif=1, termina la transmisión
    BIT_CLEAR(PIR1,3); // borra el flag sspif
    RETURN(SSPBUF);
}

//***** FUNCIÓN ESCRIBIR_SD_SECTOR *****

INT ESCRIBIR_SD_SECTOR(INT32 DIR_SECTOR)
{
    INT32 ADRESS;
    IF (SD_INSERTADA() == TRUE)
    {
        IF (SD_WRITER_PROTECTED() == FALSE)
        {

```

```

        ADDRESS = DIR_SECTOR * (INT32)512;
        IF (ESCRIBIR_SD_ADRESS(ADDRESS) == TRUE)
            RETURN(TRUE);
        ELSE
            RETURN(FALSE);
    }
    ELSE
        RETURN(FALSE);
}
ELSE
    RETURN(FALSE);
}

//*****
//***** FUNCIÓN ESCRIBIR_SD_ADRESS *****
//*****

INT ESCRIBIR_SD_ADRESS(INT32 ADDRESS)
{
    UNSIGNED LONG I;
    LONG LTIEMPO_FUERA=0;
    INT RESPUESTA;
    OUTPUT_LOW(CS_SD);
    LEER_SPI();
    DO
    {
        RESPUESTA = COMANDO_ESCRIBE(ADDRESS);
        DELAY_US(100);
        IF(LTIEMPO_FUERA++ > LTIMEOUT)
        {
            OUTPUT_HIGH(CS_SD);
            RETURN(FALSE);
        }
    }
    WHILE(RESPUESTA != 0 );

    FOR (I = 1 ; I <= 3 ; I++) // leer 3 rlb
        LEER_SPI();
    ESCRIBIR_SPI(0XFE); // 1 byte inicio de bloque de 512

    FOR(I = 0 ; I <= 511 ; I++)
        ESCRIBIR_SPI(SD_SECTOR[I]);

    ESCRIBIR_SPI(0XFF); // 2 BYTES CRC
    ESCRIBIR_SPI(0XFF);

    LTIEMPO_FUERA=0;
    DO
    {
        RESPUESTA = ( LEER_SPI() & 0B00001111);
        DELAY_US(100);
        IF(LTIEMPO_FUERA++ > LTIMEOUT)
        {
            OUTPUT_HIGH(CS_SD);

```

```

        LEER_SPI();
        RETURN(FALSE);
    }
}
WHILE ( (RESPUESTA != 0B00000101)

OUTPUT_HIGH(CS_SD);
LEER_SPI();
IF (RESPUESTA == 0B00000101)
    RETURN(TRUE);
ELSE
    RETURN(FALSE);
}

//*****
//***** FUNCIÓN COMANDO_LEE *****

INT COMANDO_LEE(INT32 ADDRESS)
{
    INT IRESPUESTA = 0XFF;
    LONG LTIEMPO_FUERA = 0;

    OUTPUT_LOW(CS_SD);
    LEER_SPI();

    ESCRIBIR_SPI(0X51);
    ESCRIBIR_SPI(MAKE8(ADDRESS,3));
    ESCRIBIR_SPI(MAKE8(ADDRESS,2));
    ESCRIBIR_SPI(MAKE8(ADDRESS,1));
    ESCRIBIR_SPI(MAKE8(ADDRESS,0));
    ESCRIBIR_SPI(0XFF); //CRC

    WHILE(IRESPUESTA == 0XFF)
    {
        IRESPUESTA = LEER_SPI();
        IF(LTIEMPO_FUERA++ > 500)
            RETURN(IRESPUESTA);
    }
    RETURN(IRESPUESTA);
}

//*****
//***** FUNCIÓN COMANDO_ESCRIBE *****

INT COMANDO_ESCRIBE(INT32 ADDRESS)
{
    INT IRESPUESTA = 0XFF;
    LONG LTIEMPO_FUERA = 0;

    ESCRIBIR_SPI(0X58);
    ESCRIBIR_SPI(MAKE8(ADDRESS,3));
    ESCRIBIR_SPI(MAKE8(ADDRESS,2));
    ESCRIBIR_SPI(MAKE8(ADDRESS,1));
    ESCRIBIR_SPI(MAKE8(ADDRESS,0));

```

```

    ESCRIBIR_SPI(0XFF); //CRC

    WHILE(IRESPUESTA == 0XFF)
    {
        IRESPUESTA = LEER_SPI();
        IF(LTIEMPO_FUERA++ > 500)
            RETURN(IRESPUESTA);
    }

    RETURN(IRESPUESTA);
}

//*****
//***** FUNCIÓN TASK_RX_GPS *****

SHORT TASK_RX_GPS(VOID)

{
    STATIC ENUM{RECIBIR=0,ES_GPGGA=1,VERIFICAR=2} ESTADO=0;

    SWITCH(ESTADO)
    {
        CASE RECIBIR: // crea una trama nueva

            IF (CREAR_TRAMA() == TRUE)
                ESTADO = ES_GPGGA;

            BREAK;

        CASE ES_GPGGA: // verifica si la trama inicia con $GPGGA

            IF (TRAMA_GPGGA() == TRUE)
                ESTADO = VERIFICAR;
            ELSE
                ESTADO = RECIBIR;

            BREAK;

        CASE VERIFICAR: // verifica si la trama esta completa

            IF (TRAMA_COMPLETA() == TRUE)
            {
                TRAMA_INIT();
                RETURN(TRUE);
            }

            BREAK;
    }
    RETURN(FALSE);
}

//*****
//***** FUNCIÓN TRAMA_GPGGA *****

CONST CHAR TRAMA_COMP[]={"$GPGGA"}

```

```

SHORT TRAMA_GPGGA(VOID)
{
  IF ( POINTER_TRAMA >= 6 )
  {
    IF ((TRAMA_IN[0]==TRAMA_COMP[0]) && (TRAMA_IN[1]==TRAMA_COMP[1])
      && TRAMA_IN[2]==TRAMA_COMP[2]))
    {
      IF ((TRAMA_IN[3]==TRAMA_COMP[3]) && (TRAMA_IN[4]==TRAMA_COMP[4])
        && (TRAMA_IN[5]==TRAMA_COMP[5]))
        RETURN (TRUE);
      ELSE
        TRAMA_INIT();
    }
    ELSE
      TRAMA_INIT();
  }
  RETURN(FALSE);
}

//*****
//***** FUNCIÓN TRAMA_COMPLETA *****

SHORT TRAMA_COMPLETA(VOID)
{
  LONG POS;

  IF((TRAMA_IN[0]=='$') && (TRAMA_IN[POINTER_TRAMA-1]=='*'))
  {
    TRAMA_FINAL_INIT();
    FOR(POS=0 ; POS < POINTER_TRAMA ; POS++)
    {
      IF (POS < SIZE_TRAMA_FIN)
      {
        TRAMA_FINAL[POINTER_FINAL] = TRAMA_IN[POS];
        POINTER_FINAL = POINTER_FINAL + 1;
      }
    }
    RETURN(TRUE);
  }
  ELSE
    RETURN(FALSE);
}

//*****
//***** FUNCIÓN EXTRAER_COORDENADAS *****

// Trama requerida Ej: $GPGGA,161054.000,0112.2241,N,07717.5956,W,1,06

VOID EXTRAER_COORDENADAS(VOID)
{
  LATITUD_DEC = CONV_ASCII_COORDENADAS(TRAMA_FINAL[18],TRAMA_FINAL[19],
    TRAMA_FINAL[20],TRAMA_FINAL[21],TRAMA_FINAL[23],TRAMA_FINAL[24],
    TRAMA_FINAL[25],TRAMA_FINAL[26]);
}

```

```

LONGITUD_DEC = CONV_ASCII_COORDENADAS(TRAMA_FINAL[30],TRAMA_FINAL[31],
    TRAMA_FINAL[32],TRAMA_FINAL[33],TRAMA_FINAL[34],TRAMA_FINAL[36],
    TRAMA_FINAL[37],TRAMA_FINAL[38],TRAMA_FINAL[39]);

CHAR_LATITUD = TRAMA_FINAL[28]; // caracter n=norte s=sur
CHAR_LONGITUD = TRAMA_FINAL[41]; // caracter e=este w=oeste
}

//*****
//***** FUNCIÓN SUPERVISAR_GPS *****

VOID SUPERVISAR_GPS(VOID)
{
    LONG INDICE;
    LONG N_SECTORES;
    UNSIGNED INT F_MUESTREO;

    IF (TASK_RX_GPS() == TRUE)
    // si llega una trama gps completa, trama_comp[]=${"$gpqga"}..*
    {
        IF (TRAMA_FINAL[43] == '1') // trama contiene datos válidos
        {
            OUTPUT_LOW(LED_GPS); // enciende led
            EXTRAER_COORDENADAS(); // extraer las coordenadas

            IF(COORDENADA_CERCA(&INDICE,&N_SECTORES,LATITUD,CHAR_LATITUD,LONGI
TUD,CHAR_LONGITUD,&F_MUESTREO) == TRUE)
            {
                IF (VOZ_ACTIVADA == FALSE) // variable para evitar que una vez se
// active la voz no lo siga haciendo
                {
                    ACTIVAR_VOZ(INDICE,N_SECTORES,F_MUESTREO);
                    VOZ_ACTIVADA = TRUE;
                }
            }
            ELSE
                VOZ_ACTIVADA = FALSE;
        }
        ELSE
            OUTPUT_HIGH(LED_GPS); // apagar led indicador gps, sin no se enlaza
// con al menos 3 satélites
    }
}

//*****
//***** FUNCIÓN COORDENADA_CERCA *****

SHORT COORDENADA_CERCA(LONG LONG *INDEX, LONG LONG *N_SECTOR,FLOAT
LAT,CHAR C_LAT,FLOAT LON,CHAR C_LON,UNSIGNED INT *FREQ_MUESTREO)
{
    UNSIGNED LONG I;
    UNSIGNED LONG J;
    UNSIGNED INT CANT_VOCES;
    UNSIGNED INT CANTIDAD_WAV;

```

```

UNSIGNED INT SENSIBILIDAD_GPS;
LONG LAT_COMP;
LONG LON_COMP;
CHAR LAT_C_COMP ;
CHAR LON_C_COMP;

IF (INICIAR_MMC() == TRUE)
  IF (MMCSO_INIT() == 0)
  {
    IF (LEER_SD_SECTOR(0) == TRUE) // El sector 0 en la posición 0,
                                  // contiene la cantidad de voces grabadas
      IF (MMCSO_READ_BLOCK(0,512,G_MMCSO_BUFFER) == 0 )
        {
          CANTIDAD_WAV = G_MMCSO_BUFFER[0];
          SENSIBILIDAD_GPS =G_MMCSO_BUFFER[1];
          CANT_VOSES =0;
          FOR (I = 1 ; I <= 10 ; I = I++)
            // desde el sector 1 y hasta el sector 10
            {
              IF (LEER_SD_SECTOR(I) == TRUE)
                IF (MMCSO_READ_BLOCK(I,512,G_MMCSO_BUFFER) == 0 )
                  {
                    FOR (J=0 ; J <= 511 ; J = J+32)
                      {
//parámetro que retorna la dirección donde comienza el archivo WAV
dentro de la SD
                        *INDEX = MAKE32(G_MMCSO_BUFFER[J],G_MMCSO_BUFFER[J+1],
                                          G_MMCSO_BUFFER[J+2],G_MMCSO_BUFFER[J+3]);

//parámetro que retorna la cantidad de sectores que ocupa el archivo
WAV dentro de la SD
                        *N_SECTOR=MAKE32(G_MMCSO_BUFFER[J+4],G_MMCSO_BUFFER
                                          [J+5],G_MMCSO_BUFFER[J+6],G_MMCSO_BUFFER[J+7]);

                        *FREQ_MUESTREO = G_MMCSO_BUFFER[J+27];

LAT_COMP=CONV_ASCII_COORDENADAS('0',G_MMCSO_BUFFER
                                  [J+8],G_MMCSO_BUFFER[J+9],G_MMCSO_BUFFER[J+10],
                                  G_MMCSO_BUFFER[J+11],G_MMCSO_BUFFER[J+12],
                                  G_MMCSO_BUFFER[J+13],G_MMCSO_BUFFER[J+14],
                                  G_MMCSO_BUFFER[J+15]);

LAT_C_COMP = G_MMCSO_BUFFER[J+16]; // caracter N o S

LON_COMP=CONV_ASCII_COORDENADAS(G_MMCSO_BUFFER
                                  [J+17],G_MMCSO_BUFFER[J+18],G_MMCSO_BUFFER[J+19],
                                  G_MMCSO_BUFFER[J+20],G_MMCSO_BUFFER[J+21],
                                  G_MMCSO_BUFFER[J+22],G_MMCSO_BUFFER[J+23],
                                  G_MMCSO_BUFFER[J+24],G_MMCSO_BUFFER[J+25]);

LON_C_COMP = G_MMCSO_BUFFER[J+26]; // caracter E o W

```

```

        IF ( (LAT_C_COMP == C_LAT) && (LON_C_COMP == C_LON) )
        {
            IF(DISTANCI_CORTA (LAT,LAT_COMP,LON,LON_COMP)=TRUE);
            RETURN(TRUE);
        }
        CANT_VOCEs = CANT_VOCEs + 1;
        IF (CANT_VOCEs >= CANTIDAD_WAV)
            RETURN(FALSE);
    }
}
ELSE
{
    ON_OFF_LED_SD(2);
    RETURN(FALSE);
}
}
ELSE
    ON_OFF_LED_SD(3);
}
ELSE
    ON_OFF_LED_SD(4);
    RETURN (FALSE);
}

//*****
//***** FUNCIÓN ACTIVAR_VOZ *****

VOID ACTIVAR_VOZ(LONG LONG INDEX, LONG LONG N_SECTOR,UNSIGNED INT
FREQ_MUESTREO)
{
    DISABLE_INTERRUPTS(GLOBAL);
    IF( TRANSFERIR_SD_RAM(INDEX,N_SECTOR) == TRUE);
    TRANSFERIR_RAM_DAC( INDEX,N_SECTOR,125);
    ENABLE_INTERRUPTS(GLOBAL);
}

//*****
//***** FUNCIÓN TRANSFERIR_RAM_DAC *****

SHORT TRANSFERIR_RAM_DAC(LONG LONG INDEX, LONG LONG N_SECTOR, UNSIGNED
INT TIEMPO)
{
    LONG LONG I;
    LONG J;
    LONG PAGINA = 0;
    INT K;

    OUTPUT_LOW(LED_RAM);
    OUTPUT_LOW(LED_DAC);
    SETUP_SPI_MODO_3(0);
    RD_MEM_BUFFER_1(PAGINA);
    FOR (I = INDEX ; I <= N_SECTOR ; I++)
    {

```

```

        IF (WAIT_NOBUSY_RAM()==TRUE)
        {
            PAGINA++;
            RD_BUFFER_1_H();
            FOR(J=0 ; J <= 255 ; J++)
            {
                WR_DAC(SD_BUFFER[J]);
                IF (J < 255)
                    DELAY_US(TIEMPO);
            }
        }
        RD_MEM_BUFFER_1(PAGINA);
        IF (WAIT_NOBUSY_RAM()==TRUE)
        {
            PAGINA++;
            RD_BUFFER_1_H();
            FOR(J=0 ; J <= 255 ; J++)
            {
                WR_DAC(SD_BUFFER[J]);
                IF (J < 255)
                    DELAY_US(TIEMPO);
            }
        }
        RD_MEM_BUFFER_1(PAGINA);
    }
    OUTPUT_HIGH(LED_RAM);
    OUTPUT_HIGH(LED_DAC);
    RETURN(TRUE);
}

//*****
//***** PRINCIPALES RUTINAS UNIDAD DE CONTROL*****

#include <18F4550.H>

#define DELAY(CLOCK=4800000)
#define FAST_IO(A)
#define FAST_IO(B)
#define FAST_IO(C)
#define FAST_IO(E)

//*****
//***** FUSIBLES *****

#define NOMCLR // Desactivación el MCLR para usar RE3
#define NOWDT // No Watch Dog
#define NOPROTECT
#define PUT
#define NOLVP
#define NOCPD

//*****
//***** ASIGNACIÓN DE PUERTOS *****

```

```

#define LED          PIN_E2
#define CONEXUSB    PIN_E3

#define STOPM1      PIN_D0    // MATRIZ INFORMACIÓN
#define OKM1        PIN_C2
#define I2C1_1      PIN_B5
#define I2C1_CL     PIN_B4

#define I2C2_1      PIN_B3    // MATRIZ RUTAS
#define I2C2_CL     PIN_B2
#define STOPM2      PIN_B1
#define OKM2        PIN_B0

#define STOPM3      PIN_C1    //MATRIZ ITINERARIO
#define OKM3        PIN_C0
#define I2C3_1      PIN_B7
#define I2C3_CL     PIN_B6

//*****
//***** SERVICIO INTERRUPCIÓN INT_RDA *****

#INT_RDA

VOID RX_SERIE(VOID)
{
    RS232_DATA=GETC();
    IF(RS232_DATA == 3) //Prepara para recibir rutas de muestra en lcd
    {
        DATA_INI = TRUE;OK_SEPARA=1;
    }
    ELSE IF(RS232_DATA==2)//Prepara para recibir paradas de muestra en lcd
    {
        DATA_INI = TRUE;OK_SEPARA=2;
    }
    ELSE IF(RS232_DATA == 1 )//Finaliza la transmisión para lcd
    {
        DATA_INI = FALSE;
    }
    ELSE IF(RS232_DATA == 4)//Prepara para recibir datos de ruta
    {
        DATA_INI = TRUE;OK_SEPARA=3;
    }
    ELSE IF(RS232_DATA == 6)//Prepara para recibir datos los itinerarios
    {
        DATA_INI = TRUE;OK_SEPARA=4;
    }
    ELSE IF(RS232_DATA == 8)//Prepara para recibir datos de próxima parada
    {
        DATA_INI = TRUE;OK_SEPARA=5;
    }
    ELSE IF((RS232_DATA == 5 ) || (RS232_DATA == 7) || (RS232_DATA == 9))
        //Finaliza la transmisión de datos para las matrices
    {
        DATA_INI = FALSE;
    }
}

```

```

    }
    ELSE IF (DATA_INI == TRUE)
    {
        PUSH(RS232_DATA, BUFFER_RX, &P_BEGIN_RX, &P_END_RX, TOPE);
    }
}

//*****
//***** FUNCIÓN CONV_INT_ASCII *****
//*****

VOID CONV_INT_ASC(INT VALOR, INT *DAT_100, INT *DAT_10, INT *DAT_1)
{
    CHAR IP_GA_MA[3];

    IF (VALOR <= 9)
        SPRINTF(IP_GA_MA, "%S%U", "00", VALOR);
    ELSE IF ((VALOR >= 10) && (VALOR <= 99))
        SPRINTF(IP_GA_MA, "%S%U", "0", VALOR);
    ELSE IF (VALOR >= 100)
        SPRINTF(IP_GA_MA, "%U", VALOR);

    *DAT_100 = IP_GA_MA[0];
    *DAT_10  = IP_GA_MA[1];
    *DAT_1   = IP_GA_MA[2];
}
//*****
//***** ENVÍO DE TRAMA RUTAS *****
//*****

PUTC(4);
FOR(I=0;I<30;I++)
{
    CONV_INT_ASC(RUTA4[I], &VAR_100, &VAR_10, &VAR_1);
    PUTC(VAR_100);
    PUTC(VAR_10);
    PUTC(VAR_1);
}
PUTC(5);

//*****
//***** FUNCIÓN CONV_ASCII_INT *****
//*****

INT CONV_ASC_INT(CHAR DAT_100, CHAR DAT_10, CHAR DAT_1)
{
    VALOR=0;
    VALOR = VALOR + ((DAT_100-48) * 100);
    VALOR = VALOR + ((DAT_10-48) * 10);
    VALOR = VALOR + ((DAT_1-48) * 1);
    RETURN (VALOR);
}

//*****
//***** FUNCIÓN MATRIZ_RUTA *****
//*****

```

```

VOID MATRIZ_RUTA (VOID)
{
    SHORT INT STATUS;
    INT I=0;
    INT ADDRESS=0;
    OUTPUT_HIGH(STOPM2);
    WHILE ( INPUT(OKM2) == 0)
    {
    SET_TRIS_B(0B11110001);
    FOR (I=0;I<30;I++)
    {
        I2C_START(I2CB);           // Inicio de la transmisión
        I2C_WRITE(0XA0);          // Dirección i2c en modo escritura
        I2C_WRITE(ADDRESS>>8);    // Envía parte alta dirección
        I2C_WRITE(ADDRESS);       // Envía parte baja dirección
        I2C_WRITE(DATA_RUTA[I]);  // Envía trama de datos de la matriz ruta
        I2C_STOP(I2CB);           // Finaliza la transmisión
        I2C_START(I2CB);          // Reinicio de la comunicación
        STATUS=I2C_WRITE(0XA0);   // Escritura correcta
        WHILE(STATUS==1)         // Si es 1 espera respuesta de eeprom
        {
            I2C_START(I2CB);
            STATUS=I2C_WRITE(0XA0);
        }
        I2C_STOP(I2CB);
        ADDRESS++;
    }
}

```

```

'=====
'           Funciones de llamada a la libreria MPUSBAPI.DLL
'=====

'-----
' Funciones equivalentes para llamar desde Visual Basic
'-----

```

```

Public Declare Function MPUSBGetDLLVersion Lib "mpusbapi.dll" () As Long
Public Declare Function MPUSBGetDeviceCount Lib "mpusbapi.dll" (ByVal
pVID_PID As String) As Long
Public Declare Function MPUSBOpen Lib "mpusbapi.dll" (ByVal instance As
Long, ByVal pVID_PID As String, ByVal pEP As String, ByVal dwDir As Long,
ByVal dwReserved As Long) As Long
Public Declare Function MPUSBClose Lib "mpusbapi.dll" (ByVal handle As
Long) As Long
Public Declare Function MPUSBRead Lib "mpusbapi.dll" (ByVal handle As
Long, ByVal pData As Long, ByVal dwLen As Long, ByRef pLength As Long,
ByVal dwMilliseconds As Long) As Long
Public Declare Function MPUSBWrite Lib "mpusbapi.dll" (ByVal handle As
Long, ByVal pData As Long, ByVal dwLen As Long, ByRef pLength As Long,
ByVal dwMilliseconds As Long) As Long

```

```
Public Declare Function MPUSBReadInt Lib "mpusbapi.dll" (ByVal handle As Long, ByVal pData As Long, ByVal dwLen As Long, ByRef pLength As Long, ByVal dwMilliseconds As Long) As Long
```

```
'-----  
' Constantes de conectividad con el PIC  
'-----
```

```
Public Const vid_pid = "vid_04d8&pid_0011" ' Vendor id (Microchip) y Periférico id
```

```
Public Const out_pipe = "\MCHP_EP1" ' Endpoint de transmisión  
Public Const in_pipe = "\MCHP_EP1" ' Endpoint de recepción
```

```
'-----
```

```
//*****  
//*****PICUSBNUT.H*****
```

```
// Este ejemplo muestra como configurar el dispositivo USB y sus  
// descriptores. Los únicos cambios con respecto a su versión  
// original (usb_desc_scope.h) han sido realizados en el apartado  
// start device descriptors, concretamente el vendor y product id  
// tomado de: www.hobbypic.com.  
// La versión completa se encuentra en el directorio C:\Archivos de  
// programa\PICC\Drivers\usb_desc_scope.h
```

```
#include <usb.h>
```

```
////////////////////////////////////  
///  
/// start device descriptors  
///  
////////////////////////////////////
```

```
//device descriptor  
char const USB_DEVICE_DESC[] =  
{  
    USB_DESC_DEVICE_LEN, //the length of this report  
    0x01, //constant DEVICE (0x01)  
    0x10,0x01, //usb version in bcd  
    0x00, //class code  
    0x00 //subclass code  
    0x00 //protocol code  
    USB_MAX_EP0_PACKET_LENGTH, //max packet size for endpoint 0.  
    0xD8,0x04, //vendor id (0x04D8 is Microchip)  
    0x11,0x00, //product id  
    0x01,0x00, //device release number  
    0x01, //index of string description of manufacturer  
    0x02, //index of string descriptor of the product  
    0x00, //index of string descriptor of serial number  
    USB_NUM_CONFIGURATIONS //number of possible configurations  
};
```

ANEXO C. ECUACIONES DE COTICCHIA SURACE PARA LA CONVERSIÓN DE COORDENADAS GEOGRÁFICAS A UTM

Las ecuaciones de Coticchia Surace permiten convertir las coordenadas obtenidas en la lectura del receptor GPS, las cuales corresponden a coordenadas geográficas o geodésicas, a coordenadas UTM expresadas en metros. La precisión de las coordenadas UTM depende del tipo de variables que se utilicen en el cálculo; empleando variables de coma flotante y doble precisión, se logra un error inferior al metro. Se debe tener en cuenta el elipsoide sobre el cual se miden las coordenadas, que para el caso del receptor GPS corresponde al elipsoide nativo del sistema, el WGS-84.

Datos del elipsoide WGS-84			
$a = \text{semieje mayor}$	6378137	$b = \text{semieje menor}$	6356752,314245
$e = 1^{\circ} \text{ excentricidad}$	0,081819191	$e' = 2^{\circ} \text{ excentricidad}$	0,082094438
$e = \sqrt{a^2 - b^2}/a$		$e' = \sqrt{a^2 - b^2}/a$	
$c = \text{radio curvatura}$	6399593,626	$\alpha = \text{aplanamiento}$	
$c = a^2/b$		$\alpha = a - b/a$	

Ecuaciones de Coticchia Surace			
$X_{UTM} = [\xi \cdot \nu \cdot (1 + (\zeta/3))] + 500.000$		$Y_{UTM} = [\eta \cdot \nu \cdot (1 + \zeta)] + B_{\phi}$	
ξ	$\frac{1}{2} \ln [(1 + A)/(1 - A)]$	η	$\text{arc tan}(\tan \varphi^* / \cos \Delta\lambda)$
ν	$0,9996 \cdot c / \sqrt{1 + e'^2 \cdot \cos^2 \varphi^*}$	B_{ϕ}	$0,9996 \cdot c(\varphi^* - \alpha J_2 + \beta J_4 - \gamma J_6)$
ζ	$\frac{1}{2} e'^2 \cdot \xi^2 \cdot \cos^2 \varphi^*$	α	$3e'^2/4$
A	$\cos \varphi^* \cdot \sin \Delta\lambda$	β	$5a^2/3$
J_2	$\varphi^* + (A_1/2)$	γ	$35a^3/27$
J_4	$(3J_2 + A_2)/4$	A_1	$\sin 2\varphi^*$
J_6	$(5J_4 + A_2 \cos^2 \varphi^*)/3$	A_2	$A_1 \cos^2 \varphi^*$
$\Delta\lambda$	$\lambda^* - \lambda_0^*$	λ_0	$6\text{huso} - 183$
huso	$(\varphi_{\text{sexadecimal}}/6) + 31$		

Ejemplo: conversión de coordenadas geográficas a UTM			
$\lambda = \text{latitud}$		$\varphi = \text{longitud}$	
$\lambda_{\text{geográfica}}$	1°11'37.29''	$\varphi_{\text{geografica}}$	77°16'37.52''
$\lambda_{\text{sexadecimal}}$	1,193691667	$\varphi_{\text{sexadecimal}}$	-77,27708889
$\lambda_{\text{radianes}}$	0,02083385	$\varphi_{\text{radianes}}$	-1,348739637
$X_{UTM} = [-0,039744529 \times 6375595,007(1 + (5,32064E^{-06}/3))] + 500000$			
$X_{UTM} = 246604,5328 \text{ m}$			
$Y_{UTM} = [1,64594E^{-05} \times 6375595,007(1 + 5,32064E^{-06})] + 131938,986$			
$Y_{UTM} = 132043,9251 \text{ m}$			

* En radianes

ANEXO D. RELACIÓN DE PARADEROS Y RUTAS PARA LA CIUDAD DE PASTO

PARADERO	RUTAS
Gorety	E5 – E6 – C4 – C5 – C10
Cesmag	E5 – C4 – C5 – C10
Castellana	E4 – E5 – C2 – C3 – C5 – C16 – C4 – C7
Sanyo	E5 – C2 – C3 – C4 – C10
Centro Comercial Popular	E5 – C2 – C3 – C4 – C10 – C14
Pollo Sorpresa	E5 – C2 – C4 – C10 – C14
Escuela # 1	C3 – C4 – C7 – C15 – C16
Bombona	E4 – E5 – E6 – E7 – C1 – C2 – C7 – C13
San Andresito Cra 22	E6 – E7 – E8 – C5 – C6
Champagnat	E4 – E5 – C1 – C2 – C4 – C7 – C8 – C11 – C16
Electromillonaria	E4 – E5 – C1 – C2 – C4 – C7 – C8 – C11 – C16
Cable Unión	E4 – E5 – E7 – C2 – C4 – C6 – C15 – C16
San Andresito Cll 15	E4 – E5 – C2 – C4 – C6 – C15 – C16
Clínica Fátima	C1 – C9 – C10 – C11 – C12 – C13 – C14
Javeriano	C1 – C9 – C10 – C11 – C12 – C13 – C14
Sede pensionados	E7 – C1 – C10 – C11 – C12 – C13 – C14
Telecom	E2 – E4 – E5 – C1 – C2 – C4 – C7 – C8 – C11 – C14 – C16
Regirepuestos	E2 – E4 – E5 – C1 – C2 – C4 – C7 – C8 – C11 – C14 – C16
Idema	E2 – E4 – E5 – C1 – C2 – C4 – C7 – C8 – C10 – C11 – C14
Iglesia Panadería	E7 – C1 – C8 – C10 – C12 – C13 – C14
Normal	E3 – C1 – C3 – C4 – C6 – C15 – C16
Museo Taminango	E3 – C3 – C4 – C15 – C16
Alcaldía Cra 26	C3 – C4 – C15 – C16
Bolívar Plaza	E1 – C3 – C13 – C14 – C16
Parque Bolívar	E1 – C3 – C13 – C16
Rhin	E6 – E7 – C5
Corazón de Jesús	E7 – C8 – C9 – C15
Iglesia Niño Jesús	E3 – C5 – C12
Colegio San Felipe Neri	E3 – C12
Drogas la Rebaja	C12 – C13 – C14
Navarrete	C12 – C13 – C14
Comfamiliar	E1 – E6 – C12 – C13
Ur. Gran Colombia	E1 – E6 – C12 – C13
Cresemillas	E3 – C1 – C3 – C4 – C6 – C15 – C16
Fransiscanas	E1 – E2 – E5 – C9 – C10 – C13 – C14
Antiguo SPA	E1 – E2 – E5 – C5 – C10 – C14
Pollos Ruano	E1 – E2 – E5 – C5 – C10 – C14
Pandiaco	E1 – E2 – E4 – E5 – C5 – C9 – C10 – C12 – C13 – C14
Terminal Mixto	E1 – E2 – E4 – E5 – C5 – C9 – C10 – C12
Universidad de Nariño	E1 – E2 – E4 – E5 – C5 – C9 – C10 – C12
Ur. Marsella	E1 – E2 – E4 – E5 – C5 – C9 – C10 – C12
CEHANI	E1 – E2 – E4 – E5 – C5 – C9 – C10 – C12
La Merced	E6 – E7 – C2 – C3 – C11

Fideos Marta	E6 – E7 – C2 – C3 – C11
Casa de Ejercicios	E4 – E5 – E6 – E7 – C2 – C13
Cedritos	E4 – E6 – E7 – C2
Aleph Teatro	E4 – E5 – E6 – C2 – C13
Campos de Castilla	E4 – E5 – E7 – C2 – C13
Mercola	E2 – E3 – E4 – E5 – C1 – C2 – C3 – C4 – C5 – C6 – C7 – C8 – C11 – C16
San Juan Bosco	E2 – E3 – E4 – E5 – C1 – C2 – C3 – C4 – C5 – C6 – C7 – C8 – C11 – C16
Caja Agraria	E2 – E3 – E4 – C1 – C2 – C11 – C16
Banco de la República	E2 – E3 – E4 – C1 – C2 – C11 – C15 – C16
Sarasty	E1 – E2 – E3 – E4 – C1 – C7 – C11 – C13 – C15 – C16
Cedenar	E1 – E2 – E3 – E4 – C1 – C7 – C13 – C15 – C16
Coomeva	E2 – C1
Caracol	E1 – E3 – E4 – C7 – C13 – C15 – C16
La California	E3 – E4 – C4 – C7 – C15 – C16
El Buen Sazón	E1 – E2 – E6
San Andrés	E4 – E5 – E7 – C2 – C10 – C14
Sevilla	E3 – C3 – C4 – C7 – C15 – C16
Davivienda	E6
San Agustín	E1 – E2 – E6
Herlú	E3
Amorel	E1 – E2 – E6
Hotel Chambú	E1 – E2 – E3 – E4 – E5 – E6 – C1 – C2 – C3 – C4 – C5 – C7 – C8 – C10 – C11 – C16
Bancolombia	E1 – E2 – E3 – E5 – E6 – C1 – C2 – C3 – C4 – C5 – C7 – C8 – C11 – C16
San Juan Bosco	E1 – E2 – E3 – E5 – E6 – C1 – C2 – C3 – C4 – C5 – C7 – C8 – C11 – C16
DAS	E1 – E2
Cacharrería Cali	E2 – C1
Whoperking	E2 – C1 – C9
Universidad Mariana	E1 – E2 – C9 – C10 – C14
Clínica Palermo	E1 – E2 – C9 – C10 – C14
Glorieta Banderas	E1 – E2 – E5 – C9 – C10 – C13 – C14

ANEXO E. ESPECIFICACIONES TÉCNICAS DE LOS PANELES ELECTRÓNICOS DE MATRIZ DE LEDS

Alimentación Eléctrica	12 Vcc
Instalación	Interior
Color de despliegue de información	Rojo
Angulo de visibilidad	Horizontal
Vida útil esperada	100.000 Horas.
Montaje	Sin partes desmontables, cubierta en perfiles de aluminio estructurado en una caja.
Panel frontal	
	
Formato de matriz completa	Matriz de 7 líneas x 80 columnas con distancia entre leds de 14mm.
Área de exhibición	80mm de altura x 900mm de largo.
Configuración de caracteres	13 caracteres configurables por software
Dimensión del gabinete	135mm de altura x 930mm de largo x 45mm de profundidad.
Despliegue de textos	En formato de caracter de 6 x 7 píxeles (130mm de altura) de una línea de información. Despliegue de información de los itinerarios de ruta. Presenta como ventaja densidad media de leds generando una excelente visualización de la tipografía a distancia.
Panel interior	
	
Formato de matriz completa	Matriz de 8 líneas x 80 columnas con distancia entre leds de 3mm.
Área de exhibición	60mm de altura x 610mm de largo.
Configuración de caracteres	13 caracteres configurables por software
Dimensión del gabinete	90mm de altura x 640mm de largo x 50mm de profundidad

Despliegue de textos	En formato de caracter de 7 x 8 píxeles (64mm de altura) de una línea de información. Despliegue de información de paradas. Visualización más uniforme a corta distancia permitiendo una mejor lectura.
Panel posterior 	
Formato de matriz completa Área de exhibición Configuración de caracteres Dimensión del gabinete Despliegue de textos	Matriz de 10 líneas x 24 columnas con distancia leds de 10 mm. 160mm de altura x 360mm de largo. 3 caracteres configurables por software 260mm de altura x 410mm de largo x 40mm de profundidad. En formato de caracter de 7 x 10 píxeles (160mm de altura). Despliegue de información de número de ruta.